Diss. ETH No. 19229

Multi-Label Classification and Clustering for Acoustics and Computer Security

A dissertation submitted to

Swiss Federal Institute of Technology

Zurich

for the degree of

Doctor of Sciences

presented by

Andreas Peter Streich

Dipl. Ing. (ETH Zurich)

born 29 April 1980

citizen of Basel, Switzerland

accepted on the recommendation of

Prof. Dr. Joachim M. Buhmann, examiner

Prof. Dr. David Basin, co-examiner

Dr. Stefan Launer, co-examiner

2010

Abstract

This thesis focuses on classification and clustering of data where a part of the data items are jointly emitted by several sources. We design an abstract generative model which offers a clear semantic interpretation for such data. Based on this model, we derive algorithms for multi-label classification and multi-assignment clustering.

For the task of multi-label classification, we show that the presented algorithms estimate source parameters more accurately and classify data items more reliably than previously proposed methods. We apply our method to classify acoustic streams in hearing instrument. Most modern hearing instruments rely on such classification to adapt to acoustic scenes encountered in daily live. In this setting, a correct detection of the present sources is essential to provide comfortable listening in spite of a hearing impairment. We propose a novel set of features for this classification task and show that our generative multi-label classification algorithm outperforms current techniques.

The generality of our model formulation allows us to describe prior work in the same framework. Starting from this unified specification, we derive the asymptotic distribution of the parameter estimators obtained by several algorithms. Furthermore, we prove that a class of popular model assumptions implies a mismatch to the assumed generative process and therefore causes an inconsistency of the parameter estimators and, consequently, suboptimal classification results.

The generative algorithms for multi-assignment clustering are applied to Boolean data. Also in this unsupervised setting, the parameters estimated by the proposed algorithms are more precise and the obtained clustering solution attains higher stability, both compared to state-of-the-art methods. We apply our method to solve an important problem in computer security

known as role mining. The Permissions of new users can be specified more precisely with the roles obtained by our generative methods than with roles detected by other multi-assignment clustering techniques.

To compare the quality of different clustering techniques independently of particular assumptions, we apply the framework of approximation set coding for cluster validation. We observe that the model selection based on this general framework is in agreement with the selection based on specific quality measures for multi-assignment clustering. According to both criteria, the proposed algorithms are identified as the best method for the given clustering task. We thus show for the first time that approximation set coding correctly regularizes the model complexity for a real-world learning task.

Zusammenfassung

Diese Dissertation behandelt die Klassifikation und das Gruppieren von Daten unter der Annahme, dass mindestens ein Teil der Datenpunkte von mehreren Quellen gemeinsam generiert wird. Wir entwerfen ein allgemeines generatives Modell mit einer klaren Semantik für derartige Daten. Basierend auf diesem Modell entwickeln wir Algorithmen für die Klassifikation mit Mehrfachzugehörigkeiten und für die Gruppierung mit Mehrfachzuweisungen.

Im ersten Teil der Arbeit gehen wir detailliert auf Klassifikationsprobleme ein, in denen ein Datenelement gleichzeitig zu mehreren Klassen gehören kann. Die von uns vorgestellten Algorithmen schätzen Quellenparameter genauer und klassifizieren synthetische Daten präziser als bisher bekannte Methoden. Anschliessend wenden wir unsere Algorithmen auf die Klassifikation von akustischen Daten an. Die meisten modernen Hörgeräte teilen die akustischen Signale in verschiedene Klassen ein und wählen anschliessend, entsprechend der geschätzten Klasseneinteilung, die der Situation angepasste Verarbeitung des Signals. Dementsprechend hängt die Gesamtleistung des Hörgerätes grundlegend von der korrekten Identifikation der vorhandenen Geräuschquellen ab. Wir präsentieren neue Kenngrössen für diese Klassifikationsaufgabe. In verschiedenen Experimenten ergeben sowohl die vorgestellten Merkmale als auch der generative Ansatz verbesserte Resultate gegenüber dem aktuellen Stand der Technik.

Die Allgemeinheit unserer Formulierung ermöglicht uns ausserdem, die bisherigen Klassifikationsmethoden als Spezialfälle unseres Modells darzustellen. Ausgehend von dieser einheitlichen Beschreibung leiten wir die asymptotische Verteilung der Parameterschätzer verschiedener Methoden her. Wir beweisen ausserdem, dass eine gängige Modellannahme eine Fehlanpassung des Modells an die Daten impliziert und dadurch zu inkonsisten-

ten Parameterschätzern sowie sub-optimalen Klassifikationsresultaten führt.

Im zweiten Teil untersuchen wir die unüberwachte Gruppierung von Daten unter der Verallgemeinerung, dass ein Datenelement gleichzeitig zu mehreren Gruppen gehören kann. Auch in diesem unüberwachten Szenario liefern die vorgeschlagenen generativen Algorithmen präzisere Schätzer der Quellenparameter und ermöglichen eine genauere Beschreibung von neuen Datenelementen, beides im Vergleich zu bisherigen Methoden. Wir wenden den generativen Gruppierungsalgorithmus auf ein wichtiges Problem aus der Computersicherheit an, nämlich dem automatischen Ermitteln einer Menge von Rollen für rollenbasierte Zugangskontrolle. Die von den vorgeschlagenen Algorithmen gefundenen Rollen beschreiben die Zugriffsrechte neuer Benutzer akkurater als die Rollen, welche von bisherigen Methoden mit Mehrfachzuweisungen gefunden werden.

Die Bewertung der Qualität einer Datengruppierung basiert häufig auf Annahmen über die Natur der Datengruppen. Wir verwenden die Methode der Codierung mittels Näherungsmengen um die Qualität der Lösungen verschiedener Gruppierungsalgorithmen zu beurteilen. Die Modellpräferenzen dieser allgemeinen Methode stimmen mit der Auswahl auf Grund von problemspezifischen Kenngrössen überein. Dieses Modellselektionsprinzip identifiziert den vorgeschlagenen Algorithmus als für die vorliegende Gruppierungsaufgabe am besten geeignet. Damit wurde zum ersten Mal an Realweltdaten bestätigt, dass Codierung mittels Näherungsmengen die Modellkomplexität korrekt kontrolliert.

Contents

Chapter 1

Introduction

The word *data* is the Latin plural of datum, which is the past participle of *dare*, "to give", hence "something given". The understanding of data in machine learning follows its etymology: Data are numbers, words, images, etc., accepted as they stand. They are viewed as the lowest level of abstraction and the base from which information and knowledge are to be derived [59].

This thesis focuses on data which is obtained from measurements on objects and thus describes properties of these objects. For many types of objects, the properties of a single object may have been generated simultaneously by several sources. Vivid examples are acoustic situations, where the emissions of various sources superpose each other and thus constitute the sound waves perceived by a human ear or measured in a microphone. For example in a cocktail bar, the properties of the acoustic situation are determined by the people discussing in parallel with the music playing in the background. An other example of properties generated by several sources are the permissions of an employee in the computer system of a company: These permissions typically consist of a set of general permissions granted to all employees and more specific permissions required to fulfil the tasks of the person in the company. The provenance of user permissions is thus adequately described by a source or *role* which describes the general permissions, and a set of roles describing the permissions for specialized duties. A user obtains all permissions contained in at least one of his roles.

In this thesis, we advocate a generative approach to model and process data generated by several sources. To achieve this goal, we first design a generative model which describes the assumed generative

process for the data at hand. Such a model typically contains a number of parameters which are to be learned based on a set of observed data.

The machine learning tasks arising in this context can be roughly split into two groups:

- If the sources that generated the data items are known, we speak of *supervised learning*. Such a situation is given e.g. by a set of recordings together with labels describing the scenery. Possible tasks in such a scenario include the automated labelling of a new recording (i.e. *classification*), or the identification of the emissions of the individual sources (i.e. *source separation*).

- The setting where the contributing sources are unknown is called *unsupervised learning*. We encounter this data analysis challenge if we are asked to group the data items according to some similarity criteria. For example, consider your friends: you may group them according to their music preference or their favourite sports. Some of these groups will be overlapping, as some of your friends might enjoy Beethoven's oeuvre as much as Madonna songs, or might be both passionate football players and skiers.

The set of properties measured from an object depends on the nature of the object under study and is typically large and diverse: The properties of an acoustic situation reach from physical characteristics such as sound intensity and frequency spectrum to qualities such as the mood of a music track. For an employee in a company, properties include his function, work place, salary, dress code and access permissions in the building as well as within the computer system. When investigating an object, we usually focus on a subset of properties: A person does not listen to a conversation partner in the same way as to street noise on a big road crossing at rush hour. In a company, the human resource department, the line manager and the IT security group are all interested in different characterizations of the same employee.

The measurement and calculation of different properties of an object is denoted by *feature extraction*. In practice, this is often a two-step process, consisting of a physical measurement and a subsequent transformation of the measurement results into a more appropriate representation. In acoustics, the physical properties are most commonly measured with a microphone which yields an electric signal. Discretisation and the extraction of Fourier

coefficients to obtain a spectral representation are popular transformations of this signal.

The relevance of individual features typically depends on the objective of the subsequent processing. *Feature design* characterizes the task of drafting a set of features, while *feature selection* chooses the most ones important for a given task. The second step formalizes the selective perception outlined above based on different criteria to measure the relevance of a feature for a given task. The design and selection of features are crucial steps in all machine learning applications, as they define which properties of the object under study are retrieved and how they are measured. The data obtained in this way is the only representation of the object in the subsequent processing. The chosen feature set thus has to capture all properties which are relevant for the subsequent machine learning task.

1.1 Thesis Overview

The thesis starts with the generic model describing how data items from several sources are generated. This model is detailed in Chapter 2. Later in this same chapter, we present and discuss the quality measures we will use throughout this thesis.

The applications of this generative model to machine learning problems are grouped in two parts. The first part covers our contributions in the field of supervised learning. Chapter 3 motivates multi-label classification and discusses connections with related problems. In Chapter 4, we present two feature sets based on sound-field indicators and evaluate their utility to predict the hearing target and the reverberation intensity in acoustic scenes. A generative classifier for multi-label classification is presented in Chapter 5 and evaluated both on synthetic and real-world data. The asymptotic distribution of estimators on multi-label data is studied in Chapter 6. The theoretical results are verified in experiments on discrete and continuous data and show that the proposed multi-label classification algorithm outperforms competing methods. Finally, we prove that some of the widely used techniques for inference based on multi-label data incur a model mismatch and therefore yield biased parameter estimators and sub-optimal classification results. The corresponding theorem is given in Chapter 7 and concludes the first part.

The second part is concerned with unsupervised learning. The requirement for multi-assignment clustering and the problem of role mining are

discussed in Chapter 8. Our approach to this problem is based on the assumed generative process and is presented in Chapter 9 along with extensive experiments on synthetic data and real-world access-control information for role mining. The evaluation of the proposed multi-assignment clustering method in the framework of approximation set coding is given in Chapter 10 and concludes the second part.

We draw the conclusions of this thesis in Chapter 11.

1.2 Original Contributions

The main contributions of this thesis are the following:

- In cooperation with Alfred Stirnemann and Manuela Feilner from Phonak, we have developed and analyzed a novel set of features based on physical properties of the sound field. This feature set enables several state-of-the-art algorithms to classify acoustic streams with higher accuracy.

- The semantics of data generated by multiple sources has been unclear in most situations. We have formulated a generic, generative process for data emitted by multiple sources which facilitates a consistent understanding of such data.

- Based on the assumed data-generating process, we have developed a generative classification algorithm for multi-label data. Experiments on synthetic data show that the proposed method estimates the parameters significantly more accurately and classifies data items with lower error rates than methods based on less specific assumptions on the generative process. Experiments on real-world acoustic data confirm the superior performance namely in the case of small training data sets.

- On a theoretical level, we have derived the asymptotic distribution of parameter estimators based on multi-label data in a general setting. The predictions on the parameter accuracy based on these theoretical results closely agree with the measurements from simulations in several concrete examples.

- To obtain a clear conclusion on the type of assumptions that lead to sub-optimal classification results, we have proven that a particu-

lar class of algorithms for inference on multi-label data incurs model mismatch and therefore yields biased parameters.

- In cooperation with Mario Frank, we have developed the method of multi-assignment clustering for Boolean data. In experiments on synthetic data, we have observed that the proposed method yields superior parameter accuracy as compared to state-of-the-art methods. On real-world role-mining data, we observe that multi-assignment clustering outperforms the other methods in terms of the ability to predict the permissions of new users. We have extended this model to incorporate business information and thus formulated a probabilistic approach to hybrid role mining.

- We have demonstrated for the first time how the theory of approximation set coding is applied to a real-world problem by using this framework to study variants of the multi-assignment clustering. Doing so, we observe that model selection based on this generic, information-theory based approach yields the same results as model selection based on the specific quality measures for the parameter accuracy and the prediction ability. The results of this section have been jointly achieved with Mario Frank.

Chapter 2

Fundamentals

In the following, we present the generative process which we assume to have produced the observed data in a general form. The main part of the data is explained by a structure represented by a set of K sources. An independent noise process perturbs the pure data. The observed data is thus a mixture of the structure and the noise contribution. Such generative models are widely used in single-label classification and clustering, but have not been formulated in a general form for data which is jointly emitted by several sources. Afterwards, we discuss the advantages and disadvantages of generative models in machine learning and justify the focus onto this approach for data generated by multiple sources. Finally, we introduce the quality measures which we use throughout this thesis to assess the quality of the results.

2.1 Structure Model for Data from Multiple Sources

We assume that the systematic regularities of the observed data are generated by a set $\mathcal{K}$ of K sources. For simplicity, we assume that the sources are numbered from 1 to K, i.e. $\mathcal{K} = \{1, \ldots, K\}$. Furthermore, we assume that all sources have the same sample space Ω. If this assumption is not fulfilled, the task of determining the set of sources which have generated a given observation x becomes easier, as sources which do not contain x in their sample space can be ruled out as single generators of x.

2.1.1 Source Emissions and Source Set

We assume that each source k emits samples $\Xi_k \in \Omega$ according to a given probability distribution $P(\Xi_k|k)$. We restrict ourselves to parametric probability distributions $P(\Xi_k|\theta_k)$, where θ_k is the parameter tuple of source k. Realizations of the random variables Ξ_k are denoted by ξ_k. Note that both the parameters θ_k and the emission Ξ_k can be vectors. In this case, $\theta_{k,1}, \theta_{k,2}, \ldots$ and $\Xi_{k,1}, \Xi_{k,2}, \ldots$, denote different components of these vectors, respectively.

Emissions of different sources are assumed to be independent of each other. The tuple of all source emissions is denoted by $\boldsymbol{\Xi} := (\Xi_1, \ldots, \Xi_K)$, its probability distribution is given by $P(\boldsymbol{\Xi}|\boldsymbol{\theta}) = \prod_{k=1}^{K} P(\Xi_k|\theta_k)$. The tuple of the parameters of all K sources is denoted by $\boldsymbol{\theta} := (\theta_1, \ldots, \theta_K)$.

Given an *observation* $X = x$, the *source set* $\mathcal{L} = \{\lambda_1, \ldots, \lambda_M\} \subseteq \mathcal{K}$ denotes the set of all sources involved in generating X. The set of all possible source sets is denoted by $\mathbb{L}$. If $\mathcal{L} = \{\lambda\}$, i.e. $|\mathcal{L}| = 1$, X is called a *single-source data item*, and X is assumed to be a sample from source λ. On the other hand, if $|\mathcal{L}| > 1$, X is called a *multi-source data item* and is understood as a combination of the emissions of all sources in the source set $\mathcal{L}$. This combination is formalized by the *combination function*

$$\mathrm{c}_\kappa : \Omega^K \times \mathbb{L} \to \Omega \ ,$$

where κ is a set of parameters the combination function might depend on. Note that the combination function $\mathrm{c}_\kappa(\cdot, \mathcal{L})$ only depends on emissions Ξ_k of sources k which are in the source set $\mathcal{L}$ and it is independent of emissions of sources that are not contained in the source set. Alternatively, one could define a set of combination functions indexed by the source set, where each function would only take the emissions of sources in the source set as arguments. For the sake of clarity, we prefer the notation with the source set as second argument of the combination function.

The generative process for a data item under the structure model is illustrated in Figure 2.1. It consists of the following three steps:

1. Draw a source set $\mathcal{L}$ from the distribution $P(\mathcal{L})$.

2. For each $k \in \mathcal{K}$, draw an independent sample Ξ_k from source k according to the distribution $P(\Xi_k|\theta_k)$. Set $\boldsymbol{\Xi} := (\Xi_1, \ldots, \Xi_K)$.

3. Combine the source samples to the observation $X = \mathrm{c}_\kappa(\boldsymbol{\Xi}, \mathcal{L})$.

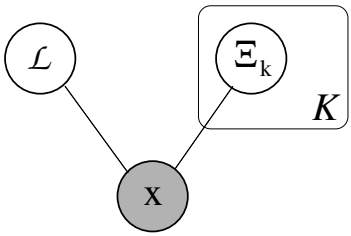


Figure 2.1: The generative model for an observation X with source set $\mathcal{L}$ under the structure model. An independent sample Ξ_k is drawn from each source k according to the distribution $P(\Xi_k|\theta_k)$. The source set $\mathcal{L}$ is sampled from the source set distribution $P(\mathcal{L})$. These samples $\boldsymbol{\Xi} := (\Xi_1, \ldots, \Xi_K)$ are then combined to observation X by the combination function $\mathrm{c}_\kappa(\boldsymbol{\Xi}, \mathcal{L})$. Note that the observation X only depends on emissions from sources contained in the source set $\mathcal{L}$.

2.1.2 The Combination Function

The combination function describes how emissions of possibly several sources are combined to the structure component of the observation X. For source sets of cardinality one, the value of the combination function is the value of the emission of the corresponding source, i.e.

$$\mathrm{c}_\kappa(\boldsymbol{\Xi}, \{\lambda\}) = \Xi_\lambda \ . \tag{2.1}$$

For source sets with more than one source, the combination function can be either deterministic or stochastic. Examples for deterministic combination functions are the sum and the Boolean OR operation. In this case, the value of X is completely determined by $\boldsymbol{\Xi}$ and $\mathcal{L}$. In terms of probability distribution, a deterministic combination function corresponds to a point mass at $X = \mathrm{c}_\kappa(\boldsymbol{\Xi}, \mathcal{L})$:

$$P(X|\boldsymbol{\Xi}, \mathcal{L}) = 1_{\{X = \mathrm{c}_\kappa(\boldsymbol{\Xi}, \mathcal{L})\}} \ . \tag{2.2}$$

Stochastic combination functions can be very diverse and allow us e.g. to formulate the well-known *mixture discriminant analysis* as a multi-source problem.

Mixture Discriminant Analysis as Multi-Label Problem. *Linear Discriminant Analysis* (LDA) [45, 91] finds a linear combination of features

which separates two or more classes of objects. This method can be viewed as a prototype classifier: Each class is represented by a prototype, for which the centroid has to be estimated based on the training data. Based on this idea, and assuming that all classes have equal variance, linear discriminant functions between different classes can be inferred.

In *Mixture Discriminant Analysis* (MDA) [55], this idea is generalized insofar that several prototypes per class are allowed. For simplicity, assume that each of the K classes consists of M prototypes, which all have equal variance. With probability $\pi_{k,m}$, $\sum_m \pi_{k,m} = 1$ for $m = 1, \ldots, M$, a sample X_n with label k comes from the m^{th} prototype of class k. Alternatively, X_n can be considered as belonging to all prototypes of class k, where "belonging to" means "was possibly generated by", with the degree of possibility parameterized by $\boldsymbol{\pi}_k := (\pi_{k,1}, \ldots, \pi_{k,M})^T$. In this setting, the label set of X with source k is translated into the source set $\mathcal{L} = \{(k,1), (k,2), \ldots, (k,M)\}$.

From a generative point of view, the model underlying MDA corresponds to drawing a sample $\Xi_{(k,m)}$ from each of the $K \cdot M$ prototypes and then passing the tuple $\boldsymbol{\Xi}$ of all emissions and the source set $\mathcal{L}_n$ to the combination function $\mathrm{c}_\kappa(\boldsymbol{\Xi}, \mathcal{L}_n)$, which is parameterized by $\kappa = (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_M)$. The stochastic behavior of the combination function is then described by

$$P(\mathrm{c}_\kappa(\boldsymbol{\Xi}, \mathcal{L}_n) = \Xi_{(k,m)}) = \begin{cases} \pi_{k,m} & \text{if } (k,m) \in \mathcal{L}_n \\ 0 & \text{if } (k,m) \notin \mathcal{L}_n \end{cases}$$

Stochastic combination functions render inference more complex, as a description of the stochastic behavior of the function (the parameter vectors $\boldsymbol{\pi}_k$ for $k = 1, \ldots, K$ in the example of MDA) has to be learned in addition to the parameters of the source distributions. In the considered applications, deterministic combination functions suffice to model the assumed generative process. For this reason, we will not further discuss probabilistic combination functions in this thesis.

2.1.3 Probability Distribution for Structured Data

Given the assumed generative process, the probability of an observation X given the source set $\mathcal{L}$ and the parameters $\boldsymbol{\theta}$ is given by

$$P(X|\mathcal{L}, \boldsymbol{\theta}) = \int P(X|\boldsymbol{\Xi}, \mathcal{L}) \, \mathrm{d}P(\boldsymbol{\Xi}|\boldsymbol{\theta}) \; . \tag{2.3}$$

We refer to $P(X|\mathcal{L}, \boldsymbol{\theta})$ as the *proxy distribution* of observations with source set $\mathcal{L}$. Note that in the presented interpretation of multi-source data, the

distributions $P(X|\mathcal{L}, \boldsymbol{\theta})$ for all source sets $\mathcal{L}$ are derived from the single source distribution according to Equation 2.3.

To get a full generative model, we introduce $\pi_{\mathcal{L}}$ as the probability of source set $\mathcal{L}$. The overall probability of a data item $(X, \mathcal{L})$ under the structure model is then given by

$$P(X, \mathcal{L}|\boldsymbol{\theta}) = P(\mathcal{L}) \cdot \int \cdots \int P(X|\boldsymbol{\Xi}, \mathcal{L}) \, \mathrm{d}P(\Xi_1|\theta_1) \cdots \mathrm{d}P(\Xi_K|\theta_K) \quad (2.4)$$

Several samples from the generative process are assumed to be independent and identically distributed (*i.i.d.*). The probability of N observations $\boldsymbol{X} = (X_1, \ldots, X_N)$ with source sets $\boldsymbol{\mathcal{L}} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$ is thus the product of the probabilities of the single samples:

$$P(\boldsymbol{X}, \boldsymbol{\mathcal{L}}|\boldsymbol{\theta}) = \prod_{n=1}^{N} P(X_n, \mathcal{L}_n|\boldsymbol{\theta}) \; . \quad (2.5)$$

The assumption of *i.i.d.* data items allows us a substantial simplification of the model but is not a requirement for the generative model presented in this thesis.

2.2 Noisy Data from Multiple Sources

In addition to the structure mode, an independent noise process might influence the observed data. Analogous to the structure part, the noise process is described by a probabilistic model. We do not distinguish between aleatory randomness (which is an intrinsic property of the observed process) and epistemic randomness, the stochastic behavior of the measurement device including effects of finite precision in the computer.

Besides the constraint that noise emissions must have the same sample space as the structure model, all noise distributions are possible. For sound recordings, Gaussian white noise is often chosen to model the random fluctuations in the signal. For discrete data, the Bernoulli and binomial distributions are popular assumptions for the noise distribution. Formally, we denote the probability distribution according to which the unstructured emissions are drawn by

$$P^U(X^U|\theta^U) \; .$$

When necessary, we use the upper index U to denotes variables, parameters and distributions of the unstructured part, and the upper index S to

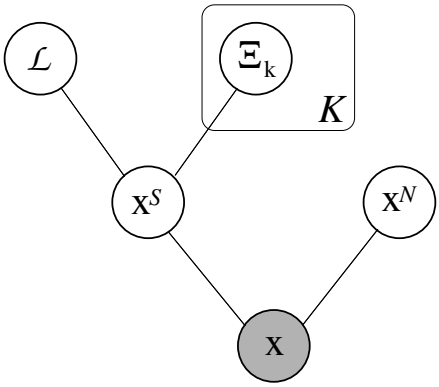


Figure 2.2: The combination function $c_\kappa^S(\boldsymbol{\Xi}, \mathcal{L})$ combines the source emissions $\boldsymbol{\Xi} := (\Xi_1, \ldots, \Xi_K)$ and the source set $\mathcal{L}$ to the structure part $X^S = c_\kappa^S(\boldsymbol{\Xi}, \mathcal{L})$. The unstructured part X^U is generated by P_U. The two parts are then combined to the observation $X = c_\kappa^M(X^S, X^U)$ by a second combination function c_κ^M.

denote the signal part. The upper index M denotes the properties of the complete model, namely X^M is the (observed) random variable generated by the entire model.

Overall Model for Noisy Data

We use a second combination function to describe the combination of the emissions X^S from the structure model and the emissions X^U generated by the unstructured model. Denoting the emission of the overall model by X^M, we have

$$X^M = c_{\kappa^M}^M(X^S, X^U) \ , \tag{2.6}$$

where κ^M denotes possible parameters this function. This combination is assumed to be a deterministic function given X^S and X^N. The probability of the (observed) X^M given the source set $\mathcal{L}$ is the integral over the probability of all pairs of structured and unstructured emissions that are

combined to the observed value, i.e.

$$P^M(X^M, \mathcal{L}|\boldsymbol{\theta}^S, \boldsymbol{\theta}^U, \mathrm{c}^S, \mathrm{c}^M) = \int\!\!\int_\Omega 1_{\{\mathrm{c}^M_{\kappa^M}(X^S, X^U) = X^M\}} \,\mathrm{d}P^S(X^S, \mathcal{L}|\boldsymbol{\theta}^S, \mathrm{c}^S)\,\mathrm{d}P^U(X^U|\boldsymbol{\theta}^U) \tag{2.7}$$

The indicator function $1_{\{\mathrm{c}^M_{\kappa^M}(X^S,X^U)=X^M\}}$ codes the condition that the combination function $\mathrm{c}^M_{\kappa^M}$ deterministically maps X^S and X^N to X^M.

Several samples from the generative process are assumed to be independent of each other. The probability of $\mathbf{X} = (X_1, \ldots, X_N)$ with corresponding source sets $\boldsymbol{\mathcal{L}} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$ is thus the product over the probability distributions of the individual pairs $(X_n, \mathcal{L}_n)$, for $n = 1, \ldots, N$:

$$P^M(\mathbf{X}^M, \boldsymbol{\mathcal{L}}|\boldsymbol{\theta}^S, \boldsymbol{\theta}^U, \kappa^S, \kappa^M) = \prod_{n=1}^{N} P^M(X^M_n, \mathcal{L}_n|\boldsymbol{\theta}^S, \boldsymbol{\theta}^U, \kappa^S, \kappa^M) \tag{2.8}$$

We develop an explicit noise model for unsupervised learning on Boolean data (Chapter 9). For the classification of acoustic data, we limit ourselves to model the structure.

2.3 Generative Models: Assets and Drawbacks

Generative models assume that the observed data is drawn from a probability distribution. The goal of the inference procedure is to determine this unknown probability distribution based on a number of data items. To simplify the inference task, parametric distributions are often presumed. With this hypothesis on the type of the distribution, inferences reduces to estimating the parameters of the distribution.

The generative process is usually modeled as a two-step procedure: First, a set of sources $\mathcal{L}$ involved in the generation of the data item X is drawn from the distribution $P(\mathcal{L})$. Second, the observation X is drawn from the distribution $P(X|\mathcal{L}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ are the parameters of the probability distribution. The joint probability of the feature vector X and the label set $\mathcal{L}$ is given by $P(X, \mathcal{L}|\boldsymbol{\theta}) = P(\mathcal{L}) \cdot P(X|\mathcal{L}, \boldsymbol{\theta})$. Our hypothesis on the generative process responsible for the data is detailed in the previous sections.

In classification, both the observation X and the source set $\mathcal{L}$ are provided in the training data. The focus of the learning task is to predict the source set of a new observation X^{new}. In the clustering task, no source sets

are available, and the main goal is to assign data items into several groups. Following a common convention, the source set $\mathcal{L}$ is called *label set* in the context of classification and *assignment set* for clustering.

Generative models define only one approach to machine learning problems. For classification, *discriminative models* do not infer a prior distribution of the sources and only infer the conditional distribution of data given the sources. A further reduction in complexity is obtained by *discriminant functions*, which map a data item directly to a set of classes or clusters [56, 54]. In clustering, aggregate techniques are examples of purely distance-based clustering methods without an underlying generative model [62].

For both supervised and unsupervised tasks, generative models are the most demanding of all alternatives. If the only goal is to classify or cluster data in an easy setting, designing and inferring the complete generative model might be a wasteful use of resources and demand excessive amounts of data. However, namely in demanding scenarios, there exist well-founded reasons for generative models [10]:

Generative Description of Data. Even though this may be considered as stating the obvious, we emphasize that assumptions on the generative process underlying the observed data can in general only be incorporated in a generative model. This is particularly important when an observation is generated by several sources. Under a purely discriminative viewpoint, knowledge about the generative process can typically not be taken into account, and alternative approaches such as the reduction to a task where data comes from a single source have to be employed. Such reduction techniques for classification are discussed in Section 5.1.1 for classification and in Section 9.3 for clustering.

Interpretability. The nature of multi-source data is best understood by studying how such data are generated. In all applications we consider in this thesis, the sources in the generative model come with a clear semantic meaning. Determining their parameters is thus not only an intermediate step to the final goal of classification or clustering, but an important piece of information on its own.

Consider the cocktail party problem, where several speech and noise sources are superposed to the speech of the dialogue partner. Identifying the sources which generate the perceived signal is a demanding problem. The final goal, however, goes even further, as we are ultimately interested in the contributions of each of the sources, or, more

specifically, in finding out what our dialogue partner said. A generative model for the sources present in the current acoustic situation enables us to determine the most likely emission of each source given the complete signal. This approach, referred to as *model-based source separation* [58], critically depends on a reliable source model.

Reject Option and Outlier Detection. Taking a generative approach allows us to identify cases where no confident assignment of a data item to a set of sources is possible. In such cases, the model can reject to deliver a doubtful assignment and instead mark the observation as not clearly assignable. Following this strategy helps us to reduce the number of wrongly or inconsistently assigned data items.
Given a generative model, we can also determine the probability of a particular data item. Samples with a low probability are called *outliers*. Their generation is not confidently represented by the generative model. Hence, the assignment to a set of sources might be uncertain in such cases, even if a particular set of sources is clearly more probable than all other sets. Furthermore, outlier detection might be helpful in the overall system in which the machine learning application is integrated: Outliers may be caused by defective measurement device or by fraud.

Since these advantages of generative models are prevalent in the considered applications, we restrict ourselves to generative methods when comparing our approaches with existing techniques.

2.4 Evaluation Criteria

In this section, we describe the evaluation criteria used in this thesis to assess results in different sections. We emphasize the need for a quality measure which is well-adapted to the problem at hand — after all, only a well-designed evaluation measure allows us to adequately compare different solutions. For the sake of completeness, we also discuss some quality measures that will not be used in this work.

For both problems in the focus of this thesis, namely multi-label classification and multi-assignment clustering, no unique, generally accepted quality measure exists, as this is the case e.g. for single-label classification or regression. For this reason, we use several measures to assess the quality of experimental results.

Type	Classification	Clustering
estimation accuracy	$a_p(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ $MSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ $RMS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$	$a_p(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ $MSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$ $RMS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})$
reconstruction error	$[ERR(\hat{\boldsymbol{\mathcal{L}}}^{train}, \boldsymbol{\mathcal{L}}^{train})]$ $[BER(\hat{\boldsymbol{\mathcal{L}}}^{train}, \boldsymbol{\mathcal{L}}^{train})]$ $[prec(\hat{\boldsymbol{\mathcal{L}}}^{train}, \boldsymbol{\mathcal{L}}^{train})]$ $[rec(\hat{\boldsymbol{\mathcal{L}}}^{train}, \boldsymbol{\mathcal{L}}^{train})]$ $[F(\hat{\boldsymbol{\mathcal{L}}}^{train}, \boldsymbol{\mathcal{L}}^{train})]$	$\Delta_p(\hat{\boldsymbol{x}}^{(1)}, \boldsymbol{x}^{(1)})$ $[\nu\text{-cov}(\hat{\boldsymbol{x}}^{(1)}, \boldsymbol{x}^{(1)})]$
generalization	$ERR(\hat{\boldsymbol{\mathcal{L}}}^{test}, \boldsymbol{\mathcal{L}}^{test})$ $BER(\hat{\boldsymbol{\mathcal{L}}}^{test}, \boldsymbol{\mathcal{L}}^{test})$ $prec(\hat{\boldsymbol{\mathcal{L}}}^{test}, \boldsymbol{\mathcal{L}}^{test})$ $rec(\hat{\boldsymbol{\mathcal{L}}}^{test}, \boldsymbol{\mathcal{L}}^{test})$ $F(\hat{\boldsymbol{\mathcal{L}}}^{test}, \boldsymbol{\mathcal{L}}^{test})$	$G_p(\hat{\mathbf{u}}^{(1)}, \boldsymbol{x}^{(2)})$ $stab(\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)})$

Table 2.1: Overview of performance measures for classification and clustering. The criteria listed in square brackets are listed for the sake of completeness and will not be applied in this thesis.

We distinguish three different types of measures, summarized in Table 2.1: Model-based performance measures compare the estimated source parameters with the true source parameters. These criteria are applicable to both supervised and unsupervised learning and widely used in statistics [76]. Quality assessment based on the reconstruction error is widespread in clustering [79]. The corresponding training errors of classification are rarely used due to the danger of overfitting and a correspondingly too optimistic evaluation of the classifier. A second, disjoint test data set allows us to accurately estimate the ability of the classifier to generalize the inferred classification rule to previously unseen data. The requirement for a solution to generalize onto new data has only recently become popular in the clustering community. More specifically, a clustering solution should be stable under resampling [75], and the inferred model should be able to explain new data [105]. This measure actually blurs the traditionally strict separation between classification (the goal is to predict labels for new data) and clustering (aiming at grouping the given data).

To obtain a generic description, we assume the parameters of source k to be D-dimensional vectors, i.e. $\theta_k = (\theta_{k,1}, \ldots, \theta_{k,D})$. The corresponding estimators are denoted by $\hat{\theta}_k = (\hat{\theta}_{k,1}, \ldots, \hat{\theta}_{k,D})$. The noise parameters are denoted by θ^N. Furthermore, the reconstruction of the $N \times D$-dimensional data matrix $\mathbf{x}$ by the model, given the inferred parameters, is denoted by $\hat{\mathbf{x}}$. The assignment of an observation x_n to a set of sources is coded either in the source set $\mathcal{L}_n$ or with a binary vector $\mathbf{z}_{n,\cdot} \in \{0,1\}^K$, with

$$z_{n,k} = \begin{cases} 1 & \text{if } k \in \mathcal{L}_n \\ 0 & \text{if } k \notin \mathcal{L}_n \end{cases} \tag{2.9}$$

We will use the notation with the source set $\mathcal{L}_n$ and notation with the indicator vector $\mathbf{z}_{n,\cdot}$ in parallel.

2.4.1 Model-Based Performance Measures

The most direct way to measure for the inference quality is to compare the true source parameters with the estimators obtained from a particular inference technique [76]. Such a direct comparison is typically only possible for experiments with synthetically generated data. The possibility to directly assess the inference quality and the extensive control over the experimental setting are actually the main reasons for experiments on synthetic data.

ℓ_p Accuracy a_p. The ℓ_p accuracy is defined as the p-norm between the true and the estimated parameter vector averaged over all sources k and over all dimensions d. In classification, the numbering of the sources is fixed, such that we can directly compute this measure as:

$$a_p(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) := \frac{1}{K \cdot D} \sum_{k=1}^{K} \sqrt[p]{\left|\left|\theta_{k,\cdot} - \hat{\theta}_{k,\cdot}\right|\right|^p} \tag{2.10}$$

The value of p is chosen depending on the data type. We use the Euclidian norm ($p = 2$) for continuous data and the Hamming distance ($p = 0$) when working with Boolean data. Furthermore, we use the Manhattan norm ($p = 1$) when Boolean values are approximated with continuous variables. Note that for Boolean data, the Hamming distance, the Manhattan norm and the Euclidian norm are identical.

In clustering, we can not assume that the estimated sources are numbered in the same order as the sources in the generative process. To account

for the arbitrary numbering of clusters, we permute the estimated centroids $\hat{\theta}_{k,\cdot}$ with a permutation π such that the estimated and the true centroids agree best:

$$a_p(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) := \frac{1}{K \cdot D} \min_{\pi \in \mathfrak{S}_K} \sum_{k=1}^{K} \sqrt[p]{\left|\left|\theta_{k,\cdot} - \hat{\theta}_{\pi(k),\cdot}\right|\right|^p} \,. \tag{2.11}$$

$\mathfrak{S}_K$ denotes the set of all permutations of K elements. The optimal permutation can be found efficiently using the Hungarian algorithm [74].

Mean Square Error MSE**.** The mean square error is defined as the average squared distance between the true parameter $\boldsymbol{\theta}$ and its estimator $\hat{\boldsymbol{\theta}}$:

$$MSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) := \frac{1}{K} \sum_{k=1}^{K} \mathbb{E}_{\hat{\theta}_k} \left[\left|\left| \theta_{k,\cdot} - \hat{\theta}_{\pi(k),\cdot} \right|\right|^2 \right] \,. \tag{2.12}$$

The MSE can be decomposed as follows:

$$MSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) = \frac{1}{K} \sum_{k=1}^{K} \left(\mathbb{E}_{\hat{\theta}_k} \left[\left|\left| \theta_{k,\cdot} - \hat{\theta}_{\pi(k),\cdot} \right|\right| \right]^2 + \mathbb{V}_{\hat{\theta}_k} \left[\hat{\theta}_k \right] \right) \tag{2.13}$$

The first term, $\mathbb{E}_{\hat{\theta}_k} \left[\left|\left| \theta_{k,\cdot} - \hat{\theta}_{\pi(k),\cdot} \right|\right| \right]$, is the expected deviation of the estimator $\hat{\theta}_{\pi(k),\cdot}$ from the true value $\theta_{k,\cdot}$, called the *bias* of the estimator. The second term, $\mathbb{V}_{\hat{\theta}_k} \left[\hat{\theta}_k \right]$ indicates the *variance* of the estimator over different data sets. We will rely on this *bias-variance decomposition* when computing the asymptotic distribution of the mean-squared error of the estimators.

Root Mean Square Error RMS**.** The root mean square error is defined as the square root of the MSE:

$$RMS(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta}) := \sqrt{MSE(\hat{\boldsymbol{\theta}}, \boldsymbol{\theta})} \,. \tag{2.14}$$

This quality measure indicates the average Euclidian distance between the parameter estimate $\hat{\theta}_k$ and the true value θ_k of the parameter. We will use this criterion to measure the accuracy of parameter estimators.

The noise parameters of the models consists of a few scalar values. We will compare these estimators directly with the true parameter values.

true classification	estimated classification $k \in \hat{\mathcal{L}}_n$	estimated classification $k \notin \hat{\mathcal{L}}_n$
$k \in \mathcal{L}_n$	true positive	false negative
$k \notin \mathcal{L}_n$	false positive	true negative

Table 2.2: Contingency table for a base label k for a data item x_n with true label set $\mathcal{L}_n$ and estimated label set $\hat{\mathcal{L}}_n$.

2.4.2 Performance Measures for Multi-Label Classification

The error rate and the balanced error rate are measures adapted from single-label classification. The quality measures *precision*, *recall* and *F-score* are inspired by information retrieval [107]. These measures are computed based on the number of true positives, true negatives, false positives and false negatives, as defined in Table 2.2. For a source k, estimated source sets $\hat{\boldsymbol{\mathcal{L}}} = (\hat{\mathcal{L}}_1, \ldots, \hat{\mathcal{L}}_N)$ and true source sets $\boldsymbol{\mathcal{L}} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$, let $tp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$, $fn_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$, $fp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$ and $tn_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$ denote the number of true positives, true negatives, false positives and false negatives.

Error Rate (ERR): The error rate is the number of wrong label sets divided by the number of data items:

$$ERR(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{1}{N} \sum_{n=1}^{N} 1_{\{\hat{\mathcal{L}}_n \neq \mathcal{L}_n\}} \ . \tag{2.15}$$

If the true label sets are not uniformly distributed, the error rate is dominated by observations with frequent label sets. For example, if 90% of the data items have the label set $\{1\}$ and only 10% are labeled $\{2\}$, a trivial classifier which assigns the label set $\{1\}$ to all data items obtains an error rate of only 10%.

Balanced Error Rate (BER): The balanced error rate is the ratio of incorrectly classified samples per label set, averaged over all label sets:

$$BER(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{1}{|\mathbb{L}|} \sum_{\mathcal{L} \in \mathbb{L}} \frac{\sum_n \left(1_{\{\hat{\mathcal{L}}_n = \mathcal{L}\}} 1_{\{\mathcal{L}_n = \mathcal{L}\}} \right)}{\sum_n 1_{\{\mathcal{L}_n = \mathcal{L}\}}} \tag{2.16}$$

The balanced error rate gives equal weight to all possible label sets. If the label sets are uniformly distributed, the balanced error rate corresponds to the error rate.

Precision ($prec$): The precision of class k is the fraction of data items correctly identified as belonging to k, divided by the number of all data items identified as belonging to k:

$$prec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{tp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})}{tp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) + fp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})} \ . \tag{2.17}$$

A high precision indicates that most of the data items assigned to class k do actually belong to class k.

Recall (rec): The recall for a class k is the fraction of instances correctly recognized as belonging to this class, divided by the number of instances which belong to class k:

$$rec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{tp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})}{tp_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) + fn_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})} \tag{2.18}$$

Hence, a high recall indicates that most of the observations belonging to a class k are recognized as such.

F-score (F): Good performance with respect to either precision or recall alone can be obtained by either very conservatively assigning data items to classes (leading to typically small label sets and a high precision, but a low recall) or by attributing labels in a very generous way (yielding high recall, but low precision). The F-score, defined as the harmonic mean of precision and recall, finds a balance between the two measures:

$$F_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{2 \cdot rec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) \cdot prec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})}{rec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) + prec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})} \tag{2.19}$$

Precision, recall and the F-score are determined individually for each

base label k. We report the average over all labels k:

$$prec(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{1}{K} \sum_{k=1}^{K} prec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$$

$$rec(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{1}{K} \sum_{k=1}^{K} rec_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$$

$$F(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}}) := \frac{1}{K} \sum_{k=1}^{K} F_k(\hat{\boldsymbol{\mathcal{L}}}, \boldsymbol{\mathcal{L}})$$

The error rate and the balanced error rate are quality measures computed on an entire data set. All these measures take values between 0 (worst) and 1 (best).

All quality measures presented for multi-label classification can be computed on either the training or the test set. As discussed above, only the results on a previously unseen test set provide an unbiased estimate of the performance of a particular classifier. Unless stated differently, these measures are reported on the test set.

2.4.3 Performance Measures for Clustering

We group the quality measures for clustering into measures for the quality of the reconstruction and measures for the ability of a solution to generalize.

Reconstruction

Two evaluation criteria for the reconstruction of the original data are introduced in this section. From a generative point of view, all these measures are sub-optimal insofar as they might punish a clustering algorithm which is able to correctly infer the structure in spite of some noise in the data. Conversely, exactly reproducing noisy data results in a higher score with respect to these measures.

ℓ_p Distance to Input Data (Δ_p): The overall reconstruction accuracy is computed as the average ℓ_p distance between the original matrix $\mathbf{x}$ and the reconstructed data matrix $\hat{\mathbf{x}}$ over all data items and dimensions:

$$\Delta_p(\hat{\mathbf{x}}, \mathbf{x}) := \sqrt[p]{\frac{1}{N \cdot D} \sum_{n=1}^{N} \sum_{d=1}^{D} |\hat{x}_{n,d} - x_{n,d}|^p} \tag{2.20}$$

Again, we set $p = 1$ for Boolean data and $p = 2$ for continuous data in $\mathbb{R}$. Note that the reconstruction is not balanced with respect to the true value of the data: In a Boolean matrix where only 10% of the entries $x_{n,d}$ are 1, a reconstruction with $\hat{x}_{n,d} = 0$ for all n and d achieves an average hamming distance of only 0.1, even though such a result would clearly be considered as a poor reconstruction.

ν-Coverage (ν-cov): The ν-coverage is a second quality measure for discrete data. It measures the ratio between the number of correctly retrieved elements with value ν and the true number of elements with value ν:

$$\nu\text{-cov} := \frac{|\{(n,d)|\hat{x}_{n,d} = x_{n,d} = \nu\}|}{|\{(n,d)|x_{n,d} = \nu\}|} \ . \tag{2.21}$$

The 1-coverage is a popular measure in role mining (see Section 8.3) referred to as *coverage* in this context. However, this measure has a severe shortcoming: Setting $\hat{x}_{n,d} = 1$ for all n, d, one trivially obtains a 1-coverage of 100%.

Generalization Ability

Stability (*stab*): The stability measure is based on the requirement that a clustering solution obtained on one data set is transferable to a second data set with the same distribution [75]. To quantify the degree to which this requirement is satisfied, two i.i.d. data sets $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are separately clustered to obtain the cluster assignment matrices $\hat{\mathbf{z}}^{(1)}$ and $\hat{\mathbf{z}}^{(2)}$. A classifier $\phi^{(1)}$ is trained on the first data set $\mathbf{x}^{(1)}$, using the cluster assignments $\mathbf{z}^{(1)}$ as labels. For the experiments reported afterwards, we used a nearest neighbor classifier with Hamming distance as a distance measure.

Ideally, the output $\phi^{(1)}(\mathbf{x}^{(2)})$ of the classifier $\phi^{(1)}$ applied to $\mathbf{x}^{(2)}$ corresponds to the clustering solution $\hat{\mathbf{z}}^{(2)}$ for every object in $\mathbf{x}^{(2)}$. Note that a multi-label classifier is needed to assess the stability of a multi-assignment clustering. Furthermore, due to the random numbering of clustering solutions, one has to find the permutation which minimizes the deviation. For single-assignment clustering, the ratio r of inconsistently clustered data item is then defined as

$$r(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) := \frac{1}{N} \min_{\pi \in \mathfrak{S}_K} \left\{ \sum_{n=1}^{N} 1_{\left\{\pi\left(\phi^{(1)}\left(\mathbf{x}^{(2)}_{n,\cdot}\right)\right) \neq \hat{\mathbf{z}}^{(2)}_{n,\cdot}\right\}} \right\} \ . \tag{2.22}$$

Note that $\pi\left(\phi^{(1)}\left(\mathbf{x}^{(2)}_{n,\cdot}\right)\right)$ and $\hat{\mathbf{z}}^{(2)}_{n,\cdot}$ are considered unequal whenever they differ in at least one component.

A ratio of $r = 0$ is trivially obtained when we have only one cluster, as there are no other clusters with which the assignment could be confused[1]. As the number of clusters K increases, it becomes more difficult to obtain a small ratio of inconsistent data items. More precisely, a random assignment of K clusters of equal size yields an inconsistency ratio $r_{rand} = (K-1)/K$. Using r_{rand} as normalization, the stability $stab(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) := 1 - r(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})/r_{rand}$ is defined as the difference between the perfect solution and $r(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})/r_{rand}$. Replacing the number of sources K by the number of possible source sets $|\mathbb{L}|$, the stability of a multi-assignment clustering solution is given by

$$stab(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = 1 - \frac{|\mathbb{L}|}{|\mathbb{L}|-1}\frac{1}{N}\min_{\pi\in P_K}\left\{\sum_{n=1}^{N} 1_{\left\{\pi\circ\left(\phi^{(1)}\left(\mathbf{x}^{(2)}_{n,\cdot}\right)\right)\neq\hat{\mathbf{z}}^{(2)}_{n,\cdot}\right\}}\right\} . \quad (2.23)$$

The normalization with r_{rand} allows us to compare clustering hypothesis with different numbers of clusters.

Test Reconstruction Error (G_p): The test reconstruction error measures to what extent a model inferred on a first data set $\mathbf{x}^{(1)}$ can explain a second dataset $\mathbf{x}^{(2)}$ generated by the same distribution as $\mathbf{x}^{(1)}$. In the task of clustering vectorial data, the inferred model is represented by the centroid estimators $\hat{\mathbf{u}}^{(1)}$. To estimate the test reconstruction error, we disclose a small percentage κ of randomly chosen dimensions $\mathbf{D}^* \subseteq \{1,\ldots,D\}$ of each data item in $\mathbf{x}^{(2)}$ to the model. This subset of dimensions is used to assign each data item to an assignment set such that the ℓ_p-distance between the input and the reconstruction is minimized:

$$\hat{\mathbf{z}}_{n,\cdot} := \arg\min_{\mathbf{z}_n}\left\{\sqrt[p]{\sum_{d\in\mathbf{D}^*}\left|x_{n,d} - \mathbf{z}_{n,\cdot}\otimes\mathbf{u}^{(1)}_{\cdot,d}\right|^p}\right\} . \quad (2.24)$$

Using the ℓ_p distance between the reconstruction and the input data allows us to compute $\hat{\mathbf{z}}_{\cdot,n}$ independently of any model assumptions which are possibly made by the clustering solution. We will set $p = 1$ to determine the

[1] A second trivial solution is obtained when $K = N$, i.e. each data item can be assigned to its own cluster. Since this is a pathological case in which clustering does not make sense, we do not further consider this situation.

generalization error on Boolean data. The estimated assignment $\hat{\mathbf{z}}_{\cdot,n}$ is then used to predict the undisclosed dimensions of x_n. The generalization error on data item n measures the deviation between the true and the predicted values on the undisclosed dimensions:

$$G_p(\hat{\mathbf{u}}^{(1)}, \mathbf{x}_{n,\cdot}) := \frac{1}{D - |\mathbf{D}^*|} \sum_{d \notin \mathbf{D}^*} \left| x_{n,d} - \mathbf{z}_{n,\cdot} \otimes \hat{\mathbf{u}}^{(1)}_{\cdot,d} \right| \ . \tag{2.25}$$

The generalization error over all data items in $\mathbf{x}^{(2)} = (\mathbf{x}^{(2)}_{1,\cdot}, \ldots, \mathbf{x}^{(2)}_{N,\cdot})$ is defined as the average over all data items:

$$G_p(\hat{\mathbf{u}}^{(1)}, \mathbf{x}^{(2)}) := \frac{1}{N} \sum_{n=1}^{N} G_p(\hat{\mathbf{u}}, \mathbf{x}^{(2)}_{n,\cdot}). \tag{2.26}$$

Furthermore, the calculations are repeated several times for different choices of $\mathbf{D}^*$ in order to average out the effect of the random choice of $\mathbf{D}^*$.

Note that both data sets $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are influenced by noise. A perfect generalization of $\mathbf{x}^{(2)}$ with $G = 0$, i.e. a complete reconstruction of the new data with the cluster centroids representing the estimated structure, is thus not desired and not possible. However, since the noise in $\mathbf{x}^{(2)}$ is assumed to be independent of the noise in $\mathbf{x}^{(1)}$, a generalization error (measured on $\mathbf{x}^{(2)}$) similar to the reconstruction error (measured on $\mathbf{x}^{(1)}$) is a strong indication that found the structure of the data generation process. On the contrary, a generalization error that clearly exceeds the reconstruction error indicates that the model is overfit on the first data set $\mathbf{x}^{(1)}$.

Part I

Supervised Learning

Chapter 3

Introduction

Data classification, the problem of assigning each data point to a set of categories or classes, is the presumably best studied machine learning problem, but it remains a challenge. Dichotomies or binary classifications distinguish between two classes, whereas multi-class classification denotes the case of several class choices. We denote the number of classes by K.

Multi-label classification characterizes pattern recognition settings where each data point may belong to more than one category. Typical situations where multi-labeled data is encountered are classification of acoustic and visual scenes [11], text categorization [67, 81], and medical diagnosis [69]. For the classification of acoustic scenes, consider for example the well-known Cocktail-Party problem [4], where several signals are mixed together and the objective is to detect the original signal. In text classification, a news report about Sir Edmund Hillary would probably belong to the categories *Sports* as well as to *New Zealand*, and in medical diagnosis, a patient may suffer from several diseases at the same time.

While in single-label classification, a single label indicates to which class an observation belongs, we introduce the *label set* for multi-label classification. This label set contains all classes an item belongs to, e.g. the label set for the imaginary article about Edmund Hillary would most likely be $\{Sports, NewZealand\}$.

In this work, we restrict ourselves to generative models, where each data item is assumed to be generated by one (in the single-label case) or several (in the multi-label case) sources. In the inference phase, the parameters of these sources are estimated based on labeled training data. In the testing

or classification phase, the goal is to determine the set of sources which has produced the given data items.

Despite its significance for a large number of application areas, multi-label classification has only recently received increased attention from the scientific community. Most of the approaches reduce the problem to one or several single-label classification tasks. The trivial approaches for this conceptual simplification either ignore data with multiple labels or consider those items with multiple labels as a new class [11]. More advanced approaches decompose the task into a series of independent binary classification problems, deciding for each of the K classes whether the data at hand belongs to it, and then combine the K classifier outputs to a solution of the original problem. We review these approaches in Chapter 5.

All these approaches have significant drawbacks. The trivial approach mainly suffers from data sparsity, as the number of possible label sets is in $\mathcal{O}\left(K^{M_{\max}}\right)$, where $M_{\max}$ is the maximal degree of the data items. Furthermore, many of these methods can only assign label sets that are present in the training data. The main criticism on the reduction of the multi-label task to a series of binary decision tasks for the confusion between frequent co-occurrence and similar source statistics — in all approaches we are aware of, the more often two sources occur together, the more similar are their estimated statistics. In this way, these methods neglect the information which multi-labeled data contains about all classes in its label set, which deteriorates the source estimates and leads to poor classification performance.

A particular interpretation of multi-label data is taken in [66]: Each training instance is given a set of candidate labels, but only one of the candidate labels is the correct one. However, this learning problem is more difficult than standard multi-class learning, because it is unclear which of the labels is the target, and the source assumed to have generated the present data first has to be estimated.

From a generative point of view, a label $k \in \mathcal{L}$ thus understood as "this data item is possibly generated by source k". In our proposed framework, this understanding is modeled with a probabilistic combination function which chooses a label out of the label set with a certain probability, as exemplified in Section 2.1.2 for mixture discriminant analysis. Note that the generative model also offers a theoretically well-founded approach to determine the target label among the multiple labels. If the combination function in the true generative process consists of selecting an emission from one of the sources in the label set, this model reconstructs the generative

model and is thus appropriate for this type of multi-label problems.

In Chapter 5, we propose a novel approach for the classification of multi-label data, which is inspired by the fundamental physical principle of superposition found in acoustics and other wave phenomena. We assume a source for each class and consider data with multiple labels as an additive mixture of independent samples generated by the respective classes. A deconvolution enables us to estimate the contributions of each source to the observed data point and thus to use multi-label data for inference of the class distributions. Similarly, the distributions of multi-label data are computed based on the source distributions. Doing so, this approach allows us to consistently model jointly occurring single- and multi-label data with a small number of parameters.

The combination function describing superposition is the addition of the respective intensities. In Chapter 6, we generalize the generative approach to further combination functions such as the Boolean OR. Furthermore, we extend the asymptotic theory for estimators inferred on single-label data to multi-label data and use these result to describe the asymptotic behavior of different inference techniques based on their assumed combination function. Finally, in Chapter 7, we proof that a common assumption for inference based on multi-label data implies a model mismatch and consequently biased parameter estimators and sub-optimal classification results.

3.1 Classification of Acoustic Streams

Our research on multi-label classification is motivated by the design of intelligent hearing instruments for hearing impaired persons. To support a comfortable and enjoyable interaction with the environment in spite of a hearing loss, different hearing aid characteristics are desired under different listening conditions. Modern hearing instruments therefore provide several hearing programs to account for different acoustic situations, such as speech, speech in noisy environments, and music. These programs can either be activated by the wearer, namely using a remote control, or they operate automatically. Manual switching is a nuisance for the user since it burdens him with the tasks of recognizing the acoustic situation and switching to the optimally adapted program. Automatic program selection systems are therefore highly appreciated by the users [17] and are provided by most modern hearing instruments.

A reliable estimation of the hearing activity is essential, as the desired

processing of the audio stream depends on the intention of the user: In a dialog situation, the emphasis is clearly on speech intelligibility, and one accepts some distortions of the voice of the dialog partner. However, when listening to music, a natural sound is essential, while intelligibility is less important: Even in an opera, you don't absolutely have to understand the text of an aria.

A fundamental limitation of all automatic program selection systems is already apparent [18]: Even if the hearing instrument is able to correctly identify all sound sources in an acoustic situation, it is not always able to recognize whether the user regards a sound as desired signal or as noise. The situation in a piano bar, where people discuss and music is played in the background, is very similar to a classic concert, where an annoying neighbor never stops talking. Most program selection systems detect the hearing activity therefore in two steps: The sources in the current acoustic situation are identified in the first step. As described above, this task is an example of a multi-label classification problem. In the second step, the users hearing activity is estimated based on the identified sources. This estimation is mostly based on investigations by audiologist and common sense. However, a more precise modeling of a situation may provide valuable additional information to choose the appropriate hearing instrument setting.

Reverberation, on the other hand, is an important phenomenon which entails a temporal and spectral smearing of the sound patterns and hence causes an important loss of speech intelligibility. Detecting the reverberation intensity is therefore a highly relevant problem not only for hearing instruments, but also for mobile and line telephony. The degree of reverberation is measured by the *reverberation time* T_{60}, defined as the time required for reflections of a direct sound to decay by 60dB below the level of the direct sound. The reverberation time is affected by the size and shape of the room as well as the materials used in the construction. Large concrete rooms have a high reverberation time, while small rooms with soft materials have a small reverberation time. In all mentioned applications, a reliable estimation of the reverberation time is a prerequisite to de-reverberate the input signal and thus increase speech intelligibility in all mentioned applications.

In both hearing activity and reverberation time detection, the large variations within the collective sound groups render the two prediction problems hard: The hearing instrument has to be able to determine the hearing activity in basically all possible rooms, and to estimate the reverberation time based on any type of signal.

3.2 Multi-Instance Learning

Multi-instance or multiple-instance learning denotes a learning scenario where less specific labels are available: While in classical learning problems, each instance is labeled, labels are only given to *bags* of instances in the case of multi-instance learning [36].

A classical application for multi-instance learning is image classification [80]. Images are typically labeled globally, i.e. a label such as *tree* is assigned if a tree is contained in the image. However, an image typically consists of several sub-regions, and only one of them, in the considered case, shows a tree. To infer characteristics of trees in images, the learning method thus first has to estimate which part of the image represents a tree. Given this partition, different sub-regions are usually independent of each other in the sense that a tree looks similar regardless of other objects shown in the image.

Images usually have several labels. Correlations between individual labels are conveniently retrieved with a prior over label sets. The generative process can be thought of drawing samples from the different objects in the image (e.g. a tree, a person and a sitting bench) and then positioning these opaque sub-regions to yield the final image. The combination of different source emissions to the observed images is thus described by the juxtaposition. Given this generative process, the multiple labels describe the content of the whole image, while — unless the image contains semi-transparent elements — each segment of the image is described by a single label. Once the different subregions of the images are identified, the classification of the different segments is a single-label task.

3.3 Multi-Task Learning

Multi-task learning describes a setting where a problem is learned simultaneously with other related problems. All learning tasks share a common representation. The restriction to a single representation regularizes the solutions of each single learning task. Namely in scenarios where overfitting would be a problem for a single learning task, e.g. if only a small number of training data is available, the constraint to solve several tasks based on the same representation therefore facilitates a superior classification performance [5]. Typical applications of multi-task learning are image classification and segmentation [24, 117].

We are not aware of any work on multi-task learning where an individ-

ual task consists of multi-label classification. However, we do not see the limitation to single-label tasks as a fundamental one. We conjecture that multi-task learning will be beneficial also for related multi-label problems.

On a theoretical level, the relatedness of several tasks can be defined through a description of the data generating process [7]. This formal notion of similarity between tasks renders the derivation of generalization bounds possible and implies general conditions under which multi-task learning is beneficial in comparison to single-task learning.

Chapter 4

Features for Audio Classification

A succinct set of features is crucial for all classification tasks. This chapter treats the design and evaluation of features for audio-classification in hearing instruments. In this application, the detection of the predicted hearing activity determines the processing of the sound signal. A reliable set of features is therefore essential to achieve a good overall performance of the hearing instrument. We introduce a set of sound field indicators (SFI) which measure basic properties of the sound field. These indicators are proposed as a novel feature set for this critical classification task.

We investigate the prediction of two important specifications of a sound clip: The *hearing activity* and the *reverberation time*. Each sound clip is assigned to one of the five acoustic situations *clean speech*, *speech in low noise* (SNR 5dB or better), *speech in high noise* (SNR 2dB or worse), *noise*, and *music*. By assumption, the corresponding hearing activity is to follow the speech whenever a speech source is present, otherwise to enjoy the music, and no active listening in the case where only noise is present. Furthermore, the reverberation time is given for each of the sound clips.

The indication on the intensity of the noise combined with the speech signal is important to operate actuators such as the noise canceler at the appropriate intensity. The *signal to noise ratio* (SNR) is used to quantify to what extent the signal is corrupted by noise. This measure is defined as the ratio between the power of the signal and the power of the noise and usually measured in decibel (dB): $\mathrm{SNR} := 10 \log_{10} (P_{\mathrm{signal}}/P_{\mathrm{noise}})$. The lower the

SNR, the more corrupted is the signal by noise.

In this chapter, we focus on the features themselves and evaluate the performance of different feature sets with standard techniques for classification (to predict the hearing activity) and regression (when estimating the reverberation time). Generative approaches for reverberation time estimation [92, 71, 70] explicitly model the reverberation process in order to estimate the reverberation time. They are based on longer frames and typically also imply a significant computational load.

4.1 Feature Sets

In this section, we introduce the feature sets which are evaluated afterwards with respect to their suitability to predict the hearing activity and the reverberation time. We first present two novel feature sets, the sound-field indicators (SFI) and the short-time statistics over the sound-field indicators (SFIst) and then review three sets of well-known and widely used features.

Note that all features are computed on frames of length $\lambda = 0.8$sec. This time constant is a trade-off between the reliability of the feature estimation and the time delay of the hearing instrument when settings are changed.

4.1.1 Sound Field Indicators

The sound field indicators (SFI) are computed on four-channel recordings, with channels denoted $x_1, \ldots, x_4$. The corresponding signals in the frequency domain are denoted by X_1 to X_4. X^* indicates the complex conjugate of X. We compute the (cross) power-spectrum densities $P_{ij} = X_i \cdot X_j^*$ between any two channels using Welch's averaged modified periodogram method of spectral estimation with 128 point segments, 75% overlap and Hanning window [86]. This yields 65 non-redundant Fourier coefficients at equidistant frequencies. We propose to compute the following indicators for each frequency:

`Power Spectral Densities:` The power spectral densities P_{11}, P_{22}, P_{33} and P_{44} of the four channels.

`Cross Power Spectral Densities:` The cross-power spectra between the front and back microphone on both sides (i.e. P_{12} and P_{34}) and between the left and right side in the front and back half (i.e. P_{13} and P_{24}).

`Mean Power Density:` The mean pressure on both sides of the head, computed as $P_l = X_l \cdot X_l^*$ with $X_l = (X_1 + X_2)/2$ for the left side, and similar for the right side for P_r.

`Transfer function:` The transfer function between two microphones is the cross-power spectral density of the microphones divided by the auto-power spectral density of the reference microphone. We use the microphone with the smaller number as reference and thus get $G_{ij} = P_{ij}/P_{ii}$ with $i < j$.

`Coherence:` The coherence measures the similarity between two channels, computed as $C_{ij} = |P_{ij}|^2/(P_{ii}P_{jj})$ with $i < j$.

`Normalized Intensities:` The signal intensity on the left side is defined as $I_{12} = P_l \cdot U_{12}^*$. $U_{12} = -(X_1 - X_2)/(\imath\omega\rho d)$ is the particle velocity, where $\omega = 2\pi f$ is the angular frequency, ρ the air density, d the distance between the two microphones of the hearing instrument and $\imath := \sqrt{-1}$ is the imaginary unit. The *active* intensity is the real part, the *reactive* intensity the imaginary part of the intensity. The intensity values are normalized with the acoustic power of the mean pressure at the left ear, i.e. with $(X_1 + X_2)^2/(4\rho \cdot c)$, where c is the speed of sound. After some algebraic simplifications, we get the following expressions for the normalized active and reactive intensities:

$$Ia_{12}n = \frac{4 \cdot c \cdot \Im(P_{12})}{\omega \cdot d \cdot (P_{11} + P_{22} + P_{12} + P_{12}^*)}$$
$$Ir_{12}n = \frac{2 \cdot c \cdot (P_{22} - P_{11})}{\omega \cdot d \cdot (P_{11} + P_{22} + P_{12} + P_{12}^*)}$$

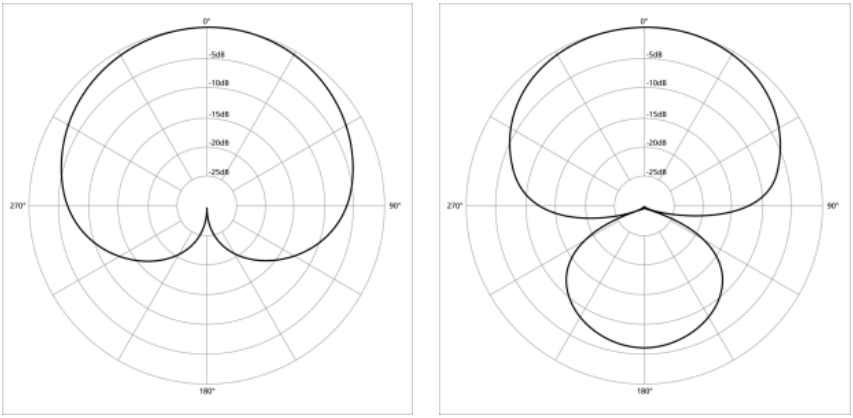
where $\Im(\cdot)$ denotes the imaginary part. The normalized intensities for the right ear are computed analogously.

`Acoustic Impedance:` This indicator measures the relation between pressure and particle velocity, where we use the mean pressure at either side. For the impedance on the left side, we thus have

$$Z_{12} = \frac{P_l}{U_{12}} = \frac{X_1 + X_2}{2} \frac{\imath \cdot \omega \cdot d}{X_2 - X_1},$$

and similar for the right side. For easier interpretation, we normalize Z_{12} by the wave impedance $(\rho \cdot c)$.

`Front Cardioid Directivity:` On either side of the head, a front cardioid characteristic is calculated for the two microphone signals. On the left side,



(a) Front Cardioid (b) Hyper-Cardioid

Figure 4.1: Polar sensitivity patterns of front cardioid and hyper-cardioid. The front cardioid focuses on signals from the front, while signals from the back are suppressed. The hyper-cardioid has a tighter front area than the front cardioid, and an additional small lobe at the back, while signals from the side are suppressed.

$D = e^{-\imath\omega d/c}$ being the delay, we compute

$$FC_{12} = X_1 - D \cdot X_2 \ .$$

Again, the front cardioid characteristic is normalized with the mean pressure: $FC_{12}n = FC_{12}/P_l$. Corresponding computations yield the front cardioid directivity for the right head side. See Figure 4.1(a) for an illustration of the polar sensitivity patterns of the front cardioid.

`Back Cardioid:` Similarly to the front cardioid, the back cardioid represents the beamformer output according to

$$BC_{12} = X_2 - D \cdot X_1$$

and the normalization is as for the front cardioid.

`Fixed Beamformer:` The beamformer aims at acoustically focussing to the front direction by computing a weighted difference between the front and back cardioid. In the fixed beamformer, the weights $\beta^{(f)}$ for this difference are fixed and thus independent of the cardioid intensities. We use the

normalized cardioid intensities, the beamformer result is thus

$$E_{12}^{(f)} = FC_{12}n - \beta^{(f)} \cdot BC_{12}n.$$

with $\beta^{(f)} = 0.3$. Analogous for the right side.

`Adaptive Beamformer:` The weight $\beta^{(opt)}$ of the beamformer defines the direction of maximum suppression. In the adaptive beamformer, the weight is chosen such that sound from the back half-space is maximally suppressed. Unlike the fixed beamformer, the adaptive beamformer is thus not limited to suppressing sounds which come directly from behind. The unrestricted optimal beamformer weights $\tilde{\beta}_{12}$ are computed as

$$\tilde{\beta}_{12} = \frac{\Re\left(FC_{12} \cdot BC_{12}^{*}\right)}{|BC_{12}|^2} \ ,$$

where $\Re(\cdot)$ denotes the real part. Restricting the weights to the suppression of the back hemisphere yields

$$\beta_{12}^{(opt)} = \begin{cases} 0 & \text{if } \tilde{\beta}_{12} < 0 \\ \tilde{\beta}_{12} & \text{if } 0 \leq \tilde{\beta}_{12} \leq 1 \\ 1 & \text{if } \tilde{\beta}_{12} > 1 \end{cases}$$

As features, we use the value of the unrestricted $\tilde{\beta}_{12}$, the optimal beamformer weight $\beta_{12}^{(opt)}$ restricted to the suppression of the back hemisphere, and the beamformer output $E_{12}^{(opt)}$ obtained with the optimal parameter $\beta_{12}^{(opt)}$.

Note that the cross-power spectral densities P_{ij}, the transfer functions G_{ij}, the impedance, the cardioid directivities and the beamformer outputs are complex values. In the following, they are represented using the absolute value (in dB) and the phase (in degrees).

After computing these values for each frequency, the indicators are averaged over 20 Bark bands. Thus, a total of 920 real-valued indicators are extracted out of each frame of the acoustic stream.

4.1.2 SFI Short-Time Statistics

We propose the short-time statistics on the sound-field indicators (SFIst) as a variant of the SFIs which also describes the variability of the indicators over time. To measure these fluctuations, the sound-field indicators are now

calculated on sub-frames of 0.08sec length, thus yielding 10 values of each indicator per 0.8sec frame. The mean and the standard deviation of each of these indicators are then computed. Only the sound field indicators on channels 1 and 2, i.e. on the left side of the head, are used to obtain these summary statistics.

The features obtained as average and standard deviation of feature `F` are denoted by `F_a10` and `F_s10`. The set of all average features is denoted by SFIavg, and SFIstd denotes the set of all standard deviation features. Each of these two feature sets contains 400 features which measure 20 properties on 20 frequency bands. The set of the Sound-Field Indicator Short-Time statistics, denoted SFIst, is the union of SFIavg and SFIstd.

4.1.3 Mel-Frequency Cepstral Coefficients

The Mel-frequency cepstral coefficients (MFCC) are a widely used feature set for speech recognition, as they represent the speech amplitude spectrogram in a compact form [88]. The individual steps in the extraction of these features are motivated by perceptual or computational considerations. They are derived from a cepstral representation of the audio clip with frequency bands chosen to be equally spaced in the Mel scale to support a representation of sound which approximates the human perception.

We use the left-front signal x_1 to determine the MFCCs. The concrete steps are as follows:

Discrete Fourier Transform: Compute the discrete Fourier transform $X_m(s)$ of the input stream x_1 at frame m and frequency s, for $s = 0, \ldots, N_s - 1$, as

$$X_m(s) = \sum_{j=0}^{N_s-1} x_1(j + mN_s) \cdot \exp\left(-\frac{\imath 2\pi jk}{N_s}\right)$$

where N_s is the number of samples per frame. In our setting, with sampling frequency $F = 20'480$Hz and frame length $\lambda = 0.8$sec, we have $N_s = F \cdot \lambda = 16'384$.

Log-Power in Mel Scale: Map the power spectrum onto the Mel scale using triangular overlapping windows and take the logarithm of the power at each of the Mel frequencies. Using non-uniform filter bank responses $H_b(s)$ approximating a triangular shape, the log-energy outputs $LE_m(b)$ for

band b, $b = 1, \ldots, N_b$, is

$$LE_m(b) = \log_{10}\left(\sum_{s=0}^{N_s-1} |X_m(s)| \cdot H_b(s)\right) \ .$$

Cepstral Coefficients: Obtain the Mel-Frequency Cepstral Coefficient of the c^{th} Mel-frequency, $c = 1, \ldots, N_c$, by computing the Discrete-Cosine transform of the log-energy outputs $LE(b)$:

$$MFCC_{m,c} = \sum_{b=1}^{N_b} LE_m(b) \cos\left(c\left(b - \frac{1}{2}\right)\frac{\pi}{N_b}\right) \ .$$

The number of Mel-frequencies is usually $N_c = 13$.

Several variations of the MFCC extraction exist, which vary mainly in the number of bands N_b and in the parametrization of the triangular filters. We use the MFCC implementation provided by Slaney [100] in MATLAB .

4.1.4 Features for Auditory Scene Analysis

The field of auditory scene analysis (ASA) [13] has inspired a series of features which are frequently used in applications of computational auditory scene analysis. We use a set of features that is employed in state-of-the-art hearing instruments to classify sounds into different hearing activities.

Most of these features are based on statistics over several sub-parts of the time window. We briefly mention these features which were extensively described and analyzed in [18, 19]. We refer to this feature set as ASA. All these features are computed on the channel x_1.

`AHWidth`: Width of the amplitude histogram, computed as the difference between the 90% and the 10% percentile.

`CGAV`: Spectral center of gravity, averaged over several sub-windows.

`CGFS`: Variance of the spectral center of gravity.

`LowFreqAbs`: Energy in the lowest frequency bands.

`MeanLev`: Root of mean square level over several sub-windows.

`Onset`: Mean onset strength over all bands in the time frame.

`PitchMean`: Mean value of the pitch in the time frame.

`Tonality`: Ratio between tonal and non-tonal segments in the time frame.

`Spect1, Spect2, Spect3`: Normalized spectral intensities in different frequency ranges.

4.1.5 Features for Music Genre Classification

Further features for sound classification [18, 37] and more specifically for music genre classification [110] are grouped into the feature set which we refer to as MUSIC. Where available, we used the software MARSYAS [109] to extract these features.

`Sample Amplitude Histogram Kurtosis (SAHK)`: This feature measures the skewness of the amplitude histogram. It allows us to discriminate between continuous sounds and sounds which contain alternations between signal and silence.

`Time-Domain Autocorrelation Maximum (TACM)`: The maximum autocorrelation time of the signal. This feature captures repetitions directly in the time domain.

`Beat-Spectrum Spectral Roll-Off (BBSR`α`)`: The spectral roll-off of the beat spectrum at level α is defined as the frequency below which $\alpha\%$ of the magnitude distribution lies. This measure of the spectral shape is computed for $\alpha = 10, 20, \ldots, 90$.

`Beat-Spectrum Threshold Crossings (BSTC)`: The number of spectral peaks with some minimal height. `BSTC` facilitates the distinction between structured clips (with few, high peaks) and unstructured clips.

`Beat-Spectrum Overall Sum (BSSUM)`: The overall sum over the beat spectrum, an indicator for the strength of the beat.

`Relative Amplitude of Highest Peak (BSRA)`: The amplitude of the second peak divided by the amplitude of the second peak.

`Beat-Spectrogram` α`-Percentile (BSPCT`α`)`: A robust way to estimate the height of the highest peak in the beat histogram. This value is computed for $\alpha = 90\%, 95\%$ and 97.5%.

4.2 Classification and Regression Techniques

In this section, we introduce the classification and regression techniques employed to analyze the suitability of the feature sets for prediction of the

reverberation time and the hearing activity. Given a feature vector of a frame, the goal is to predict the hearing activity h or the reverberation time T_{60}. Predicting h is a single-label multi-class classification problem with $K = 5$ classes. To determine the reverberation time, the setting of *regression* is more adequate, since the reverberation time is a continuous value.

We use support vector machines (SVM) and random forests (RF) for classification, and linear models, SVM and RF for regression. These techniques are briefly introduced in the following. Note that linear discriminant analysis (LDA) is not applicable for classification in our setting, as the number of features exceeds the number of observations for all feature sets based on the sound field indicators.

In the following, $\mathbf{x}_n = (x_{n,1}, \ldots, x_{n,D})^T$ denotes the feature vector, and y_n the label of sample n. The number of samples is N.

Linear Regression Models

In a linear regression model, the target variable y_n (in our setting: the reverberation time T_{60}) is related to the vector of independent variables $\mathbf{x}_n$ according to

$$y_n = \mathbf{w}^T \cdot \mathbf{x}_n + w_0 + \epsilon_n \ ,$$

where ϵ_n is a random variable representing the deviations in this relationship. The weight vector $\mathbf{w}$ and the bias w_0 are chosen such that the mean square error, averaged over all training samples, is minimized. Under the assumption that the noise terms ϵ_n are independent and identically distributed according to a Gaussian distribution, this corresponds to a maximum likelihood estimation of the parameters $\mathbf{w}$ and w_0.

Support Vector Machines

Support vector machines are designed based on the idea that data items from different classes should be maximally separated. Assume the separation is done by a plane. Any plane can be represented as the set of points $\mathbf{x}$ which satisfy $\mathbf{w}^T \cdot \mathbf{x} - b = 0$, where $\mathbf{w}$ is a the normal vector, and $b/||\mathbf{w}||$ determines the offset of the plane along the normal vector $\mathbf{w}$ from the origin. In the simplest case of two linearly separable classes, the margin between data from the two classes is the distance between the two planes $\mathbf{w}^T \cdot \mathbf{x} - b = \pm 1$, where $\mathbf{w}$ and b are chosen such that the distance $2/||\mathbf{w}||$ is maximal, under the constraint that both planes still separate the two classes. Thus, for all data

items $\mathbf{x}_n$ with labels $y_n \in \{\pm 1\}$, we have the condition $y_n \cdot (\mathbf{w}^T \cdot \mathbf{x}_n - b) \geq 1$, and the optimization problem can be formulated as

$$\min_{\mathbf{w},b} ||\mathbf{w}|| \quad \text{s.t.} \quad y_n \cdot (\mathbf{w}^T \cdot \mathbf{x}_n - b) \geq 1 \ , \quad n = 1, \ldots, N$$

Writing this optimization problem in the dual form, it becomes apparent that the classification task only depends on the so-called *support vectors*, the training samples on the margin. With $||\mathbf{w}||^2 = \mathbf{w}^T \cdot \mathbf{w}$ and substituting $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$, one derives the optimization problem

$$\max_{\alpha} \tilde{L}(\alpha) \quad \text{s.t.} \quad \alpha_n \geq 0 \quad n = 1, \ldots, N$$

with the Lagrange function

$$L(\alpha) := \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n,m=1}^{N} \alpha_n \alpha_m y_n y_m \cdot \mathbf{x}_n^T \mathbf{x}_m \ .$$

Note that the data samples $\mathbf{x}_n$ only enter via their scalar product $\mathbf{x}_n^T \mathbf{x}_m$. With the so-called *kernel trick*, the scalar product is replaced by a *kernel* $k(\mathbf{x}_n, \mathbf{x}_m)$, corresponding to a transformation of the feature space. We use support vector machines with the following two kernels:

Linear Kernel: The linear kernel is the scalar product between the coordinates of the two data items: $k(\mathbf{x}_n, \mathbf{x}_m) = \mathbf{x}_n^T \cdot \mathbf{x}_m$

Radial Basis Function (rbf) Kernel: The value of the radial basis function only depends on the pairwise distance between the data items: $k(\mathbf{x}_n, \mathbf{x}_m) = \exp\left(-\gamma ||\mathbf{x}_n - \mathbf{x}_m||^2\right)$

In the rbf kernel, γ is a hyperparameter of the model which we optimize via cross-validation (see below). For multi-class classification, the standard approach reduces the problem to a series of binary classification problems [22]. The adaption of the support vector machine for regression [38] employs a risk function which ignores data points which are close to the predicted value and thus predicts the value for a new input based on only a small number of training data.

Random Forests

A random forest [14] is an ensemble method consisting of many (decision) trees. For every tree, a set of N *bootstrap samples* is drawn with replacement

from the training data set. The remaining samples are called the *out of bag* samples. As the sampling is done with replacement, some data items occur multiple times in the bootstrap sample set, while $1/e \approx 36.8\%$ remain in the out-of bag sample set.

The bootstrap samples form the root node of the tree. The tree is grown recursively for each leaf node which is larger than the maximal size: Randomly select a number of features, pick the best out of the chosen features, split the data of the current node into two child nodes, and memorize the split condition.

To predict the value of a sample, it is propagated down to the leaf nodes in all trees according to the stored split conditions. For classification, the label of the new data item is then attributed according to the majority vote over all trees. For regression, the predicted value is the mean of the values of the dependent variable in all leaf notes.

The parameters of random forests are the number of trees in the forest (usually 500) and the number of features tested at reach split (usually one third of all features for regression, and the square root of the number of features for classification). The performance of random forests is remarkably robust with respect to these values [57]. Random forests are specially suited for classification and regression in settings where the number of variables is high compared to the number of samples.

In order to measure the prediction strength of the j^{th} variable, the prediction result on the out of bag samples is compared to the prediction result on the same sample when the values of the j^{th} variable are randomly permuted. These values are averaged over all trees and then used to indicate how much the random forest relies on the values of this feature.

4.3 Evaluation of Feature Sets

In order to be able to precisely control the reverberation times, the evaluation of the different feature sets is done on artificially reverberated sound clips. The original sounds were recorded in a reverberation-free room using a KEMAR dummy head, which simulates the changes that occur to sound waves when they pass a human head and torso. This setup allows us to realistically record sound signals subject to effects such as diffraction and reflection around the human ear.

The sound source is placed straight ahead (at 0°) at a distance of 1.5m. On both sides of the dummy head, a behind-the-ear hearing instrument

with two microphones is mounted to record two channels of the signal. As a result, we obtain four channels corresponding to the microphone positions left-front (x_1), left-back (x_2), right-front (x_3) and right-back (x_4). All recordings are made with a sampling rate of 20'480Hz.

We record sound clips of the three primary hearing activities *clean speech*, *noise* and *music*. The clean speech clips contain recordings from persons of both genders in English and German. The noise recordings contain a variety of typical noise situations including social noise (e.g. in a bar), human body noise (e.g. laughing), office noise (e.g. a printer) and household noise (e.g. a vacuum cleaner). The music recordings contain samples of different instruments as well as recordings of pop and classical concerts. Roughly 20 recordings of each hearing activity are produced for the experiments. Furthermore, the clean speech and noise clips are synthetically mixed at different signal to noise ratios (SNR) to yield recordings with the hearing targets *speech in low noise* (SNR 5dB or more) and *speech in high noise* (SNR 2dB or below). These two hearing targets contain roughly 50 clips each.

The dry sound recordings are artificially reverberated using the professional software Altiverb, which enables realistic simulation of reverberation and is delivered with a wide range of high-quality recordings of impulse responses of real rooms of all sizes. We chose room impulse responses (RIR) with 10 reverberation times T_{60} between 0.52sec and 12.6sec to cover the reverberation characteristics of all rooms a person typically stays in.

To obtain independent samples, we use only one frame per sound clip as sample for the respective hearing activity and regression time. Since some of the ASA features need a settling time of roughly 10 seconds, we use the features extracted on frame number 15, corresponding to the recording time 11.2sec to 12sec. As the values of the features are at different ranges, we have standardized all features setting the average to 0 and the standard deviation to 1. Doing so, we get commensurate features [53], which is important namely for linear discriminant analysis and support vector machines.

After normalization, the samples are split into a training and a hold-out set. The training set is used to infer the model. For models with hyperparameters (such as the cost value in the support vector machines), these parameters are determined using grid-search and cross-validation on the training set. For each hyperparameter, the training set is again divided into 10 sub-sets. In 10 runs, each of these 10 subsets is used once as a test set and 9 times as a training set. The hyperparameters are then set to the value that yielded the best average performance on these test sub-sets.

Then, the model is trained on the whole train set, and the performance on the hold-out set is reported. The random forest is used with the default parameter values and the cross-validation step is therefore omitted. All results reported in the following are performance measures on the hold-out data set.

As the number of samples varies for different hearing activities, we use the average rate of misclassified samples per class, the *balanced error rate* BER, to assess the classification results in a way that takes the different number of samples per hearing activity into account.

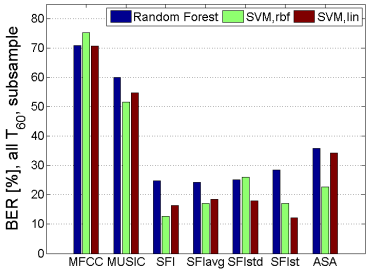
4.3.1 Hearing Activity Classification

We test the different feature sets first independently and then conditioned on the reverberation time.

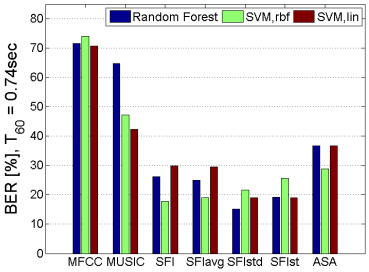
When computing the classification performance over all reverberation times, we have to keep in mind that all sound clips are reverberated with 10 different reverberation times. To get independent samples, we therefore randomly choose a single reverberation time for each sound clip. The balanced error rate for the hearing target classification is depicted in Figure 4.2(a). The four SFI-based feature sets outperform the three other feature sets, namely the linear SVM yields more accurate results on the new feature sets. Differences within the proposed feature sets can be observed depending on the classifier. The performance on the ASA features is below the performance on the SFI-based features for the linear SVM and for the random forest classifier. The two feature sets MFCC and MUSIC seem to be too specifically tailored for a particular application and fail in this overall comparison.

The most important features for prediction of the hearing activity with a random forest are listed in Table 4.1. Note that predominantly the original SFI features and some features from the feature set SFIavg are important for this task. Both classes of features measure the average behavior of the sound-field in the frame, but with different averaging methods. Short-time variations in the sound field indicators (which would be retrieved by the SFIstd features) seem to be of little importance to predict the hearing activity. The important features are measured in the medium and higher frequencies. Furthermore, the decay in importance is relatively slow, there does not exist a group of features which are clearly more important than any other feature.

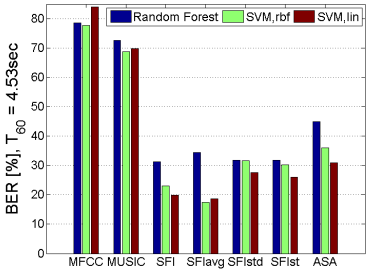
To study the influence of the reverberation time onto the classification



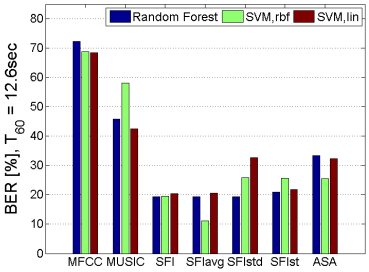
(a) All reverberation times (Subsampled)



(b) Reverberation time $T_{60} = 0.74$sec



(c) Reverberation time $T_{60} = 4.53$sec



(d) Reverberation time $T_{60} = 12.6$sec

Figure 4.2: Classification performance of different classification techniques on the task of predicting the hearing activity. The upper-left panel shows the performance on a data set containing samples from all reverberation times. The balanced error rate on samples with reverberation time 0.74sec, 4.53sec and 12.6sec is displayed in the other figures.

performance, we repeat the same experiment using only samples with a defined reverberation time. Following this procedure, the training and hold-out data sets shrink by a factor of ten, which deteriorates the performance, as can be seen in Figure 4.2(a). Apart from this deterioration due to the small training data set, there is no clear effect of the reverberation time onto the classification accuracy.

Feature Name	Importance [%]
betaopt34n_960	0.882
ea34n_abs_3920	0.689
ef12n_abs_3920_a10	0.689
fc34n_abs_3920	0.677
PitchMean	0.671
ea12n_abs_3920_a10	0.629
fc12n_abs_3920_a10	0.613
betaopt12n_3360_a10	0.535
betaopt34n_160	0.532
ef34n_abs_3920	0.503
ef34n_ph_480	0.492
bc12n_abs_480_a10	0.492
betaopt12n_1120	0.481
betaopt12n_960	0.481
betaopt34n_3360	0.465

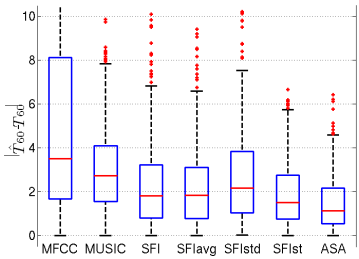
Table 4.1: The 15 most important features for prediction of the hearing activity with random forests. The importance of the feature is determined by the increase in the balanced error rate (in %) when the values of the feature are randomly permuted over all data items.

4.3.2 Regression for Reverberation Time

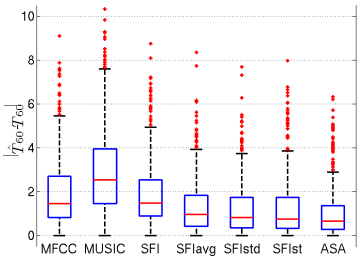
In this section, we report results for the reverberation time prediction in different settings and with several regression methods.

In a first experiment, the regression analysis is conducted on samples from all hearing activities. The distributions of the absolute values of the residual for different methods and different feature sets is reported in Figure 4.3. Independent of the feature set, linear models only poorly predict the reverberation time. The three other methods yield comparable accuracy, but nonlinear models (random forests and SVM with rbf kernel) tend to be more accurate than the linear SVM.

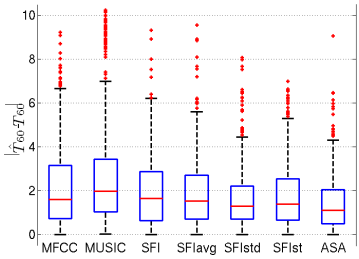
Namely the MUSIC yield poor results, and also the results obtained on MFCC are mostly behind the results obtained with the SFI-based or the ASA features. This observation indicates their independence on the room characteristics and is an important property in the application they are originally designed for: If the goal is to represent speech or characterize



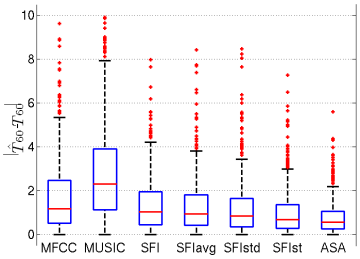
(a) linear model



(b) Random Forest



(c) SVM, linear kernel

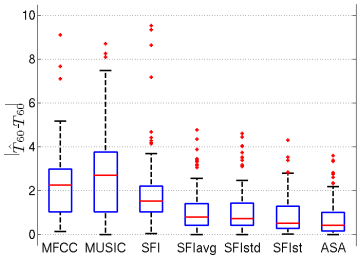


(d) SVM, rbf kernel

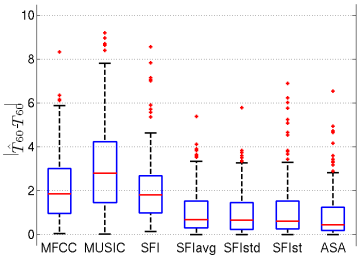
Figure 4.3: Residuals for regression on the reverberation time T_{60} for sound clips of all hearing activities. The linear regression model performs poorly compared to the random forest. The SVM with both linear and rbf kernel perform comparable to the random forest.

different genres of music, the features are ideally invariant to reverberation. It is therefore to be expected that predicting T_{60} based on these features yields a poor performance. The SFI features, the SFI short-time statistics and the ASA features are all comparable in their prediction performance.

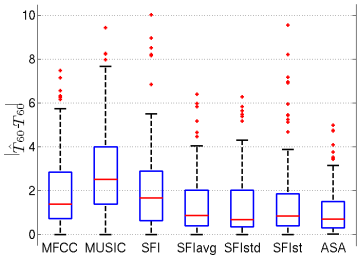
In a second experiment, the regression of the reverberation time is performed conditioned on the hearing activity. For human listeners, reverberation estimation becomes more difficult as the signal loses structure. We simulate this setting by predicting T_{60} for samples with hearing activity *clean speech*, *speech in low noise* (SNR $\geq$ 5dB), *speech in high noise* (SNR $\leq$ 2dB) and *noise* separately, using random forests (see Figure 4.4). Surprisingly,



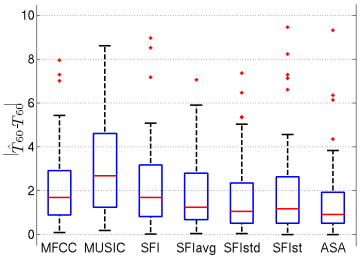
(a) clean speech



(b) speech in noise, SNR $\geq 5dB$



(c) speech in noise, SNR $\leq 2dB$



(d) noise

Figure 4.4: Residuals for regression on the reverberation time T_{60} for hearing activities *clean speech*, *speech in low noise* ($SNR \geq$ 5dB), *speech in high noise* (SNR $\leq$ 2dB) and *noise*, for regression with random forests.

there is on the average only a small decrease in the performance as the signal looses structure. However, the variance in the performance clearly increases as the signal looses structure.

The most relevant features for reverberation prediction with random forests are listed in Table 4.2. These are mostly features from set SFIstd that measure the short-time variations of the sound field indicators on low frequencies. Compared to the features which are reliable to predict the hearing activity (see Table 4.1), these are clearly different feature statistics, and the relevant features measure properties at lower frequency bands. `Onset` is the only feature from a different feature set, namely the ASA feature set. Furthermore, a strong decay in the feature importance is observed,

Feature Name	Importance
P11_320_s10	0.7800
PMean12_320_s10	0.5830
P22_320_s10	0.4853
PMean12_800_s10	0.4584
P22_800_s10	0.4233
P11_800_s10	0.4204
P11_640_s10	0.2239
Ia12n_1840_a10	0.1928
Onset	0.1904
PMean12_640_s10	0.1802
P22_640_s10	0.1571
P11_960_s10	0.1339
P22_480_s10	0.1247
G12_ph_9440_s10	0.1117
P11_1840_s10	0.1107

Table 4.2: The 15 most important features for regression of reverberation time with random forests. The importance of the feature measure the increase in the residuum when the values of the feature are randomly permuted over all data items.

with a vast majority of the features having a negligible importance for the prediction of the reverberation time.

Chapter 5

Methods for Multi-Label Classification

We review methods for multi-label classification and then propose a novel, generative approach to this task. Experiments on both synthetic and real-world data show that the proposed method outperforms state-of-the-art techniques with respect to parameter and classification accuracy.

5.1 Related Work

The approaches to solve the task of multi-label classification can be grouped into three categories: *Transformation* methods reduce the multi-label problem to a series of single-label classification problems, whose solutions are then combined to answer to the original task of multi-label classification. *Algorithm adaptation* methods extend established classification techniques such that they can handle multiple labels. Finally, *ranking-based* methods infer a ranking on the relevance of all possible labels for a given label along with a cut-off. Labels ranked above the cut-off are then assigned to the label set for the data item at hand. In the following, we present these three approaches in detail.

5.1.1 Transformation Methods

Reducing a given problem to one or several problems for which a solution is already known is a widely-used approach in mathematics and computer science. In supervised learning, it is often easier to design algorithms for dichotomies, i.e. that distinguish between two classes. While some of these algorithms can naturally be extended to separate multiple classes, such a generalization becomes more involved for other techniques such as AdaBoost and support vector machines. A single-label, multi-class problem, where a given data item is to be assigned to one of K classes, with $K > 2$, can be solved by deciding for each class individually whether or not a given data item belongs to this class. With this approach, often termed *one-against-all classification* [95], the original multi-class problem is reduced to a series of K dichotomies. The results of these dichotomies are then combined to a single class label, using the constraint that each data item belongs to exactly one class. Advanced combination techniques include e.g. the use of error-correcting output codes [35]. A unifying framework for the reduction of multi-class problems to binary problems for margin classifiers is presented in [3].

Given one-against-all classification, the extension to multi-label classification is straightforward: Drop the constraint that each data item belongs to exactly one class and allow assignments to more than one class. This technique was successfully applied to scene classification [11] and to classification of emotions in music [77].

More sophisticated methods either alleviate the burden of large-scale multi-label classification tasks by reducing the number of possible label sets to be taken into consideration, or they obtain more accurate classification results by using prior information about the distribution of label sets. The principle of maximum entropy [63, 64] is employed in [122] to capture correlations in the label set. The assumption of small label sets is exploited in the framework of compressed sensing by [60]. Conditional random fields are used in [51] to parameterize label co-occurrences. Instead of independent dichotomies, a series of classifiers is build in [94], where a classifier gets the output of all preceding classifiers in the chain as additional input.

To our knowledge, all transformation methods use all data items with label k to infer the binary classifier for this class, thereby ignoring all other labels of the data items. Inspired by [11], we call this inference technique *cross training* and denote it by $\mathcal{M}_{cross}$. Note that in this approach, the total influence of a data item on the set of all parameter estimators grows

with its label degree: While a single-label data item is used to train the parameters of one source, a data item with degree, say, three influences the estimation of the parameters of three sources.

A variation of cross training is proposed in [97] for improved functional prediction of proteins. The total weight of each data item is equal, the individual weights of labels are either given along with the label set, or, in the absence of such information, are computed by uniformly distributing the total weight over all labels in the label set. We refer to this method as *probabilistic training* $\mathcal{M}_{prob}$.

By ignoring the fact that a multi-label data item contains at least one label besides the currently considered class k, transformation methods explain the entire data item with the single class k. This serious simplification typically leads to a high variation in the data used to train the classifier for class k. Thus inferring very broad classes, a classifier trained by cross training will very generously assign labels, thus yielding a high recall, but a low precision. We expect the same effect for probabilistic training, but to a smaller extend, as with this training method, multi-label data items have a smaller weight on the estimation of the parameters of an individual source than in cross training.

Two further transformation methods are conceptually simple. The most agnostic method to handle multi-label data is to simply ignore data items with multiple labels. We denote this method by $\mathcal{M}_{ignore}$ [11]. Alternatively, consider each label set found in the training data as label of a new multi-class single-label classification task [107]. This method, dubbed new training $\mathcal{M}_{new}$, is also called *combination method* or *label powerset learning*. Due to the high number of possible label sets even for moderate numbers of classes, new training typically has to learn the parameters of a large number of classes based on a small number of data items per class. To alleviate this problem, the *Pruned sets* method [93] splits label sets which occur infrequently in the training data into smaller label sets. In the *random k-labelsets* (RAKEL) algorithm [108], the pruning is implemented with respect to the maximal size of label sets, with too large label sets being discarded.

5.1.2 Algorithm Adaptation Methods

Several adaptations of instance-based classifiers for multi-label classification were proposed in the literature. The k-nearest neighbor algorithm was adapted to multi-label data in [121]. Based on the k nearest neighbors,

a decision on the class membership of a new data item is taken independently for each class. A modified entropy formula was employed in [25] to adapt the C4.5 algorithm for knowledge discovery in multi-label phenotype data. Given the modified entropy formula, frequently co-occurring classes are distinguished only on the bottom of the decision tree. Support vector machines were introduced for this task in [67] and were shown to outperform competing algorithms such as nearest neighbor and C4.5 algorithms.

Boosting was applied to multi-label text classification e.g. in [98]. Weak learners were trained to either minimize the Hamming loss (AdaBoost.MH) or the ranking loss (AdaBoost.MR). Kernel methods are introduced in [42] for dichotomy learning and ranking in order to solve the multi-label problem.

Latent semantic indexing (LSI) [32] is a technique for unsupervised dimension reduction which aims at finding a linear mapping from the input space to some low-dimensional latent space, thereby recovering most of the structure in the data. Reformulating the cost function of LSI as a combination of the reconstruction error of the observation and the label data [120], a low-dimensional latent space for both the observations and the label sets is obtained.

An adaptation of a generative process to multi-label classification for text categorization was presented in [81]. The generative process for a document is as follows: Select a label set $\mathcal{L}$ and then a vector of mixture weights among the classes in the label set. For each word in a document, first choose a class out of the label set according to the mixture weight, and then sample a word from this class according to the class-conditional distribution of words. For simplicity, words are assumed to be independent of each other. Since the mixture weights can not be observed, the model parameters are inferred by estimation-maximization. A similar idea is pursued in [111], where the class-conditional word probabilities are combined to word probabilities for documents which belong to multiple categories. Two versions of the model are presented, where the mixture weight of the class-conditional word probabilities is either equal for all classes in a label set, or estimated during the inference phase.

Strictly speaking, the approach presented in [81] models the task of text classification as a multi-instance rather than a multi-label problem: Every word is generated by a single source, and the text is obtained by concatenating the different words. The inference tasks, on the other hand, consists of first identifying the subset of words that are generated by each of the sources in the label set. This division of the text into words implies a single-label

classification of words, and inferring the class-conditional densities reduces to a standard task.

5.1.3 Ranking-Based Methods

The problem of label ranking consists of learning a mapping from data items to rankings over a given number of K class labels. Label ranking and classification are related to each other as follows: Given a relevance ranking over classes for a data item, single-label classification selects the most relevant class as the label for the data item at hand. Multi-label classification with a given number $|\mathcal{L}|$ of classes for a data item selects the $|\mathcal{L}|$ most relevant classes as label sets for this data item [16]. Conversely, a label or label set implies that the respective class(es) are more relevant for the observed data item than any class which is not in the label set of the data item.

Based on the training set, a classifier can be trained for every pair of labels (λ_1, λ_2) to decide which of the two labels are more relevant for a given data item. If the label set of a data item contains λ_1, but not λ_2, then λ_1 is more important than λ_2 for the given data item. If λ_1 and λ_2 are both in the label set or both not in the label set, then no information about the relative relevance can be derived. To compute the label ranking for a new observation, the $K \cdot (K-1)/2$ binary comparisons are combined to a relevance ranking over all labels, with ties broken randomly [61]. To determine the size of the label set, a "neutral" calibration label is introduced to separate relevant from non-relevant labels [49]. The label set then consists of all labels ranked more important than the neutral label.

The training of the binary classifier for a pair of labels (λ_1, λ_2) relies on a similar assumption as cross training: Information about the importance is deducted from every data item which contains exactly one of the two labels in its label set, thus ignoring all other labels which are possibly in the label set as well. Moreover, the characteristics of data items with a particular label λ_1 depend on the label pair which is to be ranked. Ranking-based methods therefore fail to provide a detailed semantic interpretation of multi-label data.

5.2 Generative Single-Label Classifiers

A classifier is a mapping that takes vector of observed features X as input and provides a class label λ (in the single-label case) or a label set $\mathcal{L}$ (in the multi-label case) as output. A generative classifier assumes a generative process (e.g. the process presented in Section 2.1.3 for the observed X in order to predict the set of sources $\mathcal{L}$ involved in the generation of X.

5.2.1 Training Phase

In the training phase, the parameters $\boldsymbol{\theta}$ of the assumed generative process are estimated. This inference is based on a set of observations $\boldsymbol{X} = (X_1, \ldots, X_N)$ with the corresponding label sets $\boldsymbol{\mathcal{L}} = (\mathcal{L}_1, \ldots, \mathcal{L}_N)$. A popular method to determine the parameters is the *maximum likelihood principle*: The parameters are chosen such that the observed data is most likely under the assumed generative model. Formally, with $\mathbf{D} := (\boldsymbol{X}, \boldsymbol{\mathcal{L}})$, the function

$$L : \boldsymbol{\theta} \mapsto P(\mathbf{D}|\boldsymbol{\theta}) \qquad L(\boldsymbol{\theta}; \mathbf{D}) = P(\mathbf{D}|\boldsymbol{\theta}) \tag{5.1}$$

is defined as the *likelihood function*. The *maximum likelihood estimator* $\hat{\boldsymbol{\theta}}_{ML}$ is chosen such that the likelihood function attains its maximum:

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta} \in \Theta} L(\boldsymbol{\theta}; \mathbf{D}) \ . \tag{5.2}$$

Since the logarithm of the likelihood function attains its maximum for the same values of $\boldsymbol{\theta}$ and is often easier to manipulate, we can alternatively compute the maximum-likelihood estimator as

$$\hat{\boldsymbol{\theta}}_{ML} = \arg\max_{\boldsymbol{\theta} \in \Theta} \ell(\boldsymbol{\theta}; \mathbf{D}) \qquad \text{with } \ell(\boldsymbol{\theta}; \mathbf{D}) := \log L(\boldsymbol{\theta}; \mathbf{D}) \ . \tag{5.3}$$

Some prior knowledge or assumptions on the value of the parameter might be available. The parameter $\boldsymbol{\theta}$ is then considered as a random variable with prior distribution $P(\boldsymbol{\theta})$. Using Bayes' theorem [6], the *a posteriori probability distribution* of $\boldsymbol{\theta}$ after observing the data set $\mathbf{D}$ is given by

$$P(\boldsymbol{\theta}|\mathbf{D}) = \frac{P(\mathbf{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})}{\int_\Theta P(\mathbf{D}|\boldsymbol{\theta'}) \cdot P(\boldsymbol{\theta'}) \, \mathrm{d}\boldsymbol{\theta}}$$

The method of *maximum a posteriori estimation* determines the parameter $\boldsymbol{\theta}$ such that the posterior distribution of this random variable attains its maximum:

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta} \in \Theta} P(\boldsymbol{\theta}|\mathbf{D}) = \arg\max_{\boldsymbol{\theta} \in \Theta} \{P(\mathbf{D}|\boldsymbol{\theta}) \cdot P(\boldsymbol{\theta})\} \ . \tag{5.4}$$

Or, written again with the logarithm,

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}\in\Theta} \{\ell(\boldsymbol{\theta}; \mathbf{D}) + \log P(\boldsymbol{\theta})\} \ , \tag{5.5}$$

where the maximization only ranges over values of $\boldsymbol{\theta}$ for which $P(\boldsymbol{\theta}) > 0$.

Comparing the two expressions for the maximum a posteriori estimator (Eq. 5.4 or Eq. 5.5) and the maximum likelihood estimator (Eq. 5.2 or Eq. 5.3), it becomes apparent that the maximum likelihood estimator corresponds to a maximum a posteriori estimator with a uniform prior distribution over all possible values of the parameter $\boldsymbol{\theta}$. Such a prior is called an *uninformative prior* [65].

It is common to assume that different data items $(X_n, \mathcal{L}_n)$, $n = 1, \ldots, N$, of a data set $\mathbf{D}$ are independent and identically distributed (*i.i.d.*). Under this assumption, the probability of $\mathbf{D}$ is the product over the probabilities of the single data items $(X_n, \mathcal{L}_n)$. Inserting this into Eq. 5.5, we get

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}\in\Theta} \left\{ \sum_{n=1}^{N} \ell(\boldsymbol{\theta}; X_n, \mathcal{L}_n) + \log P(\boldsymbol{\theta}) \right\} \ . \tag{5.6}$$

As the number of data items grows, the maximum a posteriori estimator thus converges to the maximum likelihood estimator, or, alternatively speaking, the evidence of the data becomes more important than the prior as more and more data items are available.

5.2.2 Decision Phase

In the decision phase, the label set $\mathcal{L}^{new}$ of a new observation X^{new} is to be determined. Similar to the estimation of the parameters in the training phase, the label set of a new emission can be determined according to the principles of maximum likelihood or maximum a posteriori.

In maximum likelihood classification, the label set $\hat{\mathcal{L}}^{new}$ of the new observation is estimated such that the observation is most likely, given the model assumptions and the parameters $\hat{\boldsymbol{\theta}}$ inferred in the training phase:

$$\hat{\mathcal{L}}^{new} = \arg\max_{\mathcal{L}\in\mathbb{L}} P(X^{new}|\hat{\boldsymbol{\theta}}, \mathcal{L}) \tag{5.7}$$

Alternatively, a prior distribution $P(\mathcal{L})$ over the set $\mathbb{L}$ of all possible label sets might be used in the classification. This prior distribution is

either inferred in the training phase or given externally. Using this prior, the maximum a posteriori label set is determined according to

$$\hat{\mathcal{L}}^{new} = \arg\max_{\mathcal{L}\in\mathbb{L}} \left\{ P(X^{new}|\hat{\boldsymbol{\theta}}, \mathcal{L}) \cdot P(\mathcal{L}) \right\} . \tag{5.8}$$

Maximum likelihood and maximum a posteriori classification results are most different when the class prior is structured, e.g. uni- or multi-modally peaked. An example for this situation are medical screenings: Often, most of the persons do not suffer from a particular disease X, i.e. we have $P(\text{Patient suffers from } X) \ll P(\text{Patient does not suffer from } X)$.

5.3 A Generative Model for Multi-Label Classification

We propose an approach to classification of multi-labeled data which extends the generative model for single-label data by interpreting multi-labeled data as a superposition of the emissions of the individual sources. A data item X with label set $\mathcal{L} = \{\lambda_1, \ldots, \lambda_M\}$ of degree M is assumed to be the sum of one sample from each of the contributing sources, i.e.

$$X = \sum_{k=1}^{M} \Xi_{\lambda_k} \qquad \text{with} \qquad \Xi_{\lambda_k} \sim P_{\lambda_k} \tag{5.9}$$

The distribution of X is thus given by the convolution of all contributing sources:

$$X \sim P_{\lambda_1} * \ldots * P_{\lambda_M} =: P_{\mathcal{L}} \tag{5.10}$$

Thus, unlike in new training, the distribution of data with multiple labels is traced back to the distribution of the contributing sources. We therefore propose the name *deconvolutive* model and refer to this model as $\mathcal{M}_{deconv}$.

Note that it is possible to explicitly give the distribution $P_{\mathcal{L}}$ for data with label set $\mathcal{L}$. In contrast to new training, which would estimate $P_{\mathcal{L}}$ based solely on the data with this label set, we propose to compute $P_{\mathcal{L}}$ from the distributions of all sources contained in $\mathcal{L}$. On the other hand, the estimation of each source distribution is based on all data items which contain the respective source in their label sets.

5.3.1 Learning a Model for Multi-Label Data

In the following, we first describe the learning and classification steps in general and then we provide an explicit formula for the special case of Gaussian distributions. In order to simplify the notation, we limit ourselves to the case of data generated by at most two sources and to inference according to the principle of maximum likelihood. The generalization to label sets of higher degree and to maximum a posteriori inference is straightforward. Furthermore, we index the probability distributions with the label, i.e. $P_k(X)$ stands for $P(X|\theta_k)$ in the rest of this chapter.

General Learning Scenario

The probability distribution of multi-labeled data is given by Eq. 5.10. The likelihood of a data item x given a label set $\mathcal{L} = \{\lambda_1, \lambda_2\}$ is

$$\begin{aligned} P_{\{\lambda_1,\lambda_2\}}(X) &= (P_{\lambda_1} * P_{\lambda_2})(X) \\ &= \int P_{\lambda_2}(X - \Xi)\, \mathrm{d}P_{\lambda_1}(\Xi) \qquad (5.11) \\ &= \mathbb{E}_{\Xi \sim P_{\lambda_1}}[P_{\lambda_2}(X - \Xi)]\,. \qquad (5.12) \end{aligned}$$

In general, it may not be possible to solve the convolution integral (Eq. 5.12, and similar terms for superpositions of more sources) analytically. In such cases, the formulation as an expected value (Eq. 5.12) renders numerical techniques such as Monte Carlo sampling possible.

In the training phase, the optimal parameters $\hat{\theta}_k$ of the distribution P_k are chosen according to the principle of maximum likelihood (Eq. 5.3), which implies the condition

$$\frac{\partial}{\partial \theta_k} \left\{ \sum_{\mathcal{L} \in \mathbb{L}} \sum_{n:\mathcal{L}_n = \mathcal{L}} \log P_{\mathcal{L}}(x_n) \right\} \stackrel{!}{=} 0 \qquad \text{for } k = 1, \ldots, K\,. \qquad (5.13)$$

Classification in the General Case

When classifying a new data item X^{new}, the estimated label set $\hat{\mathcal{L}}^{new}$ is determined according to the principle of maximum a posteriori estimation (Eq. 5.8). As in training, if the probability distribution of a data item X^{new} with label set $\mathcal{L}^{new} = \{\lambda_1^{new}, \lambda_2^{new}\}$ of degree 2 can not be expressed in closed form, Eq. 5.12 might be used to get an estimate of

$P_{\{\lambda_1^{new},\lambda_2^{new}\}}(X^{new})$ by sampling Ξ from $P_{\lambda_1^{new}}$. The generalization to label sets of degree larger than 2 is straight forward.

The classification rule in Eq. 5.8 corresponds to a search over the set $\mathbb{L}$ of possible labels. The large size of the search space for this optimization can render the search for the optimal label very demanding. However, this complexity is a property of the assumed generative process and thus an inherent property of the data. We present an approximation method for the classification in Section 5.3.2.

Gaussian Distributions

Let us assume for the remainder of this section that all source distributions are D-dimensional Gaussian distributions, i.e. $P_k = \mathcal{N}(\mu_k, \Sigma_k)$, with $\mu_k \in \mathbb{R}^{1\times D}$ and a positive-definite $D \times D$ matrix Σ_k, for $k = 1, \ldots, K$. The convolution of Gaussian distributions is again a Gaussian distribution, where the mean vectors and the covariance matrices are added:

$$\mathcal{N}(\mu_1, \Sigma_1) * \mathcal{N}(\mu_2, \Sigma_2) = \mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2) \ . \tag{5.14}$$

By induction, a corresponding rule holds for convolutions of more than two Gaussian distributions. This property drastically simplifies the algebraic expressions in our model.

Training for Gaussian Distributions. To find the optimal values for the means and the covariance matrices, we have to solve the maximum likelihood conditions

$$\frac{\partial}{\partial \mu_k}\left\{\sum_{\mathcal{L}\in\mathbb{L}}\sum_{n:\mathcal{L}_n=\mathcal{L}} \log P_{\mathcal{L}}(x_n)\right\} \stackrel{!}{=} 0 \qquad \frac{\partial}{\partial \Sigma_k}\left\{\sum_{\mathcal{L}\in\mathbb{L}}\sum_{n:\mathcal{L}_n=\mathcal{L}} P_{\mathcal{L}}(x_n)\right\} \stackrel{!}{=} 0$$

for $k = 1, \ldots, K$. These conditions yield a set of coupled nonlinear equations, which can be decoupled by proceeding iteratively. As initial values for this iterative optimization procedure, we choose the sample mean and variance of the single-label training data:

$$\mu_k^{(0)} = \frac{\sum_{n:\mathcal{L}_n=\{k\}} x_n}{|\{n : \mathcal{L}_n = \{k\}\}|} \qquad \Sigma_k^{(0)} = \frac{\sum_{n:\mathcal{L}_n=\{k\}} (x_n - \mu_k^{(0)})(x_n - \mu_k^{(0)})^T}{|\{n : \mathcal{L}_n = \{k\}\}|}.$$

For simpler notation, we define the mean and covariance matrix of all

sources except k in $\mathcal{L}_n$ as follows:

$$m^{(t)}_{\mathcal{L}_n\setminus\{k\}} := \sum_{\substack{\lambda\in\mathcal{L}_n\\ \lambda\neq k}} \mu^{(t)}_\lambda \qquad S^{(t)}_{\mathcal{L}_n\setminus\{k\}} := \sum_{\substack{\lambda\in\mathcal{L}_n\\ \lambda\neq k}} \Sigma^{(t)}_\lambda \ ,$$

where upper indices indicate the iteration steps. Using an iterative approach, the condition for the mean values yields the following update formula for μ_k, $k = 1, \ldots, K$:

$$\mu_k^{(t+1)} := \left(\sum_{n:\mathcal{L}_n\ni k} (x_n - m^{(t)}_{\mathcal{L}_n\setminus\{k\}}) \left(\Sigma^{(t)}_{\mathcal{L}_n}\right)^{-1}\right)\left(\sum_{n:\mathcal{L}_n\ni k} \left(\Sigma^{(t)}_{\mathcal{L}_n}\right)^{-1}\right)^{-1} . \quad (5.15)$$

Deriving the data likelihood with respect to the covariance matrix Σ_k yields the following condition:

$$\frac{1}{2}\sum_{n:\mathcal{L}_n\ni k} \left(\left(\mathbf{1}_D - (x_n - \mu_{\mathcal{L}_n})(x_n - \mu_{\mathcal{L}_n})^T\Sigma^{-1}_{\mathcal{L}_n}\right)\Sigma^{-1}_{\mathcal{L}_n}\right) \stackrel{!}{=} 0,$$

where $\mathbf{1}_D$ denotes the identity matrix in D dimensions. With $\Sigma_{\mathcal{L}_n} = \Sigma_k + S_{\mathcal{L}_n\setminus\{k\}}$ and

$$V^{(t)}_{n,\mathcal{L}_n} = (x_n - \mu^{(t)}_{\mathcal{L}_n})(x_n - \mu^{(t)}_{\mathcal{L}_n})^T \ ,$$

the optimality condition for Σ_k can be rewritten as

$$\begin{aligned}\sum_{n:\mathcal{L}_n=\{k\}} &\left(\left(\mathbf{1}_D - V_{n\mathcal{L}_n}\Sigma_k^{-1}\right)\Sigma_k^{-1}\right)\\ &+ \sum_{\substack{n:\mathcal{L}_n\ni k\\ |\mathcal{L}_n|>1}} \left(\left(\mathbf{1}_D - V_{i\mathcal{L}_n}(S_{\mathcal{L}_n\setminus k} + \Sigma_k)^{-1}\right)(S_{\mathcal{L}_n\setminus k} + \Sigma_k)^{-1}\right) \stackrel{!}{=} 0\end{aligned} \quad (5.16)$$

Note that for a training set containing only single label data, the second sum in Eq. 5.16 vanishes, and the condition implies estimating Σ_k by the sample covariance matrix. If the training set contains data with multiple labels, the optimality condition can in general not be solved analytically, as the condition for Σ_k corresponds to a polynomial whose degree is twice the number of allowed label sets in $\mathbb{L}$ containing k. In this case, the optimal value of $\Sigma_k^{(t)}$ can either be determined numerically or using the Taylor approximation

$$\begin{aligned}(S_{\mathcal{L}_n\setminus k} + \Sigma_k)^{-1} &= \Sigma_k^{-1}\left(S_{\mathcal{L}_n\setminus s}\Sigma_k^{-1} + \mathbf{1}_D\right)^{-1}\\ &\approx \Sigma_k^{-1}\left(\mathbf{1}_D - \Sigma_k S^{-1}_{\mathcal{L}_n\setminus k}\right) = \Sigma_k^{-1} - S^{-1}_{\mathcal{L}_n\setminus k} \ .\end{aligned}$$

The approximation is typically quite crude; we therefore prefer using a numerical solver to determine $\Sigma_k^{(t)}$ for all sources k after having determined the mean values $\mu_k^{(t)}$. We observed that the estimator for the mean values is relatively robust with respect to changes in the covariance matrix. Furthermore, the relative importance per data item for the estimation of $\mu_k^{(t)}$ decreases as the degree of its label increases. If enough data items with low degree label sets are available in the training phase, the convergence of the training step can be increased by discarding data items with high label degrees with only minor changes in the accuracy of the parameter estimates.

Classification for Gaussian Distributions. Recall the explicit formula for the convolution of two Gaussian distributions (Eq. 5.14). This relation yields a simple expression for the likelihood of the data X^{new} given a particular candidate label set $\mathcal{L}^{new} = \{\lambda_1^{new}, \lambda_2^{new}\}$:

$$P_{\mathcal{L}^{new}}(X^{new}) = \mathcal{N}(X^{new}; \hat{\mu}_{\lambda_1^{new}} + \hat{\mu}_{\lambda_2^{new}}, \hat{\Sigma}_{\lambda_1^{new}} + \hat{\Sigma}_{\lambda_2^{new}})$$

Again, the label set for the new data item is assigned according to the maximum a posteriori rule (Eq. 5.8). As the density functions for data with multiple labels are computed based on the single source densities, this yields more accurate density estimates namely for data with medium to large label degree. This is the second major advantage of the proposed algorithm.

Further Examples of Stable Probability Distributions

The methods presented in the previous section for Gaussian distributions are very general and they are applicable to all parametric distributions and combination functions. However, an explicit expression for the probability distribution of multi-label data is exists only for specific pairs of distribution and combination function. As exemplified in the previous section, the addition of source emissions yields a random variable whose distribution is described by the convolution of the distributions of the involved sources. Since the convolution of Gaussian distributions is again a Gaussian distribution, the probability distribution of the proxy distributions $P_{\mathcal{L}}$ for $|\mathcal{L}| > 1$ are of the same type as the source distributions P_k, and their parameters are easily computed based on the parameters of the source distributions. This fact dramatically simplifies the calculations, and for small problem sizes, the optimal parameter values can even be computed explicitly. Such a closed form expression for the convolution integral and analytical solution of the

optimal parameter values leads to much faster training and classification. In the following, we describe further pairs of source distributions and combination functions for which the distribution of the combination of several emissions is in the same family as the distribution of the single emissions.

Exponential distribution: If $\Xi_j \sim Exp(\theta_j)$ with $\theta_j > 0$ for $j = 1, 2$, then the minimum of the two emissions is also exponentially distributed: $\min(\Xi_1, \Xi_2) \sim Exp(\theta_1 + \theta_2)$.

Log-normal Distribution: If $\Xi_j \sim \text{Log-}\mathcal{N}(\mu_j, \Sigma_j)$, $j = 1, 2$, the product also has a log-normal distribution: $\Xi_1 \cdot \Xi_2 \sim \text{Log-}\mathcal{N}(\mu_1 + \mu_2, \Sigma_1 + \Sigma_2)$.

Bernoulli Distribution: If $\Xi_j \sim Ber(p_j)$, $j = 1, 2$, then the conjunction is Bernoulli distributed as well: $\Xi_1 \wedge \Xi_2 = \min\{\Xi_1, \Xi_2\} \sim Ber(p_1 \cdot p_2)$. For the disjunction, we have $\Xi_1 \vee \Xi_2 = \max\{\Xi_1, \Xi_2\} \sim Ber(p_{12})$, with $p_{12} := p_1 + p_2 - p_1 \cdot p_2$.

Gamma Distribution: If $\Xi_j \sim \gamma(b, p_j)$ with $b > 0$ and $p_j > 0$ for $j = 1, 2$, the sum also follows a Gamma distribution: $\Xi_1 + \Xi_2 \sim \gamma(b, p_1 + p_2)$.

5.3.2 Efficient Classification

In the proposed model, the classification tasks consist of choosing a subset of the given sources such that the observed data item has maximal likelihood. If no restrictions on the set $\mathbb{L}$ of possible label sets apply, all possible subsets of the source set $\mathcal{K}$ have to be considered. As there are $\mathcal{O}\left(2^K\right)$ subsets of $\mathcal{K}$, even classification problems with moderate value of K have a prohibitively high running time. However, good approximations are possible in the present case, as we exemplify in the following for sources with Gaussian distribution.

A Heuristic to Reduce the Search Space. For Gaussian distributions with equal spherical covariance matrix $\Sigma_k = \sigma^2 \cdot \mathbf{1}_D$ for all sources $k = 1, \ldots, K$, maximum a posteriori classification (Eq. 5.8) of a new data item $X^{new} \in \mathbb{R}^D$ can be reduced to

$$\hat{\mathcal{L}}^{new} = \arg\max_{\mathcal{L} \in \mathbb{L}} \left\{ \frac{\pi_{\mathcal{L}}}{\sigma^D (2\pi|\mathcal{L}|)^{D/2}} \exp\left(-\frac{||X^{new} - \mu_{\mathcal{L}}||_2^2}{2|\mathcal{L}|\sigma^2} \right) \right\} \tag{5.17}$$

$$= \arg\min_{\mathcal{L} \in \mathbb{L}} \left\{ ||X^{new} - \mu_{\mathcal{L}}||_2^2 + |\mathcal{L}|\sigma^2 (D \log(2\pi\sigma^2|\mathcal{L}|) - 2\log(\pi_{\mathcal{L}})) \right\} , \tag{5.18}$$

where $\pi_{\mathcal{L}}$ is the prior probability of the label set $\mathcal{L}$. In cases where the set $\mathbb{L}$ of admissible label sets is relatively small, $\hat{\mathcal{L}}^{new}$ can be found directly within reasonable computation time. Such a situation arises e.g. when the new data can only be assigned to a label set that is present in the training set, i.e. if $\mathbb{L}$ is the set of all label sets contained in the training sample, or when other restrictions are coded in the prior distribution if $\mathcal{L}$, e.g. a limitation to label sets with small degree. However, in a more general setting, there are no such constraints, and the classifier should also be able to assign a label set that is not seen during the training phase. In this case, $\mathbb{L}$ contains $|2^{\mathcal{K}}| - 1 = 2^K - 1$ possible label sets. The time for direct search thus grows exponentially with the number of labels K.

To address this problem, we determine a subset of sources $\mathcal{K}^- \subset \mathcal{K}$ which, with high probability, have not contributed to X^{new}. The optimization in Eq. 5.8 is then restricted to label sets $\mathcal{L}$ containing only sources which have not been excluded. This constraint limits the search space and consequently speeds up classification.

Note that all terms in Eq. 5.18 are positive. The label set prior typically decreases as the degree increases, and the second term grows logarithmically in the size of the label set. The later term thus tends to privilege smaller label sets, and neglecting these two terms might thus yield larger label sets. This is a type of regularization which we omit in the following, as we approximate Eq. 5.18 by the following subset selection problem:

$$\hat{\mathcal{L}}^{new} = \arg\min_{\mathcal{L}\in\mathbb{L}} \left\{ ||X^{new} - \mu_{\mathcal{L}}||_2^2 \right\}, \tag{5.19}$$

We define the indicator vector $\hat{\mathbf{z}}^{new} \in \{0,1\}^K$, with $\hat{z}_k^{new} = 1$ if $k \in \hat{\mathcal{L}}^{new}$ and $\hat{z}_k^{new} = 0$ otherwise, for all sources k. Using this notation, the above minimization problem can be written as

$$\hat{\mathbf{z}}^{new} = \arg\min_{\mathbf{z}\in\{0,1\}^K} \left\{ \sum_{k=1}^{K} z_k \mu_k - X^{new} \right\}.$$

Relaxing the constraints on $\hat{\mathbf{z}}^{new}$ to $\tilde{\mathbf{z}} \in \mathbb{R}^K$, we get the following regression problem:

$$\tilde{\mathbf{z}}^{new} = \arg\min_{\tilde{\mathbf{z}}\in\mathbb{R}^K} \left\{ \sum_{k=1}^{K} \tilde{z}_k \mu_k - X^{new} \right\}.$$

Defining the matrix M of mean vectors as $M = (\mu_1, \ldots, \mu_K)^T \in \mathbb{R}^{K\times D}$, we obtain the least-squares solution for the regression problem:

$$\tilde{\mathbf{z}}^{new} = X^{new} M^T (M M^T)^{-1} \tag{5.20}$$

In order to reduce the size of the search space for the label set, we propose to compute a threshold τ for the components of $\tilde{\mathbf{z}}^{new}$. Only sources k with $\tilde{z}_k^{new} > \tau$ are considered further as potential members of $\tilde{\mathcal{L}}^{new}$.

As we have omitted the constraints favoring small label sets, single sources with mean close to X^{new} might be discarded. This effect can be compensated by adding label sets of small degree (up to 2 is mostly sufficient) containing only discarded classes to the reduced label set. Formally, we define $\mathcal{K}^+ := \{k \in \mathcal{K} | \tilde{z}_k^{new} > \tau\}$ and $\mathbb{L}^+ := \{\mathcal{L} \in \mathbb{L} | \mathcal{L} \subseteq \mathcal{K}^+\}$ and replace $\mathbb{L}$ in Eq. 5.17 by $\mathbb{L}^+$. In our experiments, we found that this heuristic can drastically reduce computation times in the classification task. The error probability introduced by this technique is discussed in the following.

Error Analysis. We assume the true label set of X^{new} is $\mathcal{L}^{new}$, with the corresponding indicator vector $\mathbf{z}^{new}$. The heuristic introduces an error whenever $\tilde{z}_k^{new} < \tau$ but $z_k^{new} = 1$, i.e. k is in the true label set $\mathcal{L}^{new}$. Thus,

$$P[\text{error}] = 1 - \prod_{k \in \mathcal{L}^{new}} P[\tilde{z}_k^{new} > \tau].$$

For the analysis, we assume that all source distributions have the same variance $\sigma^2 \cdot \mathbf{1}_D$. Then, we have

$$X^{new} = \mathbf{z}^{new} M + \epsilon \ , \quad \text{with } \epsilon \sim \mathcal{N}(0, |\mathcal{L}^{new}| \cdot \sigma^2 \mathbf{1}_D) \ .$$

Inserting this into Eq. 5.20, we find

$$\tilde{\mathbf{z}}^{new} = \mathbf{z}^{new} + \epsilon M^T (M M^T)^{-1} =: \mathbf{z}^{new} + \epsilon' \ ,$$

where we have defined $\epsilon' := \epsilon M^T (M M^T)^{-1} \sim \mathcal{N}(0, |\mathcal{L}^{new}| \sigma^2 (M M^T)^{-1})$. Using the eigenvalue decomposition of the symmetric matrix $M M^T = U \Lambda U^T$, the distribution of ϵ' can be rewritten as

$$\epsilon' \sim U \mathcal{N}(0, |\mathcal{L}^{new}| \sigma^2 \Lambda^{-1}) U = U(|\mathcal{L}^{new}|)^2 \Lambda^{-2} \cdot \mathcal{N}(0, \sigma^2) \ .$$

Note that Λ scales with the squared 2-norm of the mean vectors μ, which typically scales with the number of dimensions D. For the special case when $U = \mathbf{1}_D$, we then have

$$P[\text{error}] \quad = \quad 1 - \prod_{k \in \mathcal{L}^{new}} \left(1 - \Phi \left(\frac{\tau - 1}{\sigma \sqrt{|\mathcal{L}^{new}| \Lambda_{kk}^{-1}}} \right) \right)$$

where $\Phi(\cdot)$ is the cumulative distribution function of the standardized normal distribution. Summing up, the probability of an error due to the heuristic decreases whenever the dimensionality grows (Λ_{kk} grows), sources become more concentrated (σ gets smaller), or the degree of the true label set decreases ($|\mathcal{L}^{new}|$ grows).

For a given classification task, $\mathcal{L}^{new}$ is unknown. In our experiments, we derived an upper limit M_{max} for the label degree from the distribution of the label set degrees in the training set. Furthermore, we used the average eigenvalue $\bar{\lambda}$ of the eigenvalue decomposition of MM^T to estimate Λ_{kk}. Finally, σ can be estimated from the variance of the single labeled data. With these estimates, we finally get

$$P[\text{error}] \leq 1 - \left(1 - \Phi\left(\frac{\tau - 1}{\sigma\sqrt{p_{\max}\bar{\lambda}^{-1}}}\right)\right)^{M_{max}} \tag{5.21}$$

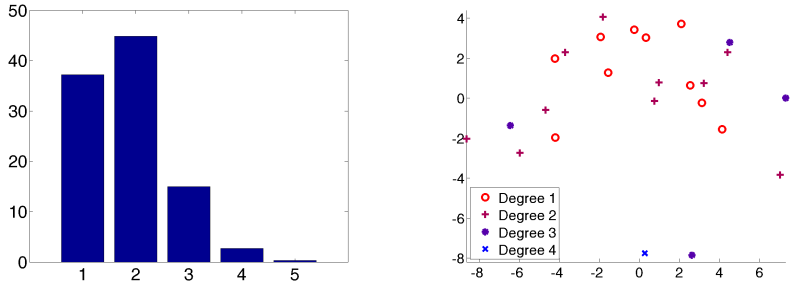
Given an acceptable error probability, this allows us to choose the threshold τ. Note that the bound is typically quite pessimistic, as most of the real-world data samples have a large number of data with label sets of small degree. For these data items, the effective error probability is much lower than indicated by (5.21). Keeping this in mind, we get a reasonable error bound also in the case where $U \neq \mathbf{1}_D$.

5.4 Experimental Evaluation

We present experiments on artificial and real-world data with multiple labels. We start with experiments on synthetic data in order to determine the accuracy of the parameter estimators and then proceed to real-world acoustic data.

5.4.1 Experiments on Synthetic Data

We use artificial data sampled from multivariate Gaussian distributions to compute the accuracy of the source parameter estimates of different models. The artificial data scenario consists of 10 sources labeled $\{1, \ldots, 10\}$. In order to avoid hidden assumptions or effects of hand-chosen parameters, the mean values of the sources are chosen uniformly from the 10-dimensional hypercube $[-2; 2]^{10}$. The covariance matrix is diagonal with diagonal elements uniformly sampled from $]0; 1]$. 25 different subsets of $\{1, \ldots, 10\}$



(a) Distribution of label set degrees $|\mathcal{L}|$ in the synthetic data

(b) Projection of the true Source centroids: First two Principal Directions

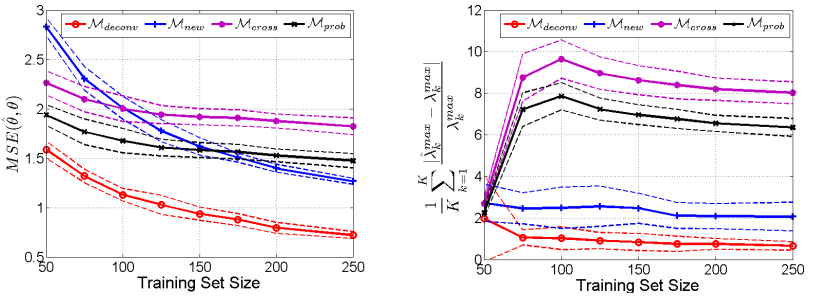
Figure 5.1: Statistics on the synthetic data. The first two principal directions depicted on the right panel cover approximately 60% of the variance of the centroids.

are randomly chosen and used as label sets. The distribution of the label degrees as well as the first two principal components of a sample of source centroids are depicted in Figure 5.1. As the principal component projections in Fig. 5.1(b) represent approximately 60% of the variation of the centroids, this classification problem is a challenging one.

Training sets of different sizes as well as a test set are sampled based on the label sets and the additivity assumption (Eq. 5.9). This procedure is repeated 10 times to average the results over different instantiations of the random variables.

Figure 5.2 shows the average deviation of the mean vectors and the average deviation of the largest eigenvalue from the corresponding true values. For the estimates of the source means (Figure 5.2(a)), it can be clearly seen that deconvolutive training is the most accurate. The deviation of the parameters of new training is explained by the small effective sample size available to estimate each of the mean vectors: As $\mathcal{M}_{new}$ learns a separate source for each label set, there are only two samples per source when the training set size is 50. $\mathcal{M}_{deconv}$, on the other hand, decomposes the contributions of each source to every data item. On the average, $\mathcal{M}_{deconv}$ has thus 2.5 times more training samples per parameter than $\mathcal{M}_{new}$. Furthermore, the samples used by $\mathcal{M}_{new}$ to estimate the density distribution of multi-labeled data have higher variance than the single label data.

For the estimation of the covariance (Figure 5.2(b)), $\mathcal{M}_{deconv}$ still yields



(a) Average Deviation of the Mean Estimator

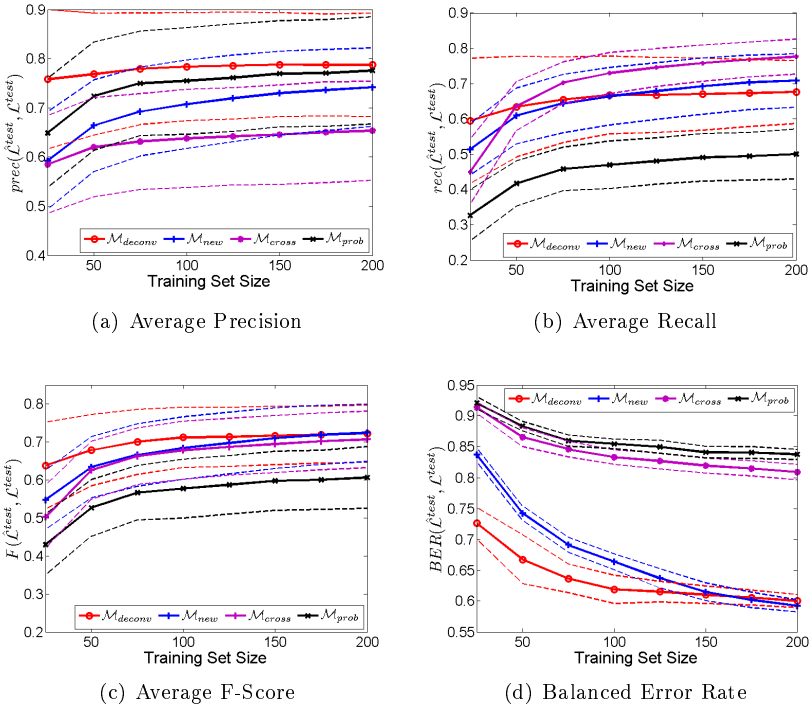
(b) Relative Deviation of the Largest Eigenvalue of the Covariance Matrix

Figure 5.2: Accuracy of the parameter estimation of different models. For each model, the average (continuous bold line) over all classes and the standard deviation based on 10-fold cross-validation (dashed lines) is plotted. We used a setting with 10 sources in 10 dimensions. The mean of each source is chosen uniformly in $[-2, 2]^{10}$. The sources are randomly combined to 25 label sets. Training data sets of different sizes are then sampled according to the generative model.

distinctly more precise values, but the difference to $\mathcal{M}_{new}$ is not as large as in the estimation of the mean values. This is due to the more complicated optimization problem that has to be solved to estimate the covariance matrix.

The estimates obtained by $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$ for both the mean and the covariance are clearly less accurate. Using a data item with multiple label as a training sample independently for each class brings the source parameters closer to each other, and thus away from their true values. As multi-labeled data have a reduced weight for the estimation of the single sources, this effect is less pronounced in $\mathcal{M}_{prob}$ than in $\mathcal{M}_{cross}$. As the estimator for the covariance matrix depends on the estimator of the mean, the large deviations of the dominant eigenvalue are a consequence of the inaccurate mean estimator.

The estimation of the covariance matrix is generally known as a hard problem [101]. As no analytic solution of the optimality condition exists and numerical methods have to be used, the computational effort to estimate the covariance grows linearly in the number of dimensions when learning



(a) Average Precision

(b) Average Recall

(c) Average F-Score

(d) Balanced Error Rate

Figure 5.3: Classification performance on synthetic data.

diagonal covariance matrices and quadratically if a full covariance matrix is assumed. Only for spherical covariances, the conditions can be solved to get a set of coupled explicit equations, which can be used for an iterative solution scheme. A possible remedy is to estimate the source covariances based on single label data only and to use the deconvolution approach only for estimating the mean values, or to assume the same covariance matrix for all sources.

The estimation of the source means is much more stable and it performs independently of the dimensionality of the data. As expected, the gain of $\mathcal{M}_{deconv}$ as compared to $\mathcal{M}_{new}$ is larger if the covariance matrix does not have to be estimated, and also the improvements in the classification are more pronounced.

For classification (see Figure 5.3), the differences between the four stud-

ied methods is most pronounced for the balanced error rate, which is the most rigorous quality measure. Here, $\mathcal{M}_{deconv}$ clearly outperforms all competitors when only a limited number of training data is available. For larger training data sets, $\mathcal{M}_{new}$ is able to catch up and reaches the same performance as $\mathcal{M}_{deconv}$. Both $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$ yield clearly worse results.

The difference between the methods are less pronounced when looking at precision and recall, as the variance of these quality measures is clearly higher then for the balanced error rate. Apart from the different number of training data per parameter, $\mathcal{M}_{new}$ behaves comparably to $\mathcal{M}_{deconv}$. For $\mathcal{M}_{prob}$, we observe a cautious assignment of labels: This method reaches almost the same precision as $\mathcal{M}_{deconv}$, but a recall which is significantly below the recall of all other methods. $\mathcal{M}_{cross}$, on the contrary, obtains the highest recall values of all methods in most settings, but pays the price of a low precision. For the F-score, we find $\mathcal{M}_{cross}$ very close to $\mathcal{M}_{new}$, which both are behind $\mathcal{M}_{deconv}$ for small and medium-sized training sets. $\mathcal{M}_{prob}$ follows with a relatively large gap.

This comparison clearly shows that the quality assessment of multi-label classification techniques depends largely on the quality measure. However, $\mathcal{M}_{deconv}$ yields the best results in all cases. For the F-score, this is a trend, while for the balanced error rate, this is highly significant for small training data sets in comparison to all other methods, and highly significant in comparison to $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$ for all training set sizes.

5.4.2 Experiments on Acoustic Data

For the experiments on real data, we use the research database provided by the hearing instrument company Phonak. This challenging data set serves as benchmark for next generation hearing instruments and captures the large variety of acoustic environments that are typically encountered by a hearing instrument user. It contains audio streams of every day acoustic scenes recorded with state of the art hearing instruments.

Each sound clip is assigned to one of the four classes *Speech* (SP), *Speech in Noise* (SN), *Noise* (NO) and *Music* (MU). While $\mathcal{M}_{new}$ learns a separate source for each of the four label sets, $\mathcal{M}_{cross}$, $\mathcal{M}_{prob}$ and $\mathcal{M}_{deconv}$ interpret SN as a mixture of SP and NO. SN is the only multi-label in our real data setting. As mentioned before, the intra-class variance is very high — just consider various genres of music, or different sources of noise! Additionally, mixtures arise in different proportions, i.e. the noise level in the mixture class varies strongly between different sound clips. All these

factors render the classification problem a difficult challenge: Even with specially designed features and a large training data set, the accuracy is at most 75%. Precision, recall and the F-score are around 80%.

Mel Frequency Cepstral Coefficients (MFCCs) [88] have been extracted from the sound clips at a rate of about 100Hz, yielding a 13-dimensional feature vector per time window. As classification is expected to be independent of the signal volume, the intensity of the sound files is normalized. Thus, the additivity assumption (Eq. 5.9) is changed to

$$x_{SN} = \frac{x_{SP} + x_{NO}}{2} \tag{5.22}$$

Since the extraction of MFCCs is nonlinear, this modified additivity property in the signal space has been transformed into the feature space. A sequence of 10 MFCC feature sets is used as feature vector, describing also the short-time evolution of the signal. Features for the training and test sets have been extracted from different sound clips.

Hidden Markov models (HMM) are widely used in signal processing and speech recognition [89]. We use a factorial HMM [50] with Gaussian output and two states per sound source a simple generative model. In the training phase, we use the approximations

$$\begin{aligned} \mathbb{E}_{\Xi\sim P_{NO}}[P_{SP}(x_n - \Xi)] \approx & P_{SP}(x_n - \mathbb{E}_{\chi\sim P_{NO}}[\Xi]) \\ \mathbb{E}_{\Xi\sim P_{SP}}[P_{NO}(x_n - \Xi)] \approx & P_{NO}(x_n - \mathbb{E}_{\Xi\sim P_{SP}}[\Xi]) \end{aligned} \tag{5.23}$$

to get a rough estimate of the individual source contributions to a data item x_n with label $\mathcal{L}_n = SN = \{SP, NO\}$. In the classification phase, the formulation of the convolution as expected value (Eq. 5.12) is used to estimate the probability of the binary label by sampling from one of the two contributing sources.

Experiments are cross-validated 10 times. In every cross validation round, the number of training samples is gradually increased from 4 (i.e. one per label set) to 60. The differences in F-score and BER are depicted in Fig. 5.4. The test sets consist of 255 data items.

Comparing the results of the four algorithms on the test data set, we observe only minor differences in the precision, with $\mathcal{M}_{deconv}$ tending to yield slightly less precise results. The recall rate of $\mathcal{M}_{deconv}$, however, is consistently higher than the corresponding results of its three competitors. The F-score obtained by the deconvolutive multi-label classifier is consistently above the F-scores obtained by $\mathcal{M}_{new}$, $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$. As can be

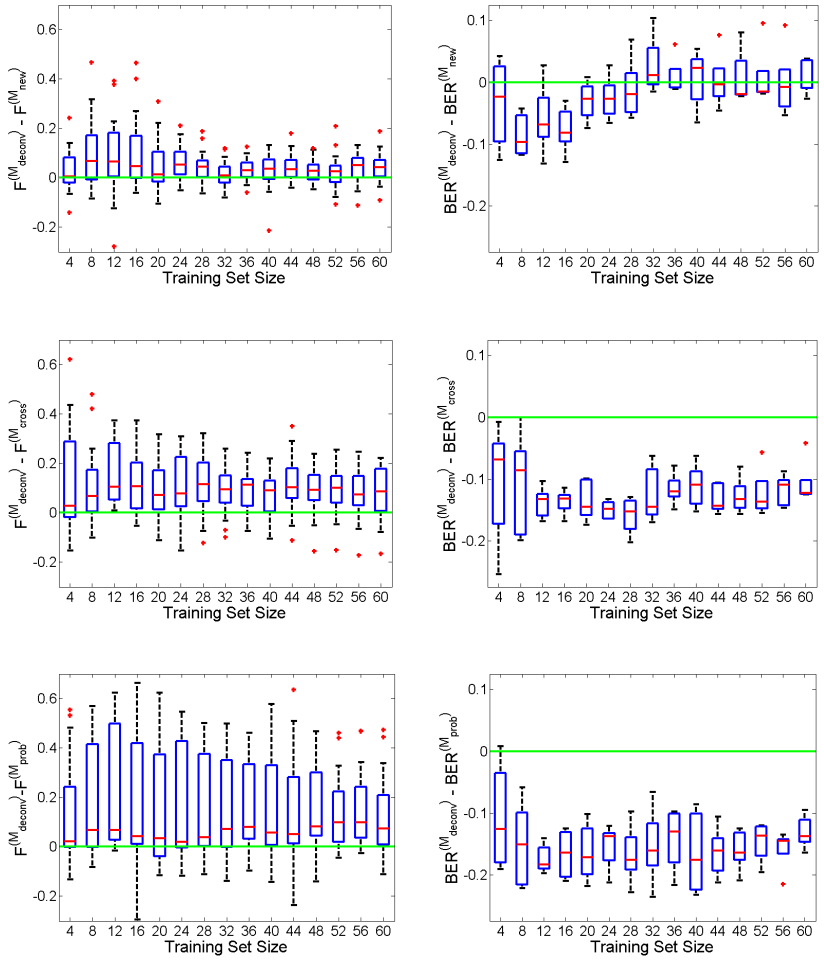


Figure 5.4: Difference of quality measures between the proposed method and the three mentioned competing methods. The left column shows the differences in F-Score (higher is better), the right one the differences in BER (lower is better). The absolute values are around 0.6 for the F-score and around 0.4 for the BER at the very small sample sizes. In all plots, the green horizontal line at 0 indicates equal performance of the two compared algorithms. Note the difference in scale between the two columns.

observed in the plots, $\mathcal{M}_{new}$ approaches $\mathcal{M}_{deconv}$ as the size of the training set increases. The difference between $\mathcal{M}_{deconv}$ and the two other models does not shows a clear dependency on the size of the training set.

Differences are more pronounced in terms of the balanced error rate BER (depicted in the right column of Figure 5.4). $\mathcal{M}_{new}$ is clearly outperformed on small training sets, but it is able to perform competitively as more training data are available. For larger training sets, learning a separate, independent class for the multi-labeled data as $\mathcal{M}_{new}$ does, sometimes even performs slightly better, as multi-label data might not exactly fulfill the additivity condition. Independently of the training set size, both $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$ are clearly performing worse than $\mathcal{M}_{deconv}$. These results confirm the observations made on synthetic data: $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$ suffer from unmatched model assumptions, and $\mathcal{M}_{new}$ has too few training data if the training set size is limited.

Sophisticated Models for Speech Separation. The experimental setting used in the experiments on acoustic data is rather limited. We briefly present some ideas to improve the performance of the classification algorithm on acoustic data, i.e. the identification of sources.

The improvements presented below allow us to address a more complex problem in digital signal processing: *source separation*. Given a mixture of signals, the objective is to determine the original signals. An accurate generative model for acoustic sources not only yields more precise classification results, but also renders *model-based source separation* possible.

Features. As discussed in Chapter 4, Mel-frequency cepstral coefficients (MFCC) are not particularly suited to distinguish several types of acoustic signals. One usually uses the log-power spectrum of the signal as features. These are approximately distributed according to a Gaussian distribution and allow us to re-synthesize a signal (up to a chosen frequency) based on the feature values [72].

Decomposition of Source Contributions. The approximation used in our experiments (see Eq. 5.23) to estimate the contribution of individual sources to a given mixture is quite rough. The Algonquin algorithm [48] is an efficient technique to accurately approximate the mixture of two sources emissions described by the log-power spectrum.

Gain Estimation. In our setup, we have implicitly assumed that the emissions of several sources are mixed together at identical intensities. This

assumption is not realistic in real-world scenarios. A heuristic to estimate the gain out of a predefined set of gains of synthetic mixtures in presented in [58].

More Accurate Dynamics. If the three aforementioned improvements have been implemented, a more detailed modeling of the temporal dynamics is beneficial for both recognition and separation of sources.

Furthermore, when the application is limited to speech signals, an additional grammar model describing the structure of a correct sentence further improves the performance of the speech separation algorithm. Combining these ingredients, model-based source separation is able to outperform human listeners in recognizing speech on synthetic mixtures of two speakers [58] and yields currently the best results in both source recognition and source separation. This achievement shows the power of elaborate generative models of the sort we proposed in the field of acoustics.

Chapter 6

Asymptotic Analysis of Estimators on Multi-Label Data

Asymptotic theory in statistics refers to the limiting distribution of a summary statistic when the amount of data items over which the statistic is computed increases to infinity [12]. It has become an essential tool in statistics, as the exact distributions of the quantities of interest is not available in most settings. In the first place, asymptotic analysis is used to check whether an estimation method is consistent, i.e. whether the obtained estimators converge to the correct parameter values if the number of data items available for inference goes to infinity. Furthermore, asymptotic theory provides approximate answers where exact ones are not available, namely in the case of data sets of finite size, and describes for example how efficiently an inference method uses the given data for parameter estimation [78].

Consistent inference schemes are essential for generative classifiers, and a more efficient inference scheme yields more precise classification results than a less efficient one given the same training data. More specifically, for maximum a posteriori classification, if the estimated parameters converge to the true parameter values, the expected error of a classifier converges to the Bayes error [34].

In this chapter, we first review the state-of-the-art asymptotic theory for estimators based on single-label data. We then extend the asymptotic

Table 6.1: Overview over the probability distributions used in this chapter. A data item $D = (X, \mathcal{L})$ is an observation X along with its label set $\mathcal{L}$.

Symbol	Meaning
$P_{\theta_k}(\Xi_k)$	true distribution of the emissions of source k, given θ_k
$P_{\boldsymbol{\theta}}(\boldsymbol{\Xi})$	true joint distribution of the emissions of all sources.
$P_{\mathcal{L},\boldsymbol{\theta}}(X)$	true distribution of the observations X with label set $\mathcal{L}$.
$P^{\mathcal{M}}_{\mathcal{L},\boldsymbol{\theta}}(X)$	distribution of the observation X with label set $\mathcal{L}$, as assumed by method $\mathcal{M}$.
$P_{\mathcal{L},\mathbf{D}}(X)$	empirical distribution of observation X with label set $\mathcal{L}$ in the data set $\mathbf{D}$.
$P_{\pi}(\mathcal{L})$	true distribution of the label sets
$P_{\mathbf{D}}(\mathcal{L})$	empirical distribution of the label sets in $\mathbf{D}$
$P_{\boldsymbol{\theta}}(D)$	true distribution of data item D.
$P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)$	distribution of data item D as assumed by method $\mathcal{M}$.
$P_{\mathbf{D}}(D)$	empirical distribution of data items D in the data set $\mathbf{D}$
$P^{\mathcal{M}}_{D,\theta_k}(\Xi_k)$	Conditional distribution of the emission Ξ_k of source k given θ_k and D, as assumed by inference method $\mathcal{M}$.
$P^{\mathcal{M}}_{D,\boldsymbol{\theta}}(\boldsymbol{\Xi})$	Conditional distribution of the source emissions $\boldsymbol{\Xi}$ given $\boldsymbol{\theta}$ and D, as assumed by inference method $\mathcal{M}$.

analysis to inference on multi-label data and proof statements about the identifiability of parameters and the asymptotic distribution of their estimators in this demanding setting. We apply our result to two scenarios encountered in our real-world problems and thus confirm the theoretical results as well as the more accurate parameter estimation of deconvolutive training.

6.1 Preliminaries

In this section, we introduce the preliminaries to study of the asymptotic behavior of the estimators obtained by different inference methods. This chapter contains some relatively heavy notation. The probability distributions used therein are summarized in Table 6.1.

6.1.1 Exponential Family Distributions

In the following, we assume that the source distributions are members of the exponential family. This means that the distribution $P_{\theta_k}(\Xi_k)$ of source k admits a density $p_{\theta_k}(\xi_k)$ which can be written in the following form:

$$p_{\theta_k}(\xi_k) = \exp\left(\langle \theta_k, \phi(\xi_k) \rangle - A(\theta_k)\right) \ . \tag{6.1}$$

Here θ_k are the natural parameters, $\phi(\xi_k)$ are the sufficient statistics of the sample ξ_k of source k, and $A(\theta_k)$ is the *log-partition function*, defined as

$$A(\theta_k) := \log\left(\int \exp\left(\langle \theta_k, \phi(\xi_k) \rangle\right) \, \mathrm{d}\xi_k\right) \ .$$

The expression $\langle \theta_k, \phi(\xi_k) \rangle$ denotes the inner product between the natural parameters θ_k and the sufficient statistics $\phi(\xi_k)$:

$$\langle \theta_k, \phi(\xi_k) \rangle := \sum_{s=1}^{S} \theta_{k,s} \cdot \left(\phi(\xi_k)\right)_s \ .$$

The number S is called the dimensionality of the exponential family. $\theta_{k,s}$ is the s^{th} dimension of the parameter vector of source k, and $\left(\phi(\xi_k)\right)_s$ is the s^{th} dimension of the sufficient statistics. The (S-dimensional) parameter space of the distribution is denoted by Θ.

The class of exponential family distributions contains many of the widely used probability distributions. The Bernoulli, Poisson and the χ^2 distribution are one-dimensional exponential family distributions; the Gamma, Beta and normal distribution are examples of two-dimensional exponential family distributions.

The joint distribution of emissions $\boldsymbol{\Xi}$ of the independent sources in the source set $\mathcal{K} = \{1, \ldots, K\}$ is given by

$$P_{\boldsymbol{\theta}}(\boldsymbol{\Xi}) = \prod_{k=1}^{K} P_{\boldsymbol{\theta}_k}(\Xi_k)$$

with the density function

$$p_{\boldsymbol{\theta}}(\boldsymbol{\xi}) = \prod_{k=1}^{K} p_{\theta_k}(\xi_k) = \prod_{k=1}^{K} \exp\left(\langle \theta_k, \phi(\xi_k) \rangle - A(\theta_k)\right)$$

In order to make the notation more compact, we define the vectorial sufficient statistic as $\boldsymbol{\phi}(\boldsymbol{\xi}) := (\phi(\xi_1), \ldots, \phi(\xi_K))^T$ and the parameter vector as $\boldsymbol{\theta} := (\theta_1, \ldots, \theta_K)^T$. The cumulative log-partition function is defined as $A(\boldsymbol{\theta}) := \sum_{k=1}^{K} A(\theta_k)$. Using the parameter vector $\boldsymbol{\theta}$ and the emission vector $\boldsymbol{\xi}$, the density function $p_{\boldsymbol{\theta}}$ of the source emissions can then be written as

$$p_{\boldsymbol{\theta}}(\boldsymbol{\xi}) = \prod_{k=1}^{K} p_{\theta_k}(\xi_k) = \exp\left(\langle \boldsymbol{\theta}, \boldsymbol{\phi}(\boldsymbol{\xi}) \rangle - A(\boldsymbol{\theta})\right) \ .$$

The product of independent exponential family distributions is thus again a member of the exponential family.

Exponential family distributions have the property that the derivatives of the log-partition function with respect to the parameter vector $\boldsymbol{\theta}$ are moments of sufficient statistics $\boldsymbol{\phi}(\cdot)$. Namely the first and second derivative of $A(\cdot)$ are the expected first and second moment of the statistics:

$$\nabla_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\Xi} \sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \qquad \nabla^2_{\boldsymbol{\theta}} A(\boldsymbol{\theta}) = \mathbb{V}_{\boldsymbol{\Xi} \sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \tag{6.2}$$

where $\mathbb{E}_{X \sim P}[X]$ and $\mathbb{V}_{X \sim P}[X]$ denote the expectation value and the covariance matrix of a random variable X sampled from distribution P.

6.1.2 Identifiability

The representation of exponential family distributions in Eq. 6.1 may not be unique, e.g. if the sufficient statistics $\phi(\xi_k)$ satisfy linear constraints. In this case, the dimensionality S of the exponential family distribution can be reduced. Unless this is done, the parameters θ_k are unidentifiable: There exist at least two values $\theta_k^{(1)} \neq \theta_k^{(2)}$ of the parameters which imply the same probability distribution $p_{\theta_k^{(1)}} = p_{\theta_k^{(1)}}$. These two parameter values can not be distinguished based on observations, they are therefore called *unidentifiable* [76].

Definition 1. ***(Identifiability)*** *Let $\wp = \{p_\theta : \theta \in \Theta\}$ be a statistical model with parameter space Θ. $\wp$ is called* identifiable *if the mapping $\theta \to p_\theta$ is one-to-one:*

$$p_{\theta^{(1)}} = p_{\theta^{(2)}} \iff \theta^{(1)} = \theta^{(2)} \quad \textit{for all } \theta^{(1)}, \theta^{(2)} \in \Theta \ .$$

Identifiability of the model in the sense that the mapping $\theta \to p_\theta$ can be inverted is equivalent to being able to learn the true parameters of the model if an infinite number of samples from the model can be observed [76].

In all concrete learning problems, identifiability is always conditioned on the data. Obviously, if there are no observations from a particular source (class), the likelihood of the data is independent of the parameter values of the never-occurring source. The parameters of the particular source are thus unidentifiable.

6.1.3 M- and Z-Estimators

A popular method to determine the estimators $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_K)$ for a generative model based on independent and identically-distributed (*i.i.d.*) data items $\mathbf{D} = (D_1, \ldots, D_N)$ is to maximize a criterion function of the type

$$\boldsymbol{\theta} \mapsto M_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} m_{\boldsymbol{\theta}}(D_n) \tag{6.3}$$

Here $m_{\boldsymbol{\theta}} : \mathcal{D} \mapsto \mathbb{R}$ are known functions. An estimator $\hat{\boldsymbol{\theta}}$ maximizing $M_N(\boldsymbol{\theta})$ is called an *M-estimator*: $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} M_N(\boldsymbol{\theta})$, where M stands for *maximization*.

For continuously differentiable criterion functions, the maximizing value is often determined by setting the derivative (or, in the multidimensional case, the set of partial derivatives) with respect to $\boldsymbol{\theta}$ equal to zero. With $\psi_{\boldsymbol{\theta}}(D) := \nabla_{\boldsymbol{\theta}} m_{\boldsymbol{\theta}}(D)$, this yields an equation of the type

$$\Psi_N(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \psi_{\boldsymbol{\theta}}(D_n) \ , \tag{6.4}$$

and the parameter $\boldsymbol{\theta}$ is then determined such that $\Psi_N(\boldsymbol{\theta}) = 0$. This type of estimator is called *Z-estimator*, with Z standing for *zero*.

Maximum Likelihood Estimators. Maximum likelihood estimators are M-estimators with the criterion function $m_{\boldsymbol{\theta}}(D) := \ell(\boldsymbol{\theta}; D)$. The corresponding Z-estimator is obtained by computing the derivative of the log-likelihood with respect to the parameter vector $\boldsymbol{\theta}$, called the *score*:

$$\psi_{\boldsymbol{\theta}}(D) = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; D) \ . \tag{6.5}$$

In the following, we mostly use the formulation of maximum likelihood estimators as Z-estimators.

Convergence of M- and Z-estimators. Assume that there exist asymptotic criterion functions $\boldsymbol{\theta} \mapsto M(\boldsymbol{\theta})$ and $\boldsymbol{\theta} \mapsto \Psi(\boldsymbol{\theta})$ such that

$$M_N(\boldsymbol{\theta}) \xrightarrow{P} M(\boldsymbol{\theta}) \qquad \Psi_N(\boldsymbol{\theta}) \xrightarrow{P} \Psi(\boldsymbol{\theta}) \qquad \text{for every } \boldsymbol{\theta} \ .$$

The maximizer $\hat{\boldsymbol{\theta}}_N$ of M_N converges to the maximizing value $\boldsymbol{\theta}_0$ of M as N goes to infinity if the deviation between $M_N(\hat{\boldsymbol{\theta}}_N)$ and $M_N(\boldsymbol{\theta})$ converges to 0 in probability and if there is a unique, well-separated maximizer $\boldsymbol{\theta}_0$ of M [115]:

Theorem 1. *Let M_N be random functions and let M be a fixed function of $\boldsymbol{\theta}$ such that for every $\epsilon > 0$*

$$\sup_{\boldsymbol{\theta} \in \Theta} |M_N(\boldsymbol{\theta}) - M(\boldsymbol{\theta})| \xrightarrow{P} 0 \qquad \sup_{\boldsymbol{\theta}: d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) \geq \epsilon} M(\boldsymbol{\theta}) < M(\boldsymbol{\theta}_0) \ .$$

Then, any sequence of estimators $\hat{\boldsymbol{\theta}}_N$ with $M_N(\hat{\boldsymbol{\theta}}_N) \geq M_N(\boldsymbol{\theta}_0) + o_P(1)$ converges in probability to $\boldsymbol{\theta}_0$.

The notation $o_P(1)$ denotes a sequence of random vectors that converge to 0 in probability.

An equivalent theorem can be found for Z-estimators by applying Theorem 1 to the functions $M_N(\boldsymbol{\theta}) = -||\Psi_N(\boldsymbol{\theta})||$ and $M(\boldsymbol{\theta}) = -||\Psi(\boldsymbol{\theta})||$:

Theorem 2. *Let Ψ_N be random vector-valued functions and let Ψ be a fixed vector-valued function of $\boldsymbol{\theta}$ such that for every $\epsilon > 0$*

$$\sup_{\boldsymbol{\theta} \in \Theta} ||\Psi_N(\boldsymbol{\theta}) - \Psi(\boldsymbol{\theta})|| \xrightarrow{P} 0 \qquad \inf_{\boldsymbol{\theta}: d(\boldsymbol{\theta}, \boldsymbol{\theta}_0) \geq \epsilon} ||\Psi(\boldsymbol{\theta})|| > ||\Psi(\boldsymbol{\theta}_0)|| = 0 \ .$$

Then any sequence of estimators $\hat{\boldsymbol{\theta}}_N$ such that $\Psi_N(\hat{\boldsymbol{\theta}}_N) = o_P(1)$ converges in probability to $\boldsymbol{\theta}_0$.

The second condition implies that $\boldsymbol{\theta}_0$ is the only zero of $\Psi(\cdot)$ outside a neighborhood of size ϵ around $\boldsymbol{\theta}_0$. Since we assumed the criterion function $\Psi(\cdot)$ to be continuous, it must be either strictly negative or strictly positive for all $\boldsymbol{\theta}$ which are more than ϵ away from $\boldsymbol{\theta}_0$. Reverting to the antiderivative $M(\boldsymbol{\theta})$ of $\Psi(\boldsymbol{\theta})$, this implies that $M(\boldsymbol{\theta})$ must be concave over the whole parameter space Θ.

Asymptotic Normality. Given consistency, the question about how the estimators $\boldsymbol{\theta}_N$ are distributed around the asymptotic limit $\boldsymbol{\theta}_0$ arises. Assuming the criterion function $\boldsymbol{\theta} \mapsto \psi_{\boldsymbol{\theta}}(D)$ to be twice continuously differentiable, $\Psi_N(\hat{\boldsymbol{\theta}}_N)$ can be expanded through a Taylor series around $\boldsymbol{\theta}_0$. Together with the central limit theorem, the estimator $\boldsymbol{\theta}_N$ is found to be normally distributed around $\boldsymbol{\theta}_0$ [115]. Defining, for a more compact notation, $\boldsymbol{v}^{\otimes}$ as the outer product of the vector $\boldsymbol{v}$, i.e. $\boldsymbol{v}^{\otimes} := \boldsymbol{v}\boldsymbol{v}^T$, we get the following theorem:

Theorem 3. *Assume that* $\mathbb{E}_D[\psi_{\boldsymbol{\theta}_0}(D)^{\otimes}] < \infty$ *and that the map* $\boldsymbol{\theta} \mapsto \mathbb{E}_D[\psi_{\boldsymbol{\theta}}(D)]$ *is differentiable at a zero* $\boldsymbol{\theta}_0$ *with non-singular derivative matrix. Then, the sequence* $\sqrt{n} \cdot (\hat{\boldsymbol{\theta}}_N - \boldsymbol{\theta}_0)$ *is asymptotically normal:*

$$\sqrt{N} \cdot (\hat{\boldsymbol{\theta}}_N - \boldsymbol{\theta}_0) \rightsquigarrow \mathcal{N}(0, \Sigma) \ , \tag{6.6}$$

with asymptotic variance Σ

$$\Sigma = (\mathbb{E}_D[\nabla_{\boldsymbol{\theta}} \psi_{\boldsymbol{\theta}_0}(D)])^{-1} \cdot \mathbb{E}_D\left[(\psi_{\boldsymbol{\theta}_0}(D))^{\otimes}\right] \cdot (\mathbb{E}_D[\nabla_{\boldsymbol{\theta}} \psi_{\boldsymbol{\theta}_0}(D)])^{-T} \ . \tag{6.7}$$

All expectation values are taken with respect to the true distribution of the data items D.

6.1.4 Maximum-Likelihood Estimation on Single-Label Data

To estimate parameters based on single-label data, a collection of data $\mathbf{D} = \{(X_1, \lambda_1), \ldots, (X_N, \lambda_N)\}$, $\lambda_n \in \{1, \ldots, K\}$ for all $n = 1, \ldots, N$, is separated according to the class label, so that one gets K sets $\mathbf{X}_1, \ldots, \mathbf{X}_K$, where $\mathbf{X}_k$ contains all observations with label k, formally $\mathbf{X}_k := \{X_n | (X_n, \lambda_n) \in \mathbf{D}, \lambda_n = k\}$. All samples in $\mathbf{X}_k$ are assumed to be *i.i.d.* random variables distributed according to $P(X|\theta_k)$. It is assumed that the samples in $\mathbf{D}_k$ do not provide any information about $\theta_{k'}$ if $k \neq k'$, i.e. parameters for the different classes are assumed to be functionally independent of each other [39]. Therefore, inference can be done independently for each class, yielding K separate parameter estimation problems. In each problem, the criterion function is

$$\Psi_{N_k}(\theta_k) = \frac{1}{N_k} \sum_{X \in \mathbf{X}_k} \psi_{\theta_k}((X, k)) \ , \tag{6.8}$$

where $N_k := |\mathbf{X}_k|$ is the number of data items with label k. The parameter estimator $\hat{\theta}_k$ is then determined such that $\Psi_{N_k}(\theta_k) = 0$.

More specifically for maximum likelihood estimation of parameters of exponential family distributions (Eq. 6.1), the criterion function $\psi_{\theta_k}(\cdot) = \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta}; D)$ (Eq. 6.5) for a data item $D = (X, \{k\})$ becomes

$$\psi_{\theta_k}(D) = \phi(X) - \mathbb{E}_{\Xi_k \sim P_{\theta_k}}[\phi(\Xi_k)] \ . \tag{6.9}$$

Choosing the $\hat{\theta}_k$ such that the criterion function $\Psi_{N_k}(\theta_k)$ is zero means changing the model parameter such that the average value of the sufficient statistics of the observations coincides with the expected sufficient statistics of the source distributions:

$$\Psi_{N_k}(\theta_k) = \frac{1}{N_k} \sum_{X \in \mathbf{X}_k} \phi(X) - \mathbb{E}_{\Xi_k \sim P_{\theta_k}}[\phi(\Xi_k)] \, . \tag{6.10}$$

Hence, maximum likelihood estimators in exponential families are moment estimators [118]. The theorems of consistency and asymptotic normality are directly applicable.

With the same formalism, it becomes clear why the inference problems for different classes are independent: Assume an observation X with label k is given. Under the assumption of the generative model, the label k states that X is a sample from source p_{θ_k}. Trying to derive information about the parameter $\theta_{k'}$ of a second source $k' \neq k$ from X, we would derive p_{θ_k} with respect to $\theta_{k'}$ to get the score function. Since p_{θ_k} is independent of $\theta_{k'}$, this derivative is zero, and the data item (X, k) does not contribute to the criterion function $\Psi_{N_{k'}}(\theta_{k'})$ (Eq. 6.10) for the parameter $\theta_{k'}$.

Fisher Information. For inference in a parametric model with a consistent estimator $\hat{\theta}_k \to \theta_k$, the Fisher information matrix $\mathcal{I}$ is defined as the second moment of the score function. Since the parameter estimator $\hat{\theta}$ is chosen such that average of the score function is zero, the second moment of the score function corresponds to the variance of the sufficient statistic $\phi(\cdot)$:

$$\mathcal{I}_{\mathbf{X}_k}(\theta_k) := \mathbb{E}_{X \sim P_{\theta_k^G}}\left[\psi_{\theta_k}(X)^{\otimes}\right] = -\mathbb{V}_{X \sim P_{\theta_k^G}}[\boldsymbol{\phi}(X)] \ , \tag{6.11}$$

where the expectation is taken with respect to the true distribution $P_{\theta_k^G}$. The Fisher Information thus indicates to what extend the score function depends on the parameter. The larger this dependency is, the more the provided data depends on the parameter value, and the more accurately this parameter value can be determined for a given set of training data.

The information of independent experiments or data sets is additive. For two disjoint data sets $\mathbf{D}_1$ and $\mathbf{D}_2$ we thus have

$$\mathcal{I}_{\mathbf{X}_k^{(1)},\mathbf{X}_k^{(2)}}(\theta_k) = \mathcal{I}_{\mathbf{X}_k^{(1)}}(\theta_k) + \mathcal{I}_{\mathbf{X}_k^{(2)}}(\theta_k) \ . \tag{6.12}$$

The additivity property of the information allows us to specify the contribution of different subsets of data in the parameter estimation based on multi-labeled data.

According to the Cramér-Rao bound [90, 31, 30], the inverse Fisher information is a lower bound on the variance of any estimator of a deterministic parameter. A consistent estimator for the parameter θ_k is called *efficient* if $\Sigma_k = \mathcal{I}_{\mathbf{X}_k}(\theta_k)^{-1}$.

Under regularity conditions [115], which are fulfilled by maximum likelihood estimators, the asymptotic variance Σ in Eq. 6.7 becomes

$$\Sigma = \mathcal{I}_{\mathbf{X}}(\boldsymbol{\theta})^{-1} \ , \tag{6.13}$$

i.e. the maximum likelihood estimators are efficient.

6.2 Asymptotic Distribution of Multi-Label Estimators

In this section, we extend the analysis to estimators based on multi-label data. We restrict ourselves to maximum likelihood estimators for the parameters of exponential family distributions. Since we are mainly interested in comparing different ways to learn from data, we also assume the parametric form of the distribution to be known, and that the inference step consists of estimating the parameters of a distribution in the exponential family.

6.2.1 From Observations to Source Emissions

In single-label inference problems, each observation provides a sample of a source indicated by the label, as discussed in Section 6.1.4. In the case of inference based on multi-label data, the situation is more involved, since the source emissions can not be observed directly. The relation between the source emissions and the observations are formalized by the combination function (see Section 2.1), which describes the observation X obtained based on an emission vector $\boldsymbol{\Xi}$.

To do inference, one needs to determine which emission vector $\boldsymbol{\Xi}$ has yielded the observed X. To solve this inverse problem, an inference method relies on further assumptions besides the assumption on the type of distribution, namely on the combination function. These design assumptions — made implicitly or explicitly — enable the inference scheme to derive information about the distribution of the source emissions given an observation.

In this analysis, we focus on differences in the assumed combination function. We denote by $P^{\mathcal{M}}(X|\boldsymbol{\Xi}, \mathcal{L})$ the probability distribution of an observation X given the emission vector $\boldsymbol{\Xi}$ and the label set $\mathcal{L}$ as assumed by method $\mathcal{M}$. $P^{\mathcal{M}}(X|\boldsymbol{\Xi}, \mathcal{L})$ denotes the probabilistic representation of the combination function. We formally describe several techniques along with the analysis of their estimators in Section 6.3. It is worth mentioning that for single-label data, all estimation techniques considered in this work are equal and yield consistent and efficient parameter estimators, as they agree on the combination function for single-label data: The identity function is the only reasonable choice in this case.

The probability distribution of X given the label set $\mathcal{L}$, the parameters $\boldsymbol{\theta}$ and the combination function assumed by method $\mathcal{M}$ can be computed by marginalizing $\boldsymbol{\Xi}$ out of the joint distribution of $\boldsymbol{\Xi}$ and X:

$$P^{\mathcal{M}}_{\mathcal{L},\boldsymbol{\theta}}(X) := P^{\mathcal{M}}(X|\mathcal{L}, \boldsymbol{\theta}) = \int P^{\mathcal{M}}(X|\boldsymbol{\xi}, \mathcal{L}) \,\mathrm{d}P(\boldsymbol{\Xi}|\boldsymbol{\theta}) \,. \tag{6.14}$$

For the probability of a data item $D = (X, \mathcal{L})$ given the parameters $\boldsymbol{\theta}$ under the assumptions made by model $\mathcal{M}$, we have

$$P^{\mathcal{M}}_{\boldsymbol{\theta}}(D) := P^{\mathcal{M}}(X, \mathcal{L}|\boldsymbol{\theta}) = P(\mathcal{L}) \cdot P^{\mathcal{M}}(X|\mathcal{L}, \boldsymbol{\theta}) \tag{6.15}$$

$$= \pi_{\mathcal{L}} \cdot \int P^{\mathcal{M}}(X|\boldsymbol{\Xi}, \mathcal{L}) p(\boldsymbol{\Xi}|\boldsymbol{\theta}) \,\mathrm{d}\boldsymbol{\Xi} \,. \tag{6.16}$$

Estimating the probability of the label set $\mathcal{L}$, $\pi_{\mathcal{L}}$, is a standard problem of estimating the parameters of a multinomial distribution [41]. According to the central limit theorem, the empirical frequency of occurrence converges to the true probability for each label set. Therefore, we do not further investigate this estimation problem and assume that the true value of $\pi_{\mathcal{L}}$ can be determined for all $\mathcal{L} \in \mathbb{L}$.

The probability of a particular emission vector $\boldsymbol{\Xi}$ given a data item D and the parameters $\boldsymbol{\theta}$ is computed using Bayes' theorem:

$$P^{\mathcal{M}}_{D,\boldsymbol{\theta}}(\boldsymbol{\Xi}) := P^{\mathcal{M}}(\boldsymbol{\Xi}|X, \mathcal{L}, \boldsymbol{\theta}) = \frac{P^{\mathcal{M}}(X|\boldsymbol{\Xi}, \mathcal{L}) \cdot P(\boldsymbol{\Xi}|\boldsymbol{\theta})}{P^{\mathcal{M}}(X|\mathcal{L}, \boldsymbol{\theta})} \tag{6.17}$$

We point out that the distribution $P^{\mathcal{M}}(\boldsymbol{\Xi}|D,\boldsymbol{\theta})$ might depend on the parameters $\boldsymbol{\theta}$. This means that the estimation of the contributions of a source may depend on the parameters of a different source. Consider for example the case where you observe the sum of emissions from two Gaussian distributions: The distribution of the emissions of one source depends on the estimated mean and variance of the other source. More generally, while two emissions Ξ_1, Ξ_2 are assumed to be independent, this independence is lost once we condition on the observation X.

The distribution $P^{\mathcal{M}}(\boldsymbol{\Xi}|D,\boldsymbol{\theta})$ describes the essential difference between inference methods for multi-label data. For an inference method $\mathcal{M}$ which assumes that an observation X is a sample from each source contained in the label set $\mathcal{L}$, $P^{\mathcal{M}}(\Xi_k|D,\boldsymbol{\theta})$ is a point mass (Dirac mass) at X. For methods which assume that several emission vectors are mapped to the same observation, and $P^{\mathcal{M}}(\boldsymbol{\Xi}|D,\boldsymbol{\theta})$ is a non-degenerate density function.

6.2.2 Conditions for Identifiability

As in the standard scenario of learning from single-label data, parameter inference is only possible if there is a one-to-one relation between the parameters $\boldsymbol{\theta}$ and the distribution $P_{\boldsymbol{\theta}}$. Conversely, parameters are unidentifiable if $\boldsymbol{\theta}^{(1)} \neq \boldsymbol{\theta}^{(2)}$, but $P_{\boldsymbol{\theta}^{(1)}} = P_{\boldsymbol{\theta}^{(2)}}$. For our setting as specified in Eq. 6.16, this is the case if

$$\sum_{n=1}^{N} \log\left(\pi_{\mathcal{L}_n} \int P^{\mathcal{M}}(X_n|\boldsymbol{\xi},\mathcal{L}_n) p(\boldsymbol{\xi}|\boldsymbol{\theta}^{(1)})\,\mathrm{d}\boldsymbol{\xi}\right) = \sum_{n=1}^{N} \log\left(\pi_{\mathcal{L}_n} \int P^{\mathcal{M}}(X_n|\boldsymbol{\xi},\mathcal{L}_n) p(\boldsymbol{\xi}|\boldsymbol{\theta}^{(2)})\,\mathrm{d}\boldsymbol{\xi}\right)$$

but $\boldsymbol{\theta}^{(1)} \neq \boldsymbol{\theta}^{(2)}$. The following situations imply such a scenario:

- A particular source k never occurs in the label set, formally

  $$|\{\mathcal{L} \in \mathbb{L} | k \in \mathcal{L}\}| = 0 \quad \text{or} \quad \pi_{\mathcal{L}} = 0 \quad \forall \mathcal{L} \in \mathbb{L} : \mathcal{L} \ni k$$

  This is the trivial case — one can not infer the parameters of a source without observing emissions from that source. In such a case, the probability of the observed data (Eq. 6.16) is invariant of the parameters θ_k of source k.

- The combination function ignores all (!) emissions of a particular source k. Thus, under the assumptions of the inference method $\mathcal{M}$, the emission Ξ_k of source k never has an influence on the observation. Hence, the combination function does not depend on Ξ_k. If this is the case for all $\mathcal{L}$, no information on the source k can be obtained from the data.

- The data available for inference does not support distinguishing different parameters of a pair of sources. Assume for example that source 2 only occurs together with source 1, i.e. for all n with $2 \in \mathcal{L}_n$, we also have $1 \in \mathcal{L}_n$. Unless the combination function is such that information can be derived about the emissions Ξ_1 and Ξ_2 of both sources 1 and 2 for some of the data items, there is a set of parameters θ_1 and θ_2 for the two sources that yields the same likelihood.
  Consider for example two sources with Gaussian distributions with parameters $\theta_k = (\mu_k, \sigma_k^2)$ for $k = 1, 2$ and the addition as combination function. Observations with label $\{1, 2\}$ are then distributed according to $\mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$. If labels 1 and 2 always occurs together, the value of the sums can be estimated, but there is no possibility to determine the values of the parameters (μ_1, σ_1^2) and (μ_2, σ_2^2) of the individual sources.

If the distribution of a particular source is unidentifiable, the assumption that the source in question exists is questionable. More specifically, in the first two cases, there is empirically no evidence for the existence of a source which is either never observed or has no influence on the data. In the last case, one might wonder whether the two classes 1 and 2 are really separate entities, or whether it might be more reasonable to merge them to a single class.

6.2.3 Maximum Likelihood Estimation on Multi-Label Data

Based on the probability of a data item D given the parameter vector $\boldsymbol{\theta}$ under the assumptions of the inference method $\mathcal{M}$ (Eq. 6.16) and using a uniform prior over the parameters, the log-likelihood of a parameter $\boldsymbol{\theta}$ given a data item $D = (X, \mathcal{L})$ is then given by $\ell^{\mathcal{M}}(\boldsymbol{\theta}; D) = \log\left(P^{\mathcal{M}}(X, \mathcal{L}|\boldsymbol{\theta})\right)$. Using the particular properties of exponential family distributions (Eq. 6.2),

we get the following expression for the score function:

$$\begin{aligned}\psi_{\boldsymbol{\theta}}^{\mathcal{M}}(D) &= \nabla \ell^{\mathcal{M}}(\boldsymbol{\theta}; D) && (6.18)\\ &= \mathbb{E}_{\boldsymbol{\Xi}\sim P_{D,\boldsymbol{\theta}}^{\mathcal{M}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \nabla A(\boldsymbol{\theta}) \\ &= \mathbb{E}_{\boldsymbol{\Xi}\sim P_{D,\boldsymbol{\theta}}^{\mathcal{M}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\,. && (6.19)\end{aligned}$$

Comparing the score function with the score function obtained in the single-label case (Eq. 6.10), the difference in the first term becomes apparent. While the first term is the sufficient statistic of the observation in the previous case, we now find the expected value of the sufficient statistic of the emissions, conditioned on $D = (X, \mathcal{L})$. This formulation contains the single-label setting as a special case: Given the single-label observation X with label k, we are sure that the k^{th} source has emitted X, i.e. $\Xi_k = X$. In the more general case of inference on multi-label data, several emission vectors $\boldsymbol{\Xi}$ might have produced the observed X. The distribution of these emission vectors (given the data item D and the parameter vector $\boldsymbol{\theta}$) is given by Eq. 6.17. The expectation of the sufficient statistics of the emissions with respect to this distribution now plays the role of the sufficient statistic of the observation in the single-label case.

As in the single-label case, we assume that several emissions are independent given their sources. The likelihood and the criterion function for a full data set $\mathbf{D} = (D_1, \ldots, D_N)$ thus factorize:

$$\ell^{\mathcal{M}}(\boldsymbol{\theta}; \mathbf{D}) = \sum_{n=1}^{N} \ell^{\mathcal{M}}(\boldsymbol{\theta}; D_n) \qquad \Psi_N^{\mathcal{M}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{n=1}^{N} \psi_{\boldsymbol{\theta}}^{\mathcal{M}}(D_n) \qquad (6.20)$$

In the following, we analyze estimators $\hat{\boldsymbol{\theta}}_N^{\mathcal{M}}$ which are Z-estimators, i.e. obtained by setting $\Psi_N^{\mathcal{M}}(\hat{\boldsymbol{\theta}}_N^{\mathcal{M}}) = 0$. We analyze the asymptotic behavior of the criterion function $\Psi_N^{\mathcal{M}}$ and derive conditions for consistent estimators. Afterwards, we compute the convergence rate of the estimator to the true value of the parameter.

6.2.4 Asymptotic Behavior of the Estimation Equation

We first analyze the criterion function as defined in Eq. 6.20. Note that the N observations used to estimate $\Psi_N^{\mathcal{M}}(\boldsymbol{\theta}_0^{\mathcal{M}})$ come from a mixture of distributions specified by the label sets. Using the *i.i.d.* assumption (Eq. 6.20),

and defining $\mathbf{D}_{\mathcal{L}} := \{(X', \mathcal{L}') \in \mathbf{D} | \mathcal{L}' = \mathcal{L}\}$, we have

$$\begin{aligned}\Psi_N^{\mathcal{M}}(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{n=1}^{N} \psi_{\boldsymbol{\theta}}^{\mathcal{M}}(D_n) \\ &= \frac{1}{N}\sum_{\mathcal{L}\in\mathbb{L}}\sum_{D\in\mathbf{D}_{\mathcal{L}}} \psi_{\boldsymbol{\theta}}^{\mathcal{M}}(D) \\ &= \frac{1}{N}\sum_{\mathcal{L}\in\mathbb{L}} |\mathbf{D}_{\mathcal{L}}| \frac{1}{|\mathbf{D}_{\mathcal{L}}|}\sum_{D\in\mathbf{D}_{\mathcal{L}}} \psi_{\boldsymbol{\theta}}^{\mathcal{M}}(D) \end{aligned} \tag{6.21}$$

Denote by $N_{\mathcal{L}} := |\mathbf{D}_{\mathcal{L}}|$ the number of training samples with label set $\mathcal{L}$, and by $P_{\mathcal{L},\mathbf{D}}$ the empirical distribution of observations with label set $\mathcal{L}$. Then,

$$\frac{1}{N_{\mathcal{L}}}\sum_{D\in\mathbf{D}_{\mathcal{L}}} \psi_{\boldsymbol{\theta}}^{\mathcal{M}}(D) = \mathbb{E}_{X\sim P_{\mathcal{L},\mathbf{D}}}\left[\psi_{\boldsymbol{\theta}}^{\mathcal{M}}((X,\mathcal{L}))\right]$$

is an average of independent, identically distributed random variables. By the central limit theorem, this empirical average converges to the true average as the number of data items, $N_{\mathcal{L}}$, goes to infinity:

$$\mathbb{E}_{X\sim P_{\mathcal{L},\mathbf{D}}}\left[\psi_{\boldsymbol{\theta}}^{\mathcal{M}}((X,\mathcal{L}))\right] \rightsquigarrow \mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\psi_{\boldsymbol{\theta}}^{\mathcal{M}}((X,\mathcal{L}))\right] \ . \tag{6.22}$$

Furthermore, we define $\hat{\pi}_{\mathcal{L}} := \frac{N_{\mathcal{L}}}{N}$. Again by the central limit theorem, we have $\hat{\pi}_{\mathcal{L}} \rightsquigarrow \pi_{\mathcal{L}}$. Inserting (6.22) into (6.21), we get

$$\begin{aligned}\Psi_N^{\mathcal{M}}(\boldsymbol{\theta}) &= \sum_{\mathcal{L}\in\mathbb{L}} \hat{\pi}_{\mathcal{L}}\mathbb{E}_{X\sim P_{\mathcal{L},\mathbf{D}}}\left[\psi_{\boldsymbol{\theta}}^{\mathcal{M}}((X,\mathcal{L}))\right] \\ &\rightsquigarrow \sum_{\mathcal{L}\in\mathbb{L}} \pi_{\mathcal{L}}\mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\psi_{\boldsymbol{\theta}}^{\mathcal{M}}((X,\mathcal{L}))\right] \end{aligned} \tag{6.23}$$

Plugging in the value of the score function as derived in Eq. 6.19 into the asymptotic behavior of the criterion function for Z-estimators (Eq. 6.23), we get

$$\begin{aligned}\Psi_N^{\mathcal{M}}(\boldsymbol{\theta}) &\rightsquigarrow \sum_{\mathcal{L}\in\mathbb{L}} \pi_{\mathcal{L}}\mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\psi_{\boldsymbol{\theta}}^{\mathcal{M}}((X,\mathcal{L}))\right] \\ &= \mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P_{D,\boldsymbol{\theta}}^{\mathcal{M}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] - \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \end{aligned} \tag{6.24}$$

This expression shows that also for inference based in multi-label data, the maximum likelihood estimator is a moment estimator. However, the

source emissions can not be observed directly, and the expected value of its sufficient statistic takes its place, where the average is taken with respect to the distribution of the source emissions assumed by the inference method $\mathcal{M}$.

6.2.5 Conditions for Consistent Estimators

In this section, we list conditions under which the estimator $\hat{\boldsymbol{\theta}}_N^{\mathcal{M}}$, determined as a zero of $\Psi_N^{\mathcal{M}}(\boldsymbol{\theta})$, is consistent.

Theorem 4 (Consistency of Estimators)**.** *If the inference method $\mathcal{M}$ uses the true conditional distribution of the source emissions $\boldsymbol{\Xi}$ given data items, i.e. $P^{\mathcal{M}}(\boldsymbol{\Xi}|(X,\mathcal{L}),\boldsymbol{\theta}) = P^{G}(\boldsymbol{\Xi}|(X,\mathcal{L}),\boldsymbol{\theta})$ for all data items $D = (X,\mathcal{L})$, then the estimator $\hat{\boldsymbol{\theta}}_N$ determined as a zero of $\Psi_N^{\mathcal{M}}(\boldsymbol{\theta})$, as defined in Eq. 6.24, is consistent.*

Proof. The true parameter of the generative process, denoted by $\boldsymbol{\theta}^G$, is a zero of $\Psi^G(\boldsymbol{\theta})$, the criterion function derived from the true generative model. According to Theorem 2, a necessary condition for consistency of $\hat{\boldsymbol{\theta}}_N^{\mathcal{M}}$ is

$$\sup_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} ||\Psi_N^{\mathcal{M}}(\boldsymbol{\theta}) - \Psi^G(\boldsymbol{\theta})|| \xrightarrow{P} 0 \ .$$

Inserting the expression for the criterion function as derived in Eq. 6.24, we get the condition

$$\begin{aligned}
&\Big\| \mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}} \Big[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \Big] - \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \\
&\qquad - \mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}} \Big[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{G}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \Big] + \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \Big\| \\
&= \Big\| \mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}} \Big[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \Big] - \mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}} \Big[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{G}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \Big] \Big\| = 0 \ . \qquad (6.25)
\end{aligned}$$

Separating the generative process for the data items $D \sim P_{\boldsymbol{\theta}^G}$ into a separate generation of the label set $\mathcal{L}$ and an observation X, $\mathcal{L} \sim P_{\pi^G}$, $X \sim P_{\mathcal{L},\boldsymbol{\theta}^G}$, the condition in Eq. 6.25 is fulfilled if

$$\sum_{\mathcal{L}\in\mathbb{L}} \pi_{\mathcal{L}} \mathbb{E}_{X\sim P^{G}_{\mathcal{L},\boldsymbol{\theta}^G}} \Big[\Big\| \mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{(X,\mathcal{L}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{\boldsymbol{\Xi}\sim P^{G}_{(X,\mathcal{L}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \Big\| \Big] = 0 \ . \qquad (6.26)$$

Using the assumption that $P^{\mathcal{M}}_{(X,\mathcal{L}),\boldsymbol{\theta}} = P^{G}_{(X,\mathcal{L}),\boldsymbol{\theta}}$ for all data items $D = (X,\mathcal{L})$, this condition is trivially fulfilled. □

Differences between $P^{\mathcal{M}}_{D^\delta,\boldsymbol{\theta}}$ and $P^{G}_{D^\delta,\boldsymbol{\theta}}$ for some data items $D^\delta = (X^\delta, \mathcal{L}^\delta)$, on the other hand, have no effect on the consistency of the result if either the probability of D^δ is zero, or if the expected value of the sufficient statistics is identical for the two distributions. The first situation implies that either the label set $\mathcal{L}^\delta$ never occurs in any data item, or the observation X^δ never occurs with label set $\mathcal{L}$. The second situation implies that the parameters are unidentifiable. Hence, we formulate the stronger conjecture that if inference procedure which yields inconsistent estimators on data with a particular label set, its overall parameter estimators are inconsistent.

As we show later in Section 6.3, ignoring all multi-label data yields consistent estimators. However, discarding a possibly large part of the data is not efficient, which motivates the quest for more advanced inference techniques to retrieve information about the source parameters from multi-label data. However, advanced models entail the risk to assume a criterium function which yields inconsistent estimators. We discuss an example of a class of such criterion functions in Chapter 7.

6.2.6 Efficiency of Parameter Estimation

Given that an estimator $\hat{\boldsymbol{\theta}}$ is consistent, the next question of interest concerns the rate at which the deviation from the true parameter value converges to zero. This rate is given by the asymptotic variance of the estimator in Eq. 6.7. In the following, we compute the asymptotic variance specifically for maximum likelihood estimators. This analysis allows us to compare different inference techniques which yield consistent estimators in terms of how efficiently they use the provided data set for inference.

Generalized Fisher Information. The Fisher information is introduced to measure the information content of an data item about the parameters of the source that are assumed to have generated the data. In multi-label classification, the definition of the Fisher information (Eq. 6.11) has to be extended, as the source emissions are only indirectly observed. We define the *Generalized Fisher Information* as follows:

Definition 2. ***Generalized Fisher Information.*** *The General Fisher Information $\mathcal{I}_{\mathcal{L}}$ measures the amount of information a data item $D = (X, \mathcal{L})$ with label set $\mathcal{L}$ contain about the parameter vector $\boldsymbol{\theta}$:*

$$\mathcal{I}_{\mathcal{L}} := \mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}}}\left[\mathbb{V}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] \tag{6.27}$$

The term $\mathbb{V}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$ measures the uncertainty about the source emissions $\boldsymbol{\Xi}$, given a data item D. This term vanishes if and only if the data item D completely determines the source emission(s) of all involved sources. In the other extreme case where the data item D does not reveal any information about the source emissions, this is equal to $\mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$, and the generalized Fisher information is thus 0.

Asymptotic Variance. We are now ready to determine the asymptotic variance of an estimator.

Theorem 5 (Asymptotic Variance)**.** *Denote by $P^{\mathcal{M}}_{D,\boldsymbol{\theta}}(\boldsymbol{\Xi})$ the distribution of the emission vector $\boldsymbol{\Xi}$ given the data item D and the parameters $\boldsymbol{\theta}$ under the assumptions made by the inference method $\mathcal{M}$. Furthermore, let $\mathcal{I}_{\mathcal{L}}$ denote the generalized Fisher information of data with label set $\mathcal{L}$. Then, the asymptotic variance of the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ is given by*

$$\Sigma = \left(\mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}]\right)^{-1} \cdot \left(\mathbb{V}_D\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right) \cdot \left(\mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}]\right)^{-1} \ , \tag{6.28}$$

where all expectations and variances are computed with respect to the true distribution.

Proof. We derive the asymptotic variance based on Theorem 3 on asymptotic normality of M-estimators. The first and last factor in Eq. 6.7 are the derivative of the criterion function $\psi^{\mathcal{M}}_{\boldsymbol{\theta}}(D)$ as defined in Eq. 6.18:

$$\nabla_{\boldsymbol{\theta}}\psi^{\mathcal{M}}_{\boldsymbol{\theta}}(D) = \nabla^2_{\boldsymbol{\theta}}\ell^{\mathcal{M}}(\boldsymbol{\theta}; D) = \frac{\nabla^2_{\boldsymbol{\theta}} P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)}{P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)} - \left(\frac{\nabla P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)}{P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)}\right)^{\otimes} \ . \tag{6.29}$$

The particular properties of the exponential family distributions imply

$$\begin{aligned}\frac{\nabla^2 P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)}{P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)} = &\left(\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right)^{\otimes} \\ &+ \mathbb{V}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\end{aligned} \ . \tag{6.30}$$

With $\nabla P^{\mathcal{M}}_{\boldsymbol{\theta}}(D)/P^{\mathcal{M}}_{\boldsymbol{\theta}}(D) = \psi^{\mathcal{M}}_{\boldsymbol{\theta}}(D)$ and using Eq. 6.19, we get

$$\nabla\psi^{\mathcal{M}}_{\boldsymbol{\theta}}(D) = \mathbb{V}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \ .$$

Taking the expectation over the data items D, we get the expected generalized Fisher information matrix over all label sets:

$$\mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}[\nabla\psi_{\boldsymbol{\theta}}(D)] = \mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{V}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] - \mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}] \ . \tag{6.31}$$

For the middle term of Eq. 6.7, we have

$$\mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}\left[(\psi_{\boldsymbol{\theta}}(D))^{\otimes}\right] = \mathbb{V}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] + \left(\mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] - \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right)^{\otimes}$$

The condition for $\hat{\boldsymbol{\theta}}$ given in Eq. 6.24 implies

$$\mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}\left[(\psi_{\boldsymbol{\theta}}(D))^{\otimes}\right] = \mathbb{V}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\xi})]\right] \tag{6.32}$$

Using Eq. 6.7 we get the expression for the asymptotic variance of the estimator $\boldsymbol{\theta}$ stated in the theorem. □

According to this result, the asymptotic variance of the estimator is determined by two factors. We analyze them in the following two subsections and afterwards derive some well-known results for special cases.

Bias-Variance Decomposition

We define the expectation-deviance for label set $\mathcal{L}$ as the difference between the expected value of the sufficient statistics under the distribution assumed by method $\mathcal{M}$, given observations with label set $\mathcal{L}$, and the expected value of the sufficient statistic given all data items:

$$\Delta\mathbb{E}^{\mathcal{M}}_{\mathcal{L}} := \mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{D'\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D',\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] \tag{6.33}$$

The middle factor (Eq. 6.32) of the estimator variance is the variance in the expectation values of the sufficient statistics of the emission vectors $\boldsymbol{\Xi}$. Using the identity $\mathbb{E}_X\left[X^2\right] = \mathbb{E}_X[X]^2 + \mathbb{V}_X[X]$, and splitting $D = (X, \mathcal{L})$ into the observation X and the label set $\mathcal{L}$, it can be decomposed as

$$\begin{aligned}&\mathbb{V}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] \\ &\quad= \mathbb{E}_{\mathcal{L}\sim P_\pi}\left[\left(\Delta\mathbb{E}^{\mathcal{M}}_{\mathcal{L}}\right)^{\otimes}\right] + \mathbb{E}_{\mathcal{L}}\left[\mathbb{V}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] .\end{aligned} \tag{6.34}$$

This decomposition shows that two independent effects can cause a high variance of the estimator:

1. The expected value of the sufficient statistics of the source emissions based on observations with a particular label $\mathcal{L}$ deviates from the

true parameter value. Note that this effect can be present even if the estimator is consistent: These deviations of sufficient statistics conditioned on a particular label set might cancel out each other when averaging over all label sets and thus yield a consistent estimator. However, an estimator obtained by such a procedure has a higher variance than an estimator which is obtained by a procedure which yields consistent estimators also conditioned on every label set.

2. The expected value of the sufficient statistics of the source emissions given the observation X varies with X. This contribution is typically large for one-against-all methods [95].

Note that for inference methods which fulfill the conditions of Theorem 4, we have $\Delta\mathbb{E}_{\mathcal{L}}^{\mathcal{M}} = 0$. Methods which yield consistent estimators on any label set are thus not only provably consistent, but also yield parameters with less variation.

Special Cases

In the following, we focus on some special cases in which the above result reduces to well-known results.

Variance of Estimators on Single-Label Data: If estimation is based on single-label data, the source emissions are fully determined by the available data, as the observations are considered to be direct emissions of the respective source. Formally, with $D = (X, \mathcal{L})$ and $\mathcal{L} = \{\lambda\}$, we thus have

$$P_{D,\boldsymbol{\theta}}^{\mathcal{M}}(\boldsymbol{\Xi}) = \prod_{k=1}^{K} P_{D,\theta_k}^{\mathcal{M}}(\Xi_k) \ , \quad \text{with } P_{D,\theta_k}^{\mathcal{M}}(\Xi_k) = \begin{cases} 1_{\{\Xi_k = X\}} & \text{if } k = \lambda \\ P(\Xi_k|\theta_k) & \text{otherwise} \end{cases}$$

The estimation procedure is thus independent for every source k. Furthermore, we have

$$\mathbb{E}_{\Xi_k \sim P_{D,\theta_k}^{\mathcal{M}}}[\phi(\Xi_k)] = X \qquad \mathbb{V}_{\Xi_k \sim P_{D,\theta_k}^{\mathcal{M}}}[\phi(\Xi_k)] = 0 \ .$$

Hence, Σ is a diagonal matrix, with diagonal elements

$$\Sigma_{kk} = \mathcal{I}_{\{k\}}^{-1} \left(\mathbb{V}_{X \sim P_{\theta_k^G}^G}[\phi(X)] + \left(\mathbb{E}_{X \sim P_{\theta_k^G}^G}[\phi(X)] - \mathbb{E}_{\Xi_k \sim P_{\theta_k}}[\phi(\Xi_k)] \right)^{\otimes} \right) \mathcal{I}_{\{k\}}^{-1}$$

Variance of Consistent Estimators: For consistent estimators, we have

$$\mathbb{E}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] = \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$$

and thus

$$\Sigma = (\mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}])^{-1} \cdot \mathbb{V}_{D\sim P_{\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{\mathcal{M}}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] \cdot (\mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}])^{-1} \ .$$

Variance of Consistent Estimators on Single-Label Data: Combining the two aforementioned conditions, we get

$$\Sigma_{\lambda\lambda} = \mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]^{-1} = \mathcal{I}_{\mathbf{X}_\lambda}(\theta_\lambda) \ , \tag{6.35}$$

which corresponds to the well-known result for inference based on single-label data obtained in Eq. 6.13.

6.3 Asymptotic Analysis of Multi-Label Inference Methods

In this section, we formally describe techniques for inference based on multi-labeled data and apply the results obtained in Section 6.2 to study the asymptotic behavior of estimators obtained with these methods.

6.3.1 Ignore Training ($\mathcal{M}_{ignore}$)

The ignore training is probably the simplest, but also the most limited way of treating multi-label data: Data items which belong to more than one class are simply ignored [11], i.e. the estimation of source parameters is uniquely based on single-label data. Thus reducing the inference problem to a single-label problem, the overall probability of an emission vector $\boldsymbol{\Xi}$ given the data item D factorizes as

$$P^{ignore}_{D,\boldsymbol{\theta}}(\boldsymbol{\Xi}) = \prod_{k=1}^{K} P^{ignore}_{D,\boldsymbol{\theta},k}(\Xi_k) \tag{6.36}$$

Each of the factors $P^{ignore}_{D,\boldsymbol{\theta},k}(\Xi_k)$, representing the probability distribution of source k, only depends on the parameter θ_k, i.e. we have $P^{ignore}_{D,\boldsymbol{\theta},k}(\Xi_k) = P^{ignore}_{D,\theta_k}(\Xi_k)$ for all $k = 1,\ldots,K$. A data item $D = (X,\mathcal{L})$ does only provide

information about source k if $\mathcal{L} = \{k\}$. In the case $\mathcal{L} \neq \{k\}$, the probability distribution of emissions Ξ_k, $P^{ignore}_{D,\hat{\theta}_k}(\Xi_k)$, is not influenced by the data item D, i.e. it maintains its value given the current parameter estimator $\hat{\theta}_k$.

$$P^{ignore}_{D,\hat{\theta}_k}(\Xi_k) = \begin{cases} 1_{\{\Xi_k = X\}} & \text{if } \mathcal{L} = \{k\} \\ P^{ignore}_{\hat{\theta}_k}(\Xi_k) & \text{otherwise} \end{cases} \tag{6.37}$$

Observing a multi-label data items does not change the assumed probability distribution of any of the classes. These data items are thus treated as uninformative by the method $\mathcal{M}_{ignore}$.

Deriving the log-likelihood function implied by Equations 6.36 and 6.37 with respect to the parameter θ_k, we obtain the following criterion function given a data item D:

$$\psi^{ignore}_{\boldsymbol{\theta}}(D) = \sum_{k=1}^{K} \psi^{ignore}_{\theta_k}(D) \tag{6.38}$$

with

$$\psi^{ignore}_{\theta_k}(D) = \begin{cases} \phi(X) - \mathbb{E}_{\Xi_k \sim P_{\hat{\theta}_k}}[\phi(\Xi_k)] & \text{if } \mathcal{L} = \{k\} \\ 0 & \text{otherwise} \end{cases} \tag{6.39}$$

The estimator $\hat{\boldsymbol{\theta}}^{ignore}$ is consistent and normally distributed:

Lemma 1. *The estimator $\hat{\boldsymbol{\theta}}^{ignore}_N$ determined as a zero of $\Psi^{ignore}_N(\boldsymbol{\theta})$ as defined in Eq. 6.20 and Eq. 6.39 is distributed according to*

$$\sqrt{N} \cdot (\hat{\boldsymbol{\theta}}^{ignore}_N - \boldsymbol{\theta}^G) \to \mathcal{N}(0, \Sigma^{ignore}) \ . \tag{6.40}$$

The covariance matrix Σ^{ignore} is given by

$$\Sigma^{ignore} = diag\left(\Sigma^{ignore}_{11}, \ldots, \Sigma^{ignore}_{KK}\right) \tag{6.41}$$

with the matrices on the diagonal given by

$$\Sigma^{ignore}_{kk} = \mathbb{V}_{X \sim P_{\theta_k}}\left[\psi^{ignore}_{\theta_k}((X, \{k\}))\right]^{-1} \ . \tag{6.42}$$

This statement follows directly from Theorem 3 about the asymptotic distribution of estimators based on single-label data. A formal proof is given in Section A.1 in the appendix.

6.3.2 New Source Training ($\mathcal{M}_{new}$)

New source training defines new meta-classes for each label set such that every data item belongs to a single class (in terms of these meta-labels) [107]. Doing so, the number of parameters to be inferred is heavily increased as compared to the generative process.

We define the number of possible label sets as $L := |\mathbb{L}|$ and assume an arbitrary, but fixed, ordering of the possible label sets. Let $\mathbb{L}[l]$ be the l^{th} label set in this ordering. Then, we have:

$$P^{new}_{D,\boldsymbol{\theta}}(\boldsymbol{\Xi}) = \prod_{l=1}^{L} P^{new}_{D,\boldsymbol{\theta},l}(\Xi_l) \tag{6.43}$$

As for $\mathcal{M}_{ignore}$, each of the factors represents the probability distribution of one of the sources given the data item D. Hence

$$P^{new}_{D,\boldsymbol{\theta},l}(\Xi_l) = P^{new}_{D,\theta_l}(\Xi_l) = \begin{cases} 1_{\{\Xi_l = X\}} & \text{if } \mathcal{L} = \mathbb{L}[l] \\ P^{new}_{\mathcal{L},\theta_l}(\Xi_l) & \text{otherwise} \end{cases} \tag{6.44}$$

For the criterion function on a data item $D = (X, \mathcal{L})$, we thus have

$$\psi^{new}_{\boldsymbol{\theta}}(D) = \sum_{l=1}^{L} \psi^{new}_{\theta_l}(D) \tag{6.45}$$

$$\psi^{new}_{\theta_l}(D) = \begin{cases} \psi(X) - \mathbb{E}_{\Xi_l \sim P_{\theta_l}}[\psi(\Xi_l)] & \text{if } \mathcal{L} = \mathbb{L}[l] \\ 0 & \text{otherwise} \end{cases} \tag{6.46}$$

The estimator $\hat{\boldsymbol{\theta}}^{new}_N$ is consistent and normally distributed:

Lemma 2. *The estimator $\hat{\boldsymbol{\theta}}^{new}_N$ obtained as a zero of the criterion function $\Psi^{new}_N(\boldsymbol{\theta})$ is asymptotically distributed as*

$$\sqrt{N} \cdot (\hat{\boldsymbol{\theta}}^{new}_N - \boldsymbol{\theta}^G) \to \mathcal{N}(0, \Sigma^{new}) \ . \tag{6.47}$$

The covariance matrix is block-diagonal

$$\Sigma^{new} = diag\,(\Sigma^{new}_{11}, \ldots, \Sigma^{new}_{LL}) \tag{6.48}$$

with the diagonal elements given by

$$\Sigma^{new}_{ll} = \mathbb{V}_{X \sim P^{new}_{\mathbb{L}[l],\theta_l}} \left[\psi_{\theta^G_l}(X) \right]^{-1} \ . \tag{6.49}$$

Again, this corresponds to the result obtained for consistent single-label inference techniques in Eq. 6.35. We refer to Section A.1 for a proof.

The main drawback of this method is that there are typically not enough training data available to reliably estimate a parameter set for each label set. Furthermore, it is not possible to assign a new data item to a label set which is not seen in the training data.

6.3.3 Cross-Training ($\mathcal{M}_{cross}$)

Cross-training, as proposed in [11], takes each sample x which belongs to class k as an emission of class k – independently of which other labels the data item has. The probability of the source emission vector $\boldsymbol{\Xi}$ thus factorizes into a product over the probabilities of the different source emissions:

$$P^{cross}_{D,\boldsymbol{\theta}}(\boldsymbol{\Xi}) = \prod_{k=1}^{K} P^{cross}_{D,\boldsymbol{\theta},k}(\Xi_k) \tag{6.50}$$

The probability distribution of each source is assumed to be independent of all other sources, i.e. we have $P^{cross}_{D,\boldsymbol{\theta},k}(\Xi_k) = P^{cross}_{D,\theta_k}(\Xi_k)$, with

$$P^{cross}_{D,\theta_k}(\Xi_k) = \begin{cases} 1_{\{\Xi_k = X\}} & \text{if } k \in \mathcal{L} \\ P_{\theta_k}(\Xi_k) & \text{otherwise} \end{cases} \tag{6.51}$$

Again, $P^{cross}_{D,\theta_k} = P_{\theta_k}(\Xi_k)$ in the case $k \notin \mathcal{L}$ means that X does not provide any information about the assumed P_{θ_k}, i.e. the estimated distribution is unchanged. For the criterion function, we have

$$\psi^{cross}_{\boldsymbol{\theta}}(D) = \sum_{k=1}^{K} \psi^{cross}_{\theta_k}(D) \tag{6.52}$$

$$\psi^{cross}_{\theta_k}(D) = \begin{cases} \phi(X) - \mathbb{E}_{\Xi_k \sim P_{\theta_k}}[\phi(\Xi_k)] & \text{if } k \in \mathcal{L} \\ 0 & \text{otherwise} \end{cases} \tag{6.53}$$

The parameters obtained by $\mathcal{M}_{cross}$ are not consistent:

Lemma 3. *The estimator $\hat{\boldsymbol{\theta}}^{cross}$ obtained as a zero of the criterion function $\psi^{cross}_N(\boldsymbol{\theta})$ are inconsistent if the training data set contains at least one multi-label data item.*

The inconsistency is due to the fact that multi-label data items are used to estimate the parameters of all sources the data item belongs to without

considering the influence of the other sources. The bias of the estimator grows as the fraction of multi-label data increases. A formal proof is given in the appendix (Section A.1).

6.3.4 Deconvolutive Training ($\mathcal{M}_{deconv}$)

The deconvolutive training method aims at estimating the distribution of the source emissions given a data item $D = (X, \mathcal{L})$. We assume in the following that the true combination function is deterministic. Modeling the generative process, the distribution of an observation X given the emission vector $\boldsymbol{\Xi}$ and the label set $\mathcal{L}$ is given by

$$P^{deconv}(X|\boldsymbol{\Xi}, \mathcal{L}) = 1_{\{X=\mathrm{c}_\kappa^{deconv}(\boldsymbol{\Xi},\mathcal{L})\}} \tag{6.54}$$

Integrating out the source emissions, we obtain the probability of an observation x given the label set $\mathcal{L}$ and the parameter vector $\boldsymbol{\theta}$ as

$$P^{deconv}(X|\mathcal{L}, \boldsymbol{\theta}) = \int P(X|\Xi, \mathcal{L}) \, \mathrm{d}P(\Xi|\boldsymbol{\theta}) \tag{6.55}$$

Using Bayes' theorem and the above notation, we have:

$$P^{deconv}(\boldsymbol{\Xi}|D, \boldsymbol{\theta}) = \frac{P^{deconv}(X|\boldsymbol{\Xi}, \mathcal{L}) \cdot P^{deconv}(\boldsymbol{\Xi}|\boldsymbol{\theta})}{P^{deconv}(X|\mathcal{L}, \boldsymbol{\theta})} \tag{6.56}$$

If the true combination function is provided to the method, or the method can correctly estimate this function, then $P^{deconv}(\boldsymbol{\Xi}|D, \boldsymbol{\theta})$ corresponds to the true conditional distribution. The target function is given as

$$\psi_{\boldsymbol{\theta}}^{deconv}(D) = \mathbb{E}_{\boldsymbol{\Xi} \sim P_{D,\hat{\boldsymbol{\theta}}}^{deconv}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{\boldsymbol{\Xi} \sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \tag{6.57}$$

Unlike in the methods presented before, the combination function c used in $\mathcal{M}_{deconv}$ has to be provided along with the data. It is typically determined based on prior knowledge of the process which generates the data. For this reason, it is not possible to describe the distribution of the estimators obtained by this method in general. However, given the identifiability conditions discussed in Section 6.1.2, the parameter estimators converge to their true values.

6.4 Addition of Gaussian-Distributed Emissions

We consider the case of two univariate Gaussian distributions with variance σ^2, $\sigma > 0$. The sample space of the Gaussian distribution is given by $\mathbb{R}$, and the probability density function is

$$p(\xi) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\xi-\mu)^2}{2\sigma^2}\right) . \tag{6.58}$$

Mean and standard deviation of the k^{th} source are denoted by μ_k and σ_k, respectively, for $k = 1, 2$. Rearranging terms in order to write the Gaussian distribution as a member of the exponential family as in Eq. 6.1, we get

$$\begin{aligned} \theta_k &= \left(\frac{\mu_k}{{\sigma_k}^2}, -\frac{1}{2{\sigma_k}^2}\right)^T \\ T &= \left(x, x^2\right)^T \\ A(\theta_k) &= \frac{{\mu_k}^2}{2{\sigma_k}^2} - \ln\left(\frac{1}{\sigma_k}\right) = -\frac{{\theta_{k,1}}^2}{4\theta_{k,2}} - \ln\left(\sqrt{-2\theta_{k,2}}\right) \end{aligned}$$

The natural parameters θ are not the most common parametrization of the Gaussian distribution. However, the usual parameters (μ_k, σ_k^2) can be easily computed from the parameters θ_k:

$$-\frac{1}{2\sigma_k^2} = \theta_{k,2} \iff \sigma_k^2 = -\frac{1}{2\theta_{k,2}} \qquad \theta_{k,1} = \frac{\mu_k}{\sigma_k^2} \iff \mu_k = \sigma_k^2 \cdot \theta_{k,1} . \tag{6.59}$$

The parameter space is $\Theta = \{(\theta_1, \theta_2) \in \mathbb{R} | \theta_2 < 0\}$. In the following, we assume $\mu_1 = -a$ and $\mu_2 = a$. The parameters of the first and second source are thus given by

$$\theta_1 = \left(-\frac{a}{{\sigma_1}^2}, -\frac{1}{2{\sigma_1}^2}\right)^T \qquad \theta_2 = \left(\frac{a}{{\sigma_2}^2}, -\frac{1}{2{\sigma_2}^2}\right)^T \tag{6.60}$$

As combination function, we choose the addition: $k(\Xi_1, \Xi_2) = \Xi_1 + \Xi_2$. We allow both single labels and the label set $\{1, 2\}$, i.e. the set of possible label sets is $\mathbb{L} = \{\{1\}, \{2\}, \{1, 2\}\}$. The expected values of the observation X conditioned on the label set are thus as follows:

$$\mathbb{E}_{X\sim P_1}[X] = -a \qquad \mathbb{E}_{X\sim P_2}[X] = a \qquad \mathbb{E}_{X\sim P_{1,2}}[X] = 0 . \tag{6.61}$$

Since the convolution of two Gaussian distributions is again a Gaussian distribution, data with the multi-label set $\{1,2\}$ is also distributed according to a Gaussian. We denote the parameters of this proxy-distribution by θ_{12}, with

$$\theta_{12} = \left(0, -\frac{1}{2({\sigma_1}^2 + {\sigma_2}^2)}\right)^T .$$

In the following, we analyze the estimation accuracy of different approaches and then compare the obtained results with results from experiments. We focus on the estimation of the mean value. However, due to the parametrization in the exponential families, we also need to compute the estimator for the standard deviation. As a quality measure for the estimator $\hat{\boldsymbol{\theta}}$, we use the mean square error (MSE) as defined in Section 2.4.1.

Lemma 4. *Assume a generative setting as described above. Denote the total number of data items by N and the fraction of data items with label set $\mathcal{L}$ by $\pi_{\mathcal{L}}$. Furthermore, we define*

$$w_{12} := \pi_2\sigma_1^2 + \pi_1\sigma_2^2 \qquad s_{12} := \sigma_1^2 + \sigma_2^2 \ .$$

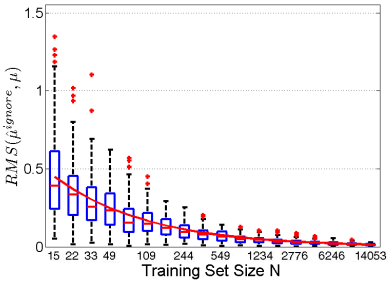
The mean square error in the estimator of the mean, averaged over all sources, for the inference methods $\mathcal{M}_{ignore}$, $\mathcal{M}_{new}$, $\mathcal{M}_{cross}$ and $\mathcal{M}_{deconv}$ is as follows:

$$MSE(\hat{\boldsymbol{\mu}}^{ignore}, \boldsymbol{\mu}) = \frac{1}{2}\left(\frac{{\sigma_1}^2}{\pi_1 N} + \frac{{\sigma_2}^2}{\pi_2 N}\right) \tag{6.62}$$

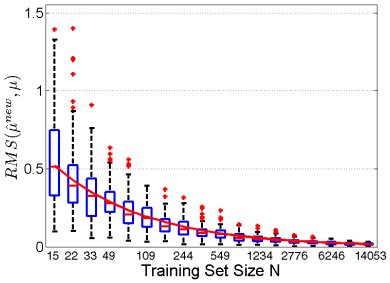
$$MSE(\hat{\boldsymbol{\mu}}^{new}, \boldsymbol{\mu}) = \frac{1}{3}\left(\frac{{\sigma_1}^2}{\pi_1 N} + \frac{{\sigma_2}^2}{\pi_2 N} + \frac{{\sigma_1}^2 + {\sigma_2}^2}{\pi_{12} N}\right) \tag{6.63}$$

$$\begin{aligned} MSE(\hat{\boldsymbol{\mu}}^{cross}, \boldsymbol{\mu}) =& \frac{1}{2}\pi_{12}^2\left(\frac{1}{(\pi_1+\pi_{12})^2} + \frac{1}{(\pi_2+\pi_{12})^2}\right)a^2 \\ &+ \frac{1}{2}\pi_{12}\left(\frac{\pi_1}{(\pi_1+\pi_{12})^3 N} + \frac{\pi_2}{(\pi_2+\pi_{12})^3 N}\right)a^2 \\ &+ \frac{1}{2}\left(\frac{\pi_1\sigma_1^2 + \pi_{12}\sigma_{12}^2}{(\pi_1+\pi_{12})^2 N} + \frac{\pi_2\sigma_2^2 + \pi_{12}\sigma_{12}^2}{(\pi_2+\pi_{12})^2 N}\right) \end{aligned} \tag{6.64}$$

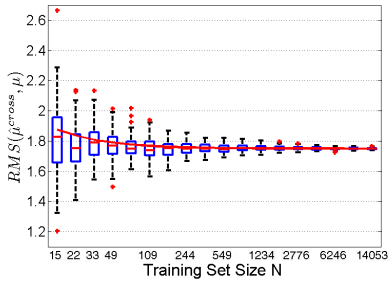
$$\begin{aligned} MSE(\hat{\boldsymbol{\mu}}^{deconv}, \boldsymbol{\mu}) =& \frac{1}{2}\left(\frac{\pi_{12}^2\sigma_2^2 w_{12} + \pi_{12}\pi_2 m_1 + \pi_1\pi_2^2 s_{12}^2}{\left(\pi_1\pi_2 s_{12} + \pi_{12} w_{12}\right)^2 N}\sigma_1^2\right. \\ &\left. + \frac{\pi_{12}^2\sigma_1^2 w_{12} + \pi_{12}\pi_1 m_2 + \pi_1^2\pi_2 s_{12}^2}{\left(\pi_1\pi_2 s_{12} + \pi_{12} w_{12}\right)^2 N}\sigma_2^2\right) \end{aligned} \tag{6.65}$$



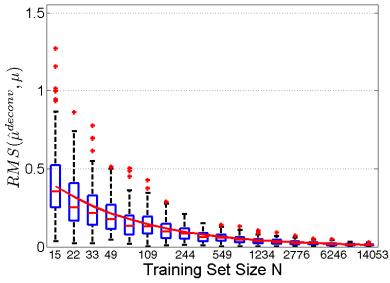
(a) Estimator Accuracy for $\mathcal{M}_{ignore}$



(b) Estimator Accuracy for $\mathcal{M}_{new}$



(c) Estimator Accuracy for $\mathcal{M}_{cross}$



(d) Estimator Accuracy for $\mathcal{M}_{deconv}$

Figure 6.1: Deviation of parameter values from true values: The box plot indicate the values obtained in an experiment with 100 runs, the red line gives the root mean square error (RMS) predicted by the asymptotic analysis. Note the difference in scale in Figure 6.1(c).

with $m_1 := (\pi_2\sigma_1^2\sigma_{12}^2 + 2\pi_1\sigma_2^2 s_{12})$ *and* $m_2 := (\pi_1\sigma_2^2\sigma_{12}^2 + 2\pi_2\sigma_1^2 s_{12})$.

The proof mainly consists of lengthy calculations and is given in Section A.2. We rely on the computer-algebra system MAPLE for parts of the calculations.

To verify the theoretical result, we apply the presented inference techniques to synthetic data, generated with $a = 3.5$ and unit variance: $\sigma_1 = \sigma_2 = 1$. The Bayes error, i.e. the error of the optimal generative classifier, in this setting is 9.59%. We use training data sets of different size and test sets of the same size as the maximal size of the training data sets. All ex-

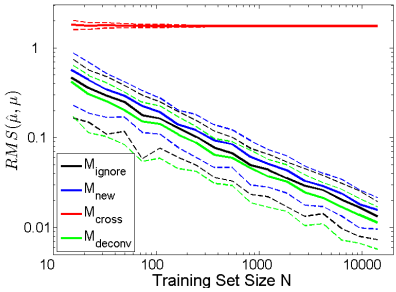


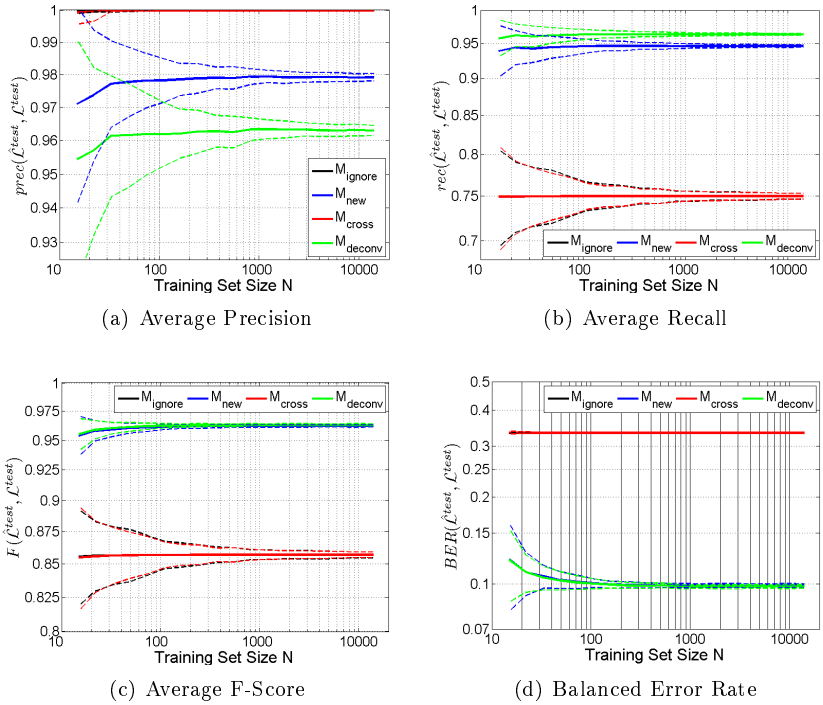
Figure 6.2: Accuracy of $\hat{\mu}$ for different training set sizes and different inference techniques. $\mathcal{M}_{cross}$ is not consistent and thus has a non-vanishing RMS error of $a/2$ in this experimental setting. The three other inference techniques yield consistent estimators, but at different convergence rates: The estimators obtained by $\mathcal{M}_{deconv}$ have the fastest convergence, $\mathcal{M}_{new}$ attains the slowest convergence of the three consistent techniques.

periments are repeated with 100 randomly sampled training and test data sets.

In Figure 6.1, the average deviation of the estimated source centroids from the true centroids are plotted for different inference techniques and a varying number of training data and compared with the values predicted from the asymptotic analysis. $\mathcal{M}_{cross}$ has a clear bias, i.e. a deviation from the true parameter values which does not vanish as the number of data items grows to infinity. All other inference technique are consistent, but differ in the convergence rate.

Furthermore, we observe that the predictions from theory agree with the deviations measured in the experiments. Small differences are obtained for small training set sizes, as in this case, the assumptions underlying the asymptotic analysis are only partially fulfilled. As the number of data items increases, these deviations vanish.

Figure 6.2 shows the asymptotic behavior of the estimation accuracy for different inference techniques. $\mathcal{M}_{cross}$ yields biased estimators, while the three other methods yield consistent estimators. $\mathcal{M}_{deconv}$ attains the fastest convergence, followed by $\mathcal{M}_{ignore}$. $\mathcal{M}_{new}$ has the slowest convergence of the analyzed consistent inference techniques, as this method infers parameters of a separate class for the multi-label data. Due to the generative process,



(a) Average Precision

(b) Average Recall

(c) Average F-Score

(d) Balanced Error Rate

Figure 6.3: Classification quality of different inference methods. Data is generated from two sources with mean ± 3.5 and standard deviation 1. The experiment is run with 100 pairs of training and test data.

these data items have a higher variance, which entails a high variance of the respective estimator. Therefore, $\mathcal{M}_{new}$ has a higher average estimation error than $\mathcal{M}_{new}$.

Finally, the quality of the classification results obtained by different methods is given in Figure 6.3. The low precision value of $\mathcal{M}_{deconv}$ shows that this classification rule is more likely to assign a wrong label to a data item than the competing inference methods. Paying this price, on the other hand, $\mathcal{M}_{deconv}$ yields the highest recall values of all classification techniques analyzed in this work. On the other extreme, $\mathcal{M}_{cross}$ and $\mathcal{M}_{ignore}$ have a precision of 100%, but a very low recall of about 75%. Note that $\mathcal{M}_{ignore}$

only handles single-label data and is thus limited to attributing single labels. In the setting of this experiments, the single label data items are very clearly separated. A confusion is thus very unlikely, which explains the very precise labels as well as the low recall rate. In terms of the F-score, which is the harmonic mean of the precision and the recall, $\mathcal{M}_{deconv}$ yields the best results for all training set sizes, closely followed by $\mathcal{M}_{new}$. $\mathcal{M}_{ignore}$ and $\mathcal{M}_{cross}$ lie clearly behind.

Also for the balanced error rate BER, the deconvolutive model yields the best results, with $\mathcal{M}_{new}$ reaching similar results. Both $\mathcal{M}_{cross}$ and $\mathcal{M}_{ignore}$ incur significantly higher errors. In $\mathcal{M}_{cross}$, this effect is caused by the biased parameter estimators, while $\mathcal{M}_{ignore}$ has discarded all training data with label set $\{1, 2\}$ and can thus "not do anything with such data".

6.5 Disjunction of Bernoulli-Distributed Emissions

We consider the Bernoulli distribution as an example of a discrete distribution in the exponential family with emissions in $\mathbb{B} := \{0, 1\}$. The Bernoulli distribution has one parameter β, which describes the probability for a 1 (usually used to represent "success"). To represent the Bernoulli distribution as a member of the exponential family, we use the following parametrization:

$$\begin{aligned} \theta_k &= \log\left(\frac{\beta_k}{1-\beta_k}\right) \\ \phi(\Xi_k) &= \Xi_k \\ A(\theta_k) &= -\log\left(1-\beta_k\right) = -\log\left(1 - \frac{\exp\theta_k}{1+\exp\theta_k}\right) \end{aligned}$$

Given the parameter θ_k, the value of β_k is given by $\beta_k = \frac{\exp\theta_k}{1+\exp\theta_k}$. We use both parameterizations in parallel, with the semantics that β is the probability for a 1, and θ is the parameter of the distribution as a member of the exponential family. For simpler notation, we define

$$e_k := \mathbb{E}_{\Xi\sim Bern(\beta_k)}[\Xi] = \frac{\exp\theta_k}{1+\exp\theta_k} = \beta_k \tag{6.66}$$

$$v_k := \mathbb{V}_{\Xi\sim Bern(\beta_k)}[\Xi] = \frac{\exp\theta_k}{(1+\exp\theta_k)^2} = \beta_k(1-\beta_k) \tag{6.67}$$

As combination function, we consider the Boolean OR, which yields a 1 if either of the two inputs is 1, and 0 otherwise. Thus, we have

$$P(X = 1|\mathcal{L} = \{1\}) = \beta_1 \tag{6.68}$$
$$P(X = 1|\mathcal{L} = \{2\}) = \beta_2 \tag{6.69}$$

For the distribution of the multi-labeled observations, we have

$$P(X = 1|\mathcal{L} = \{1, 2\}) = \beta_1 + \beta_2 - \beta_1\beta_2 := \beta_{12} \tag{6.70}$$

Note that $\beta_{12} \geq \beta_1$ and $\beta_{12} \geq \beta_2$: When combining the emissions of two Bernoulli distributions with a Boolean OR, the probability of a one is at least as large as the probability that one of the sources emitted a one. Equality implies either that the partner source never emits a one, i.e. $\beta_{12} = \beta_1$ if and only if $\beta_2 = 0$, or that one of the sources always emits a one, i.e. $\beta_{12} = \beta_1$ if $\beta_1 = 1$.

The conditional probability distributions are as follows:

$$P(\boldsymbol{\Xi}|(X, \{1\}), \boldsymbol{\theta}) = 1_{\{\Xi^{(1)}=X\}} \cdot Bern(\Xi^{(2)}|\theta^{(2)}) \tag{6.71}$$
$$P(\boldsymbol{\Xi}|(X, \{2\}), \boldsymbol{\theta}) = Bern(\Xi^{(1)}|\theta^{(1)}) \cdot 1_{\{\Xi^{(2)}=X\}} \tag{6.72}$$
$$P(\boldsymbol{\Xi}|(0, \{1, 2\}), \boldsymbol{\theta}) = 1_{\{\Xi^{(1)}=0\}} \cdot 1_{\{\Xi^{(2)}=0\}} \tag{6.73}$$
$$P(\boldsymbol{\Xi}|(1, \{1, 2\}), \boldsymbol{\theta}) = \frac{P(\boldsymbol{\Xi}, X = 1|\mathcal{L} = \{1, 2\}, \boldsymbol{\theta})}{P(X = 1|\mathcal{L} = \{1, 2\}, \boldsymbol{\theta})} \tag{6.74}$$

In particular, the joint distribution of the emission vector $\boldsymbol{\Xi}$ and the observation X is as follows:

$$P(\boldsymbol{\Xi} = (0, 0), X = 0|\mathcal{L} = \{1, 2\}, \boldsymbol{\theta}) = (1 - \beta_1)(1 - \beta_2)$$
$$P(\boldsymbol{\Xi} = (0, 1), X = 1|\mathcal{L} = \{1, 2\}, \boldsymbol{\theta}) = (1 - \beta_1)\beta_2$$
$$P(\boldsymbol{\Xi} = (1, 0), X = 1|\mathcal{L} = \{1, 2\}, \boldsymbol{\theta}) = \beta_1(1 - \beta_2)$$
$$P(\boldsymbol{\Xi} = (1, 1), X = 1|\mathcal{L} = \{1, 2\}, \boldsymbol{\theta}) = \beta_1\beta_2$$

All other combinations of $\boldsymbol{\Xi}$ and X have probability 0.

Lemma 5. *Consider the generative setting described above, with N data items in total. The fraction of data items with label set $\mathcal{L}$ by $\pi_{\mathcal{L}}$. Furthermore, define $v_1 := \beta_1(1 - \beta_1)$, $v_2 := \beta_2(1 - \beta_2)$, $v_{12} := \beta_{12}(1 - \beta_{12})$, $w_1 := \beta_1(1 - \beta_2)$, $w_2 := \beta_2(1 - \beta_1)$ and*

$$\hat{v}_1 = \frac{\pi_{12}}{(\pi_1 + \pi_{12})^2} w_2 \left(1 - \pi_{12} w_2\right) \quad \hat{v}_2 = \frac{\pi_{12}}{(\pi_2 + \pi_{12})^2} w_1 \left(1 - \pi_{12} w_1\right) \ . \tag{6.75}$$

The mean square error in the estimator of the Bernoulli parameter $\hat{\beta}$, averaged over all sources, for the inference methods $\mathcal{M}_{ignore}$, $\mathcal{M}_{new}$, $\mathcal{M}_{cross}$ and $\mathcal{M}_{deconv}$ is as follows:

$$MSE(\hat{\boldsymbol{\beta}}^{new}, \boldsymbol{\beta}) = \frac{1}{3}\left(\frac{\beta_1(1-\beta_1)}{\pi_1 N} + \frac{\beta_2(1-\beta_2)}{\pi_2 N} + \frac{\beta_{12}(1-\beta_{12})}{\pi_{12} N}\right) \tag{6.76}$$

$$MSE(\hat{\boldsymbol{\beta}}^{ignore}, \boldsymbol{\beta}) = \frac{1}{2}\left(\frac{\beta_1(1-\beta_1)}{\pi_1 N} + \frac{\beta_2(1-\beta_2)}{\pi_2 N}\right) \tag{6.77}$$

$$\begin{aligned} MSE(\hat{\boldsymbol{\beta}}^{cross}, \boldsymbol{\beta}) =& \frac{1}{2}\left(\frac{\pi_{12}}{\pi_1+\pi_{12}} w_2\right)^{\otimes} + \frac{1}{2}\left(\frac{\pi_{12}}{\pi_2+\pi_{12}} w_1\right)^{\otimes} \\ &+ \frac{1}{2}\frac{1}{\pi_1 N}\frac{v_1^2}{\hat{v}_1^2}\left(\frac{\pi_{12}^2\left(\beta_1-\beta_{12}\right)^2}{\left(\pi_1+\pi_{12}\right)^3} + \frac{\pi_1 v_1 + \pi_{12} v_{12}}{\left(\pi_1+\pi_{12}\right)^2}\right) \\ &+ \frac{1}{2}\frac{1}{\pi_2 N}\frac{v_2^2}{\hat{v}_2^2}\left(\frac{\pi_{12}^2\left(\beta_2-\beta_{12}\right)^2}{\left(\pi_2+\pi_{12}\right)^3} + \frac{\pi_2 v_2 + \pi_{12} v_{12}}{\left(\pi_2+\pi_{12}\right)^2}\right) \end{aligned} \tag{6.78}$$

$$\begin{aligned} MSE(\hat{\boldsymbol{\beta}}^{deconv}, \boldsymbol{\beta}) =& \frac{1}{2}\frac{1}{\pi_1 N}\frac{\pi_2\beta_{12} + \pi_{12} w_2}{\pi_{12}(\pi_1 w_2 + \pi_2 w_1) + \pi_1\pi_2\beta_{12}} v_1 \\ &+ \frac{1}{2}\frac{1}{\pi_2 N}\frac{\pi_1\beta_{12} + \pi_{12} w_1}{\pi_{12}(\pi_1 w_2 + \pi_2 w_1) + \pi_1\pi_2\beta_{12}} v_2 \end{aligned} \tag{6.79}$$

The proof of this lemma involves lengthy calculations that we partially perform in MAPLE. Details are given in Section A.3 in the appendix.

To evaluate the estimators obtained by the different inference methods, we use a setting with $\boldsymbol{\beta}_1 = 0.40 \cdot \mathbf{1}_{10\times 1}$ and $\boldsymbol{\beta}_2 = 0.20 \cdot \mathbf{1}_{10\times 1}$, where $\mathbf{1}_{10\times 1}$ denotes a 10-dimensional vector of ones. Each dimension is treated independently, and all results reported here are averages and standard deviations over 100 independent training and test samples.

The root mean square error RMS of the estimators obtained by different inference techniques are depicted in Figure 6.5. We observe that values predicted by theory are in good agreement with the deviations measured in the experiments. Comparing the accuracy of estimators from different inference techniques, we observe that $\mathcal{M}_{cross}$ yields clearly biased estimators. For the three other methods, $\mathcal{M}_{new}$ has the largest deviation, followed by $\mathcal{M}_{ignore}$. $\mathcal{M}_{deconv}$ yields the most accurate parameters. The empirical RMS of all considered methods is depicted in Figure 6.4. Note the high bias of $\mathcal{M}_{cross}$,

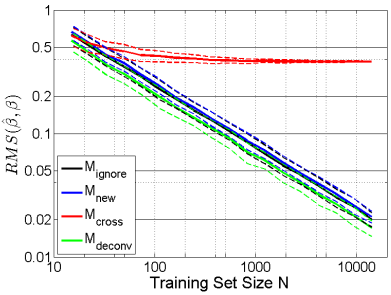
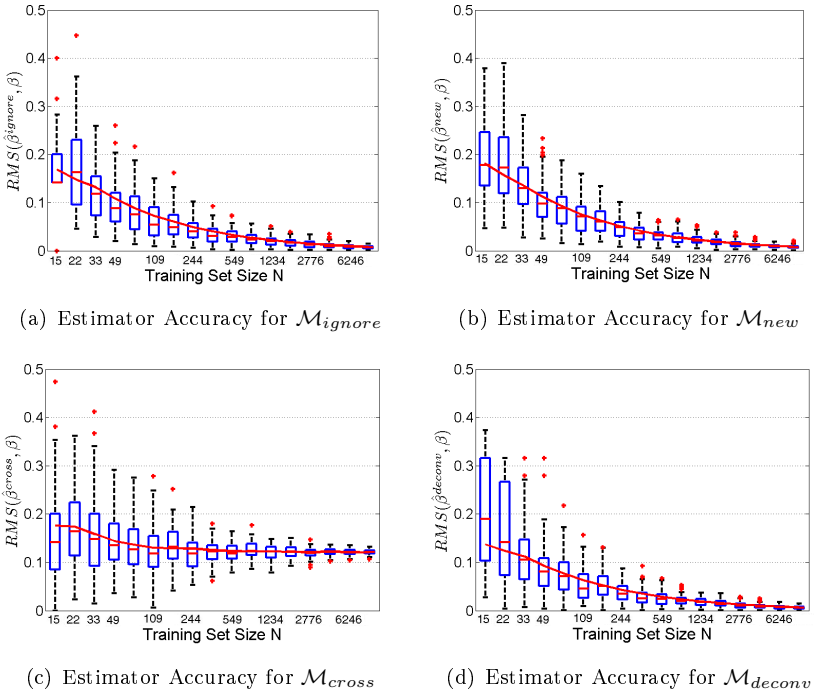


Figure 6.4: Root mean square error of estimators obtained by different estimation techniques as a function of the training set size.

while the remaining three techniques yield continuously more accurate parameters as the training set increases.

Recall that the parameter describing the proxy distribution of data items from the label set $\{1, 2\}$ is defined as $\beta_{12} = \beta_1 + \beta_2 - \beta_1\beta_2$ (Eq. 6.70) and thus larger than any of β_1 or β_2. While the expectation of the Bernoulli distribution (Eq. 6.66) is thus increasing, the variance (given in Eq. 6.67) $\beta_{12}(1-\beta_{12})$ of the proxy distribution is smaller than the variance of the base distributions. To study the influence of this effect onto the estimator precision, we compare the RMS of the source estimators obtained by $\mathcal{M}_{deconv}$ and $\mathcal{M}_{new}$, illustrated in Figure 6.6: The inference method $\mathcal{M}_{deconv}$ is most advantageous if at least one of β_1 or β_2 is small. In this case, the variance of the proxy distribution is approximately the sum of the variances of the base distributions. As the parameters of the base distribution increase, the advantage of $\mathcal{M}_{deconv}$ in comparison to $\mathcal{M}_{new}$ decreases. If both β_1 and β_2 are high, the variance of the proxy distribution is smaller than the variance of any of the base distributions, and $\mathcal{M}_{new}$ yields more accurate parameter estimators than $\mathcal{M}_{deconv}$.



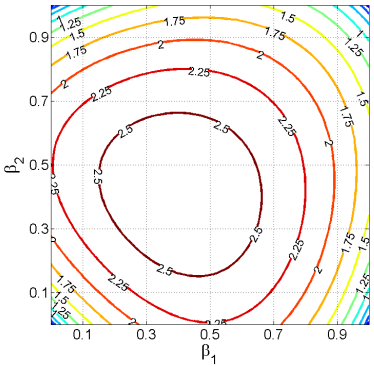
(a) Estimator Accuracy for $\mathcal{M}_{ignore}$

(b) Estimator Accuracy for $\mathcal{M}_{new}$

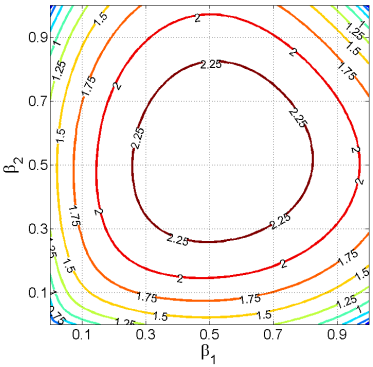
(c) Estimator Accuracy for $\mathcal{M}_{cross}$

(d) Estimator Accuracy for $\mathcal{M}_{deconv}$

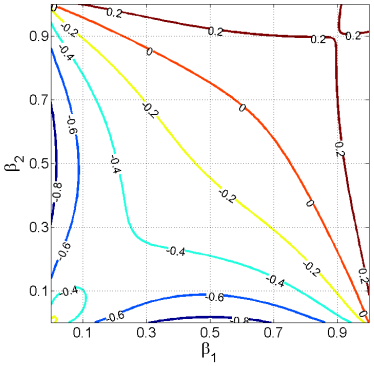
Figure 6.5: Deviation of parameter values from true values: The box plot indicate the values obtained in an experiment with 100 runs, the red line gives the RMS predicted by the asymptotic analysis.



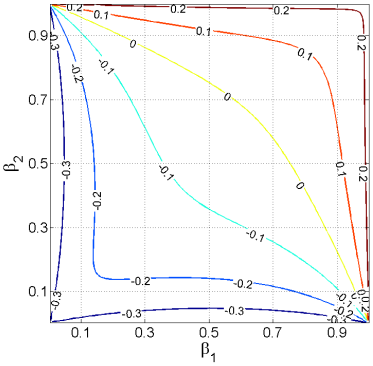
(a) $RMS(\hat{\boldsymbol{\beta}}^{new}, \boldsymbol{\beta})$



(b) $RMS(\hat{\boldsymbol{\beta}}^{deconv}, \boldsymbol{\beta})$



(c) $RMS(\hat{\boldsymbol{\beta}}^{deconv}, \boldsymbol{\beta}) - RMS(\hat{\boldsymbol{\beta}}^{new}, \boldsymbol{\beta})$



(d) $\frac{RMS(\hat{\boldsymbol{\beta}}^{deconv}, \boldsymbol{\beta}) - RMS(\hat{\boldsymbol{\beta}}^{new}, \boldsymbol{\beta})}{RMS(\hat{\boldsymbol{\beta}}^{new}, \boldsymbol{\beta})}$

Figure 6.6: Comparison of the estimation accuracy for $\boldsymbol{\beta}$ for the two methods $\mathcal{M}_{new}$ and $\mathcal{M}_{deconv}$ for different values of β_1 and β_2.

Chapter 7

Ignoring Co-Occurrence Implies Model Mismatch

In the previous chapter, we have observed that $\mathcal{M}_{cross}$ yields biased parameter estimators. In this chapter, we show that if the true combination function is a bijection in the emission of a single source, training an independent generative classifier by maximum likelihood for every class implies a model mismatch, which in turn causes biased estimators.

7.1 Preliminaries

Before starting with the main theorem, we define the type of co-occurrence ignoring inference procedures and give a few auxiliary lemmata. We assume that inference is based on a data set $\mathbf{D} = (D_1, \ldots, D_N)$ of N observations X_n with corresponding label set $\mathcal{L}_n$. The tuple $D_n = (X_n, \mathcal{L}_n)$, for $n = 1, \ldots, N$, is called a data item.

7.1.1 Co-Occurrence-Ignoring Inference Procedures

We first define sources co-occurrence:

Definition 3. ***(Co-Occurring Sources)*** *Two sources k_1, k_2, $k_1 \neq k_2$, are called* co-occurring *in the data set $\mathbf{D}$ if there exists at least one label set in*

D *containing both* k_1 *and* k_2*. Formally, we have*

$$k_1, k_2 \text{ co-occurring in } \mathbf{D} :\Longleftrightarrow \exists n \in \{1, \ldots, N\} : k_1 \in \mathcal{L}_n \wedge k_2 \in \mathcal{L}_n$$

Partial independence and partial conditional independence are defined as follows:

Definition 4. ***(Partially (Conditionally) Independent)*** *Two vectors of random variables* $X = (X_1, \ldots, X_{Q_X})$ *and* $Y = (Y_1, \ldots, Y_{Q_Y})$ *are called* partially independent*, denoted by* $X \perp\!\!\!\perp_\partial Y$*, if there exist at least one pair of vector components* X_{q_1}*,* Y_{q_2}*,* $1 \leq q_1 \leq Q_X$ *and* $1 \leq q_2 \leq Q_Y$*, which are independent, i.e.* $P(X_{q_1}, Y_{q_2}) = P(Y_{q_2}) \cdot P(X_{q_1})$*.*
If there exists at least one pair of vector components X_{q_1}*,* Y_{q_2}*, for some* $1 \leq q_1 \leq Q_X$ *and* $1 \leq q_2 \leq Q_Y$*, which are independent conditioned on* **D** *(i.e.* $P(X_{q_1}, Y_{q_2}|\mathbf{D}) = P(Y_{q_2}|\mathbf{D}) \cdot P(X_{q_1}|\mathbf{D})$*), then* X*,* Y *are called* partially conditionally independent*, denoted by* $X \perp\!\!\!\perp_\partial Y|\mathbf{D}$*.*

In the remainder of this paper, we focus on the training or inference phase of classifiers. Using a particular inference scheme $\mathcal{M}$, parameter estimates are computed based on a data set with corresponding labels.

Definition 5. ***(Co-Occurrence Ignoring Inference Procedures)*** *An inference procedure* $\mathcal{M}$ *based on maximum-likelihood is called* co-occurrence ignoring *on a training data set* **D** *if it fulfills the three following conditions:*

1. $\mathcal{M}$ *handles multi-labeled data.*

2. *The likelihood of the parameters* θ_k *of source* k *depends only on data items which contain* k *in their label sets.*

3. *The parameters* θ_{k_1}*,* θ_{k_2} *of two co-occurring sources* k_1*,* k_2 *are assumed to be partially independent given the training data* **D***:*

$$\theta_{k_1} \perp\!\!\!\perp_\partial \theta_{k_2}|\mathbf{D} \qquad \forall\ k_1, k_2 \in \mathcal{K}.$$

The second condition generalizes the assumption made in single-label classification that parameter estimators for source k depend only on data with this label (see Section 6.1.4).

7.1.2 Model Mismatch

Since the training set has only finite size, inference procedures typically suffer from an estimation error, which typically decreases as more samples are available. If the estimated distribution deviates from the true distribution even in the asymptotic case of infinite training data, the inference methods is said to yield inconsistent parameter estimators. A possible cause for such estimators is a mismatch between the model assumed by the inference procedure and the true model that generated the data.

As stated before, maximum likelihood estimators are unbiased. We show in the following that co-occurrence ignoring inference schemes imply a model mismatch and thus cause biased parameter estimators. Note that for identifiable parameters (defined in Def. 1), a difference between the estimated and the true parameters implies a difference between the estimated and the true probability distribution.

In later sections, we rely on representations of the density and the combination function as infinite Taylor series. Functions which can be represented as (infinite) Taylor series are called analytic:

Definition 6. *(**Analytic Function**) An* analytic function *is an infinitely differentiable function f on Ω such that the Taylor series at any point $x_0 \in \Omega$, $T(f, x_0, x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n$, converges to $f(x)$ for x in a neighborhood of x_0.*

Polynomials, trigonometric functions, the logarithm and the exponential function are analytic. Sums, products and compositions of analytic functions are again analytic. Furthermore, the reciprocal of an analytic function that is nowhere zero is analytic.

7.1.3 Auxiliary Lemmata

In this section, we present four auxiliary lemmata, which we refer to in the proof of the main theorem. All proofs are given in the appendix (Section A.4).

First, we need to define some notation: Denote by k_1 and k_2, $k_1 \neq k_2$, two sources, parameterized by θ_1 and θ_2, which are assumed to be partially conditionally independent. We denote by $(\theta_{1,c_1}, \theta_{2,c_2})$ a pair of components of θ_1 and θ_2 which are assumed to be conditionally independent given the training data $\mathbf{D}$. Denote by $\mathrm{c}_{\kappa,(k_1,k_2)}(\xi_1, \xi_2) := \mathrm{c}_\kappa(\boldsymbol{\xi}, \{k_1, k_2\})$ the combination function for the label set $\{k_1, k_2\}$. Let $\mathrm{d}_\kappa(\xi_1, x_n) := \mathrm{c}^{-1}_{\kappa,(k_1,k_2)}(\xi_1, \cdot)$

be the inverse of the combination function $\mathrm{c}_{\kappa,(k_1,k_2)}(\xi_1,\xi_2)$ with respect to the second argument, and set $\mathrm{d}_n(\xi) := \mathrm{d}_\kappa(\xi_1, x_n)$. We assume that $\mathrm{c}_{\kappa,(k_1,k_2)}(\xi_1,\xi_2)$ is a bijection in ξ_2. All computations are analogous if $\mathrm{c}_{\kappa,(k_1,k_2)}(\xi_1,\xi_2)$ is a bijection in ξ_1.

Derivatives of probability densities with respect to parameters are denoted with an upper dot on the density (we assume the parameter with respect to which the derivative is taken is clear from the context). Derivatives with respect to the random variable are denoted by the degree of the derivation in upper brackets:

$$\dot{p}_k(\xi) := \left.\frac{\partial p_k(\xi)}{\partial \theta_{k,c_k}}\right|_{\theta_{k,c_k}=\hat{\theta}^{ML}_{k,c_k}} \qquad p_k^{(m)}(\xi_k) := \frac{\partial^m p_k(\xi)}{\partial \xi_k^m} \qquad \text{for } k = 1, 2 \ .$$

Lemma 6. *Assume independent probability density functions $p_k(\xi_k)$ parameterized by θ_k, for $k = 1, 2$. Then, the derivative of the joint distribution $p_{12}(\cdot)$ with respect to both parameters evaluated at the value of the maximum likelihood estimator $\hat{\theta}_k^{ML}$ of the parameter is zero. Formally:*

$$\xi_1 \perp\!\!\!\perp \xi_2 \implies \left.\frac{\partial^2 p_{12}(\xi_1,\xi_2|\theta_1,\theta_2)}{\partial\theta_1\partial\theta_2}\right|_{\theta_1=\hat{\theta}_1^{ML},\theta_2=\hat{\theta}_2^{ML}} = 0 \ .$$

The following lemma allows us to rewrite the independence of two parameters given the data as an equality of two sums:

Lemma 7. *Given a training data set $\mathbf{D}$ generated according to the generative process described in Chapter 2 with a combination function c_κ being a bijection in the emission of at least one source in the label set. If θ_1 and θ_2, the parameter of sources k_1 and k_2, are learned by maximum likelihood, partial conditional independence of θ_1 and θ_2 given $\mathbf{D}$ implies*

$$\begin{aligned}\sum_n \frac{\int \dot{p}_1(\xi)\dot{p}_2(d_n(\xi))\,\mathrm{d}\xi \cdot \int p_1(\xi)p_2(c_n(\xi))\,\mathrm{d}\xi}{p(x_n)^2} \\ = \sum_n \frac{\int p_1(\xi)\dot{p}_2(d_n(\xi))\,\mathrm{d}\xi \cdot \int \dot{p}_1(\xi)p_2(d_n(\xi))\,\mathrm{d}\xi}{p(x_n)^2} \ ,\end{aligned} \tag{7.1}$$

where $p(x_n) := \int p_1(\xi) \cdot p_2(d_n(\xi))\,\mathrm{d}\xi$. Note that only indices n with $\mathcal{L}_n = \{k_1, k_2\}$ might have a non-zero contribution to the sum. For n with $\mathcal{L}_n \neq \{k_1, k_2\}$, the contributions on either side are 0.

Informally speaking, the independence assumption between two parameter components implies that the partial derivatives of the data likelihood with respect to the respective parameter components can be distributed without changing the value of the expression: On the left hand side, the partial derivative of p_1 and p_2 stand under the same integral, while they are under different integrals on the right hand side.

The following lemma allows us to write the equality condition implied by the independence assumption as an equality of two Taylor series:

Lemma 8. *Assume $c(\cdot)$ is an analytic function and the density functions $p_k(\xi_k|\theta_k)$ are continuously differentiable with respect to their parameters θ_k and analytic functions with respect to the random variables ξ_k, for $k = 1, 2$. Then, Equation 7.1 can be rewritten as an infinite Taylor series*

$$\sum_{i=0}^{\infty} C_{lhs}^{i} \cdot \xi^{i} = \sum_{i=0}^{\infty} C_{rhs}^{i} \cdot \xi^{i} \tag{7.2}$$

with coefficients C_{lhs}^{i} and C_{rhs}^{i} given by

$$C_{\alpha}^{i} = \sum_{n} \frac{1}{p(x_n)^2} \sum_{j=0}^{i} \frac{C_{\alpha,1}^{j}(x_n)}{j!} \cdot \frac{C_{\alpha,2}^{i-j}(x_n)}{(i-j)!} \ , \tag{7.3}$$

where $\alpha = lhs, rhs$ and

$$C_{lhs,1}^{j}(x_n) = \sum_{m=0}^{j-1} \binom{j-1}{m} \dot{p}_1^{(j-1-m)}(0) \cdot S_m(\dot{p}_2, n) \tag{7.4}$$

$$C_{lhs,2}^{j}(x_n) = \sum_{m=0}^{j-1} \binom{j-1}{m} p_1^{(j-1-m)}(0) \cdot S_m(p_2, n) \tag{7.5}$$

$$C_{rhs,1}^{j}(x_n) = \sum_{m=0}^{j-1} \binom{j-1}{m} p_1^{(j-1-m)}(0) \cdot S_m(\dot{p}_2, n) \tag{7.6}$$

$$C_{rhs,2}^{j}(x_n) = \sum_{m=0}^{j-1} \binom{j-1}{m} \dot{p}_1^{(j-1-m)}(0) \cdot S_m(p_2, n) \tag{7.7}$$

for $l \geq 1$. $C^0_{\alpha,i}(x_n)$ *denote integration constants. Furthermore,*

$$S_m(f,n) = \sum_{t \in T_m} b_m(t,f,n) \tag{7.8}$$

$$b_m(t,f,n) = \frac{m! \cdot f^{(\sum_{l=1}^m t_l)}(d_n(0))}{\prod_{l=1}^m t_l!} \cdot \prod_{l=1}^m \left(\frac{d_n^{(l)}(0)}{l!} \right)^{t_l} \tag{7.9}$$

$$T_m = \left\{ (t_1, \ldots, t_m) \in \mathbb{N}_0^m | \sum_{l=1}^m i \cdot t_l = m \right\} . \tag{7.10}$$

The next lemma shows that the equality condition for the two Taylor series (Eq. 7.2) implies that all derivatives of $\mathrm{d}_n(\cdot)$ evaluated at 0 must be equal to 0.

Lemma 9. *Given coefficients* C^i_{lhs}, C^i_{rhs} *as defined in Equation 7.3, Equation 7.2 implies that all derivatives of* $d_n(\xi)$ *with respect to* ξ *evaluated at* $\xi = 0$ *must be zero, i.e.* $d_n^{(m)}(0) = 0 \, \forall m \in \mathbb{N}$.

This lemma implies that $\mathrm{d}_n(\cdot)$ is a constant in the neighborhood of 0.

7.2 Provable Model Mismatch due to Ignored Co-Occurrence

We are now ready to formate the main theorem on model mismatch. We first give the theorem for binary labels and then sketch the extension to label sets of higher degree.

Main Theorem for Binary Labels

Theorem 6. *Given is a training data set* $\mathbf{D}$ *with single-label and binary-label data generated according to the generative process described in Chapter 2. All source distributions are assumed to be continuously differentiable w.r.t. the parameters, and analytic functions w.r.t. the random variables. The binary combination function is a bijection in the emission of at least one source in the label set, and its inverse with respect to any single argument is an analytic function. Then, any co-occurrence ignoring inference scheme* $\mathcal{M}$ *trained by maximum likelihood on* $\mathbf{D}$ *suffers from model mismatch.*

The proof is done by contradiction. We assume that $\mathcal{M}$ finds asymptotically the true parameters and then show that this assumption yields a contradiction.

Proof. **Theorem 6.** Assume the model adopted by $\mathcal{M}$ to explain the data set $\mathbf{D}$ matches the true model of the generative process. As the inference scheme $\mathcal{M}$ is co-occurrence ignoring, there is at least one pair of parameters θ_1, θ_2 which do not parameterize the same source and which $\mathcal{M}$ assumes to be conditionally independent given the training data $\mathbf{D}$. We denote the source distributions parameterized by θ_k by p_k $(k = 1, 2)$. By Lemma 7, conditional independence of parameters θ_1, θ_2 implies that the optimality condition of maximum likelihood (Eq. 5.2) yields the condition given in Eq. 7.1.

Since $\mathrm{d}_n(\cdot)$ is an analytic function, we can use Lemma 8 to rewrite Equation 7.1 as an infinite Taylor series as given in Equation 7.2. By Lemma 9, this implies that all derivatives $\mathrm{d}_n^{(m)}(0)$ for $m \geq 1$ are zero. Since $\mathrm{d}_n(\cdot)$ is assumed to be analytic, it must be a constant in the neighborhood of 0. This is a contradiction to the assumption that $\mathrm{d}_n(\cdot)$ is a bijection. Therefore, the assumption that $\mathcal{M}$ adopts the true model has to be rejected. □

Extension to Labels of Higher Degree

In order to avoid too much clutter in the notation, we have given the proof only for training data containing single- and binary-labeled data. The following corollary generalizes Theorem 6 to combination functions of any arity.

Corollary 1. *Given a training data set* $\mathbf{D}$ *with single-label and multi-label data of any order generated according to the generative process described in Chapter 2. All source distributions are assumed to be continuously differentiable w.r.t. the parameters, and analytic functions w.r.t. the random variables. The combination function is a bijection in the emission of at least one source in the label set, and its inverse with respect to any single argument is an analytic function. Then, any co-occurrence ignoring inference scheme trained by maximum likelihood on* $\mathbf{D}$ *suffers from model mismatch.*

The proof is very similar to the proof given for the case of binary labels. We give a sketch of the proof in the appendix (Section A.4).

7.3 Implications for Multi-Label Classification

In the generative model described in Chapter 2 and depicted in Figure 2.1, the single label sources are independent, and the observations with multiple labels are combinations of the emissions of these sources. Since the binary combination function is assumed to be a bijection in one argument if the other argument is fixed to a fixed value, this introduces a one-to-one functional dependency between the observation and one of the source samples. It is therefore not surprising that co-occurrence ignoring classifiers incur a model mismatch. In the following, we discuss the implications of the theorem for the performance of different multi-label classifiers.

First of all, instance-based classification schemes such as the adaption of the k-nearest neighbor algorithm [121] or C4.5 algorithm with an entropy formula adapted for multi-label classification [25] do not estimate any distribution parameters. Theorem 6 is therefore not applicable to these classifiers and does not allow to draw any conclusions on their performance.

The definition of co-occurrence ignoring inference schemes (Section 7.1.1) requires that the inference scheme handles data with multiple labels. Out of the techniques presented in Section 6.3, $\mathcal{M}_{ignore}$ and $\mathcal{M}_{new}$ do not match this requirement.

The classification methods $\mathcal{M}_{cross}$ and $\mathcal{M}_{prob}$ do handle multi-label data and are co-occurrence ignoring as they are described in the mentioned publications. Both methods independently learn source parameters for each class and use also data with multiple labels for this. In doing so, they are disregarding the contributions of classes in the label set other than the currently trained one. This leads to a systematic deviation of the parameter estimators from the true parameter values.

In the pairwise ranking method [49], a different set of source parameters is learned for each pair of labels. This allows a different parametrization of the same source for different "partner" labels. This model assumption does not agree with the generative model in Eq. 2.4 and thus facilitates a model mismatch.

The mixture model for the word distribution was presented in [81] as well as $\mathcal{M}_{deconv}$ take into account co-occurring labels. Provided the combination function assumed in the models matches the true combination function, and that the true source distributions can be described with the parametric distributions assumed by the model, the presented theorem does therefore not imply a mismatch of these two methods with the true source.

With regard to the probability distribution, the theorem assumes density

functions which are continuously differentiable with respect to the parameters and analytic functions with respect to the random variables. Most continuous probability distributions fulfill these requirements. Exceptions are e.g. the Dirac delta function and the Cantor distribution.

The combination function is assumed to be a bijection in one of the arguments. Most elementary mathematical operations like (weighted) sum and difference, trigonometric functions, the logarithm, the product and the exponential function as well as combinations thereof are bijections. Softmax is also a bijection in any of the arguments. However, other combination functions like maximum and minimum are not bijections, and the theorem does not allow to draw any conclusion on inference procedures in this case.

Keeping this in mind, we recommend to use generative classifiers for multi-label classification whenever the generative process is sufficiently well known and the resulting optimization problem is stable and solvable within reasonable time. If a generative model can not be employed, one might either still use a classifier based on independent pairwise classifications, being aware that a model mismatch is inevitable and might lead to sub-optimal classification performance. Alternatively, the problem might be addressed by an instance-based classification technique, or using a generative inference procedure which do not rely on any source independence, such as $\mathcal{M}_{ignore}$ or $\mathcal{M}_{new}$. The latter option, however, is not recommended if only a small number (compared to the number of label sets) of training data is available, as is the case in most real-world applications.

Part II

Unsupervised Learning

Chapter 8

Introducing Clustering

Clustering is the assignment of data items into subgroups or *clusters* such that objects within the same cluster are "similar", while objects from different clusters are "different". This definition is very general and symptomatic for the situation in unsupervised learning: Widely accepted target functions or quality measures rarely exist. The goal of clustering varies between different approaches and applications. We review and discuss several objectives in Section 8.1.

Conventional clustering approaches assume that each data item belongs to exactly one cluster and therefore yield a partitioning into disjoint subsets. This assumption of mutually exclusive cluster assignments is too restrictive in many applications where the properties of a data set can be more adequately explained when data items might simultaneously belong to more than one cluster. Fuzzy clustering weakens this constraint by allowing partial memberships: An object can belong to several clusters, with a weight vector indicating the degree of the membership to a cluster. All membership weights sum up to 1. Classical single-assignment clustering is thus a special case of fuzzy clustering, where all weight is concentrated on a single cluster.

We present a novel approach which allows simultaneous assignments to several clusters. This approach, termed *multi-assignment clustering* and abbreviated *MAC*, goes beyond fuzzy clustering: Membership to a second cluster does not reduce the degree of membership in the first cluster. Standard (single-assignment) clustering is extended insofar as the weight vector only contains zeros and ones, but the weighted sum can exceed 1, indicating

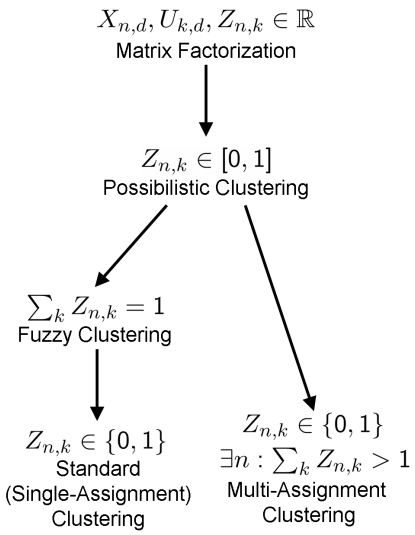


Figure 8.1: Overview over matrix factorization and clustering problems. An arrow denotes specialization. $Z_{n,\cdot}$ is an indicator vector, $Z_{n,k} = 1$ implies that the data item number n belongs to cluster k. While classical clustering as well as MAC require $Z_{n,k} \in \{0,1\}$, possibilistic and fuzzy clustering relax this constraint to $Z_{n,k} \in [0,1]$.

full membership to several clusters. An overview over different clustering problems and their relations is given in Figure 8.1.

Multi-assignment clustering is applicable in all situations where data items are described as vectors in a D-dimensional space. We study the clustering of Boolean data to exemplify our method. This focus is motivated by an important problem arising in computer security, namely the efficient and secure management of user privileges for access control system. Previous work on clustering of Boolean data is reviewed in Section 8.2, followed by an introduction of the security application in Section 8.3. The generative model to cluster Boolean data and the inference process are described in detail in Chapter 9. Finally, in Chapter 10, we apply an information-theoretic model for cluster validation to evaluate and compare several clustering algorithms without largely relying on specific assumptions.

8.1 Objectives for (Boolean) Data Clustering

Besides the task of partitioning data items into similar groups, we understand clustering also as a method to derive descriptive statistics about the data. Each clusters substantially differs from the other clusters in some of its properties. Accordingly, the formal objectives of data clustering can be split into two groups: The first group focusses on the reconstruction of a given data set, while in the second group, the emphasis lies on inference of an assumed underlying structure.

In the following, we assume that the data set consists of N tuples in D dimensions. We represent the data set as an $N \times D$-matrix $\mathbf{X}$, where the n^{th} row $X_{n,\cdot}$ represents the n^{th} data item. $X_{n,d}$ is the value of the n^{th} data item in dimension d. The cluster memberships are coded in a binary matrix $\mathbf{Z} \in \{0,1\}^{N \times K}$, while the source centroids are stored in the $K \times D$-dimensional matrix $\mathbf{U}$. $U_{k,d}$ describes the centroid value of the k^{th} source in dimension d.

We restrict ourselves to Boolean input data and clusters with Boolean centroids, i.e. we have $X_{n,d} \in \{0,1\}$ and $U_{k,d} \in \{0,1\}$. Hence, the source emissions are not stochastic but completely determined by the source centroids. For a compact description of the matrix decomposition, we define the matrix multiplication operator $\otimes$ such that

$$\mathbf{X} = \mathbf{Z} \otimes \mathbf{U} \quad \Longleftrightarrow \quad X_{n,d} = \bigvee_k \left[Z_{n,k} \wedge U_{k,d} \right] \ . \tag{8.1}$$

Data item $X_{n,\cdot}$ is obtained from the source centroids $\mathbf{U}$ given the membership indicator vector $Z_{n,\cdot}$ as the Boolean matrix product, i.e. the combination function (see Section 2.1.2) $\mathrm{c}(\cdot,\cdot)$ is given by

$$X_{n,\cdot} = \mathrm{c}(\mathbf{U}, Z_{n,\cdot}) = Z_{n,\cdot} \otimes \mathbf{U} \ ,$$

The same formalism is applicable to other data types. As an example, defining the multiplication operator as the conventional matrix product describes the superposition of continuous source emissions.

Data Reconstruction

When focusing on data reconstruction, the goal is to explain a given data set in an optimal way. Two main problem formulations exist from this point of view. In the first formulation, the number of clusters is given and the goal

is to minimize the reconstruction error. This criterion is termed *Min-Noise Approximation*.

Definition 7. *(**Min-Noise Approximation**) Given* $\mathbf{X}$*,* K *and the matrix norm* p*, find the matrices* $\mathbf{Z} \in \{0,1\}^{N \times K}$ *and* $\mathbf{U} \in \{0,1\}^{K \times D}$ *as*

$$(\hat{\mathbf{Z}}, \hat{\mathbf{U}}) = \arg\min_{\mathbf{Z},\mathbf{U}} ||\mathbf{X} - \mathbf{Z} \otimes \mathbf{U}||_p \ .$$

Alternatively, the reconstruction error is bounded by δ and the goal is to find the minimal number of sources such that the reconstruction error does not exceed the target value. This setting is termed *δ-Approximation*.

Definition 8. *(δ-**Approximation**) Given* $\mathbf{X}$*,* $\delta \geq 0$ *and* p*, find the minimal number of roles* K *and the matrices* $\mathbf{U}$ *and* $\mathbf{Z}$ *such that*

$$||\mathbf{X} - \mathbf{Z} \otimes \mathbf{U}||_p < \delta \ .$$

The order p of the norm is set to 1 for Boolean data. For continuous data, the order is typically chosen to be $p = 2$.

Inference

Inference methods aim at determining the structure which is believed to have generated to observed data.

Definition 9. *(**Structure Inference**) Given* $\mathbf{X}$ *and the assumption that* $\mathbf{X}$ *has an underlying structure* $\mathbf{X}^S = \mathbf{Z} \otimes \mathbf{U}$*, which is altered by a noise process:* $\mathbf{X}^S \sim P(\mathbf{X}^S; \mathbf{Z}, \mathbf{U})$*,* $\mathbf{X} \sim P(\mathbf{X}; \mathbf{X}^S, \Theta^N)$*. Find the structure encoded in* $\mathbf{Z}$ *and* $\mathbf{U}$*.*

There are two important differences between inference and reconstruction. First, a complete reconstruction of the data set $\mathbf{X}$ with the structure $\mathbf{Z} \otimes \mathbf{U}$ is not desired for inference, as these methods explicitly assume a noise process. The observed data $\mathbf{X}$ is thus assumed to be a noisy version of the structure. Trying to completely reconstruct the noisy data with a structure would imply to adapt to the noise in the data set and thus would yield inaccurate estimators for the structure. Second, inference methods yield a generative model for the data. Given such a model, it is possible to add additional data items to the existing clusters. The explanation obtained for the data through the clustering process can be generalized to new data which is not available during the inference phase. See Section 2.4.3 and Chapter 10 for details on how we measure the ability of the inferred structure to generalize to new data.

8.2 Related Work

Clustering Boolean data is a demanding problem. Starting from the formulation as min-noise or δ-approximation problem (Definitions 7 and 8), the problem is also known as *Boolean matrix factorization*. Standard approaches for general matrix factorization, such as singular value decomposition (SVD), have been adapted for this problem but often yield poor results. These methods neither take into account the limitation to values 0 and 1 for all matrices, nor the particularities of the Boolean matrix product (Eq. 8.1). While the first K singular vectors might still yield a relatively accurate decomposition, rounding them to obtain $\mathbf{Z}$ and $\mathbf{U}$ yields very poor results in both decomposition and prediction. The singular value decomposition is, however, useful to denoise a Boolean matrix [84, 85].

To adapt independent component analysis (ICA) [28] for Boolean data, the assumption of orthogonal centroids is maintained, while the mixture weights are constrained to be non-negative and sum up to 1. The generative model formulated based on these assumptions is then solved with a variational approach [68]. Binary independent component analysis (BICA) yields good results on data which is generated from sources with orthogonal centroids, but this assumption is too strict for most real-world data.

The Discrete Basis Problem Solver (DBPS) [84] is a greedy algorithm which optimally describes the rows of a Boolean matrix as combinations of basis vectors. The candidate set of basis vectors is determined by considering the individual rows as well as intersections between them, an idea inspired by association rule mining [1]. A predefined number of basis vectors is then iteratively chosen such that the approximation quality maximally increases in each step. Further combinatorial methods [26, 43] have been specially designed for the application of role-mining (see next section). However, we consider the DBPS as the best representative of combinatorial methods. Note that there is no generative model underlying this type of approaches.

A non-parametric generative model for Boolean data was proposed in [119]. Emissions of individual sources are combined by the noisy-OR function [87], which adds additional entries 1 in the matrix. The model for cluster assignments is the Indian Buffet Process (IBP) [52], where an infinite number of clusters is available, but only a finite number of them is responsible for the observed data. We refer to this model as the *Infinite Noisy-Or* (INO).

8.3 Application: Role Mining

In the modern economy, information constitutes a critical resource in everyday business. Data represents a value as it is expensive to collect and often builds the basis for current and future operation. Furthermore, especially for companies in the service industry, data is often confidential, as it might reveal details about the health or financial situation of customers. A professional, secure handling of such data is thus essential to build up the indispensable trust between the client and the service provider. Companies therefore have to control the use their computer and network resources as well as read and write permissions on their data.

The principle of *least privileges* states that users should not have any privileges except those needed to fulfill the job. Assigning extra privileges to users is understood as a security risk, as a malicious employee might abuse his or her permissions and inflict damage on the company.

Following the principle of least privileges, it is reasonable to assume that employees with similar job duties share similar permissions. Deviations from this assumption can have two causes: A user might get permissions for a special duty besides the normal tasks. The respective permissions are granted for a limited time and have to be revoked once the task is completed. The second type of irregularities results from erroneous maintenance of the user-permission matrix. Such irregularities are critical to the security of the computer system. An access control system which is able to highlight irregular permissions allows the IT security group to discover, review and possibly revoke such irregular permissions and thus increases the security of the computer system.

The standard approach for access control consists of designing a binary access-control matrix $\mathbf{X} \in \{0,1\}^{N \times D}$, with

$$X_{n,d} = \begin{cases} 1 & \text{if user } n \text{ has access to resource } d \\ 0 & \text{if user } n \text{ has no access to resource } d \end{cases}$$

This approach is called *direct access control* and is still used in many companies. However, it has two important drawbacks: First, maintaining the access-control matrix involves a lot of human work, as a large number of permissions has to be set whenever a new employee joins the company or when an employee changes position within the company, e.g. due to a promotion. For this reason, direct access control is expensive and error-prone. Second, irregularities in the access-control matrix are very difficult to detect, and security leaks might remain undiscovered for many years.

Role-based access control (RBAC) formalizes the expected regularities in the user permission matrix. Instead of direct assignments of users to permissions, access is granted via *roles*: Each role defines a set of permissions. Users are assigned to roles and obtain all permissions of the roles they are assigned to. The definition of the industrial standard for RBAC [44] explicitly states that users might belong to several roles. This design freedom yields a clear and interpretable structure of the roles. Basic permissions (such as checking email) can be covered by a role shared by all employees, while specialized permissions are granted via specific roles. If multiple assignments were not allowed, a new role would have to be defined for each combination. Doing so would dramatically increase the number of roles.

The design of the set of roles is a crucial step when setting up an RBAC system. Most companies start with a direct access control system. The problem of *Role Mining* [73] consists of identifying roles in a user-permission matrix $\mathbf{X}$ and thus describes a data-centered approach to replace the direct assignments by a user-role assignment matrix $\mathbf{Z}$ and a role-permission assignment matrix $\mathbf{U}$. Several formal definitions of the role mining problem have been proposed in the literature [112, 46]. Based on the assumption that permissions are granted based on the job profile, we regard role-mining as an inference problem. Furthermore, role-based access control requires a set of roles that allows the system administrator to equip new employees with all required permissions. This step corresponds to the generalization step discussed in Section 2.4.3.

From the business perspective, the roles should be interpretable as functional roles within the company [27]. From a generative point of view, several sets of roles might be roughly equally well suited to explain the given user-permission matrix. In such a case, a particular set of roles can be chosen such that the agreement with the provided business information is maximized. We propose a probabilistic method to combine a generative role-mining method with business information [47]. Experiments on data from a Swiss company show that our approach yields a role set with largely improved interpretability, while the generalization ability of the roles only insignificantly deteriorates.

Chapter 9

Generative Multi-Assignment Clustering

To address the challenging problem of role mining, we concretize the general generative model introduced in Chapter 2 for Boolean data. We describe the inference procedure in detail and discuss some theoretical aspects of the model. In experiments on synthetic and real-world data, we show that our proposed method for multi-assignment clustering outperforms state-of-the-art algorithms in both parameter accuracy and generalization ability.

9.1 Structure and Noise Models

We first present a probabilistic model for the structure in the data. In Section 9.1.2 we describe two noise models and present a unifying view for both of them.

9.1.1 Structure Model

The structure model is formalized by the Boolean matrix product $\mathbf{x}^S = \mathbf{z} \otimes \mathbf{u}$, with the operator $\otimes$ defined in Eq. 8.1. Given $\mathbf{z}$ and $\mathbf{u}$, the matrix

$\mathbf{x}^S$ is completely determined: The probability distribution

$$p^S(x_{n,d}|\mathbf{z},\mathbf{u}) = \left[\prod_{k=1}^{K} u_{k,d}^{z_{n,k}}\right]^{x_{n,d}^S} \left[1 - \prod_{k=1}^{K} u_{k,d}^{z_{n,k}}\right]^{1-x_{n,d}^S} \tag{9.1}$$

is a point mass. Note that the Boolean nature of the data allows us to use the data as an indicator variable: In the above equation, the first factor is only relevant if $x_{n,d}^S = 1$. To simplify the notation, we introduce the assignment set $\mathcal{L}_n$ as the set of all sources data item x_n is assigned to:

$$\mathcal{L}_n := \{k \in \{1,\ldots,K\} | z_{n,k} = 1\} \tag{9.2}$$

The signal contribution of a data item that belongs to the assignment set $\mathcal{L}$ is given by

$$u_{\mathcal{L},d} := \bigvee_{k\in\mathcal{L}} u_{k,d} \ , \tag{9.3}$$

The probability distribution $p^S(x_{n,d}|\mathcal{L}_n,\mathbf{u})$ can then be written as

$$p^S(x_{n,d}|\mathcal{L},\mathbf{u}) = [u_{\mathcal{L},d}]^{x_{n,d}^S} [1 - u_{\mathcal{L},d}]^{1-x_{n,d}^S} \ . \tag{9.4}$$

Searching the optimal matrices $\mathbf{z}$ and $\mathbf{u}$ for a given $\mathbf{x}^S$ is a combinatorial optimization problem and is proven to be NP-hard [113]. Furthermore, due to noise effects, the data $\mathbf{x}$ are random variables. We therefore switch over to a probabilistic representation and treat also $\mathbf{u}$ as a random variable. We assume each element of $\mathbf{u}$ to be independently distributed according to a Bernoulli distribution with parameter $\beta_{n,d} := p(u_{n,d} = 0)$. Defining the Bernoulli parameter $\beta_{n,d}$ as the probability of $u_{n,d} = 0$ and not, as usual, as the probability of $u_{n,d} = 1$, allows a compact notation: The Boolean OR of a series of bits is zero only if all bits are zero, while all other inputs yield a 1. Defining $\beta_{n,d} := p(u_{n,d} = 0)$ captures this asymmetry: The probability of $u_{\mathcal{L},d} = 0$ is given by

$$p(u_{\mathcal{L},d} = 0) = p\left(\sum_{k\in\mathcal{L}} u_{k,d} = 0\right) = \prod_{k\in\mathcal{L}} p(u_{k,d} = 0) = \prod_{k\in\mathcal{L}} \beta_{k,d} =: \beta_{\mathcal{L},d} \ .$$

To obtain the probability distribution of $x_{n,d}^S$ given the parameters $\boldsymbol{\beta}$, we

integrate out the Boolean centroids $\mathbf{u}$ and obtain

$$p(x^S_{n,d}|\mathbf{z},\boldsymbol{\beta}) = \sum_{\{u_{\cdot,d}\}} \left\{p^S(x_{n,d}|\mathbf{z},u_{\cdot,d}) \cdot p(u_{\cdot,d})\right\}$$
$$= \left[\prod_{k=1}^{K} \beta_{k,d}^{z_{n,k}}\right]^{1-x^S_{n,d}} \left[1 - \prod_{k=1}^{K} \beta_{k,d}^{z_{n,k}}\right]^{x^S_{n,d}} \tag{9.5}$$

The matrix $\mathbf{u} \in \{0,1\}^{N\times D}$ is thus replaced by $\boldsymbol{\beta} \in [0,1]^{N\times D}$. This reformulation of the problem has two important advantages: First, the optimization problem is drastically simplified and allows us to find a solution in a reasonable amount of time. Second, the probabilistic representation enables us to detect exceptional elements in the matrix $\mathbf{x}$. These elements are separated out. In this manner, inference is not disturbed by irregular matrix entries and yields accurate parameter estimates even from data with high noise level.

Using $\beta_{\mathcal{L},d}$ as the parameter of the proxy distribution of data items with label set $\mathcal{L}$, the probability distribution of $x_{n,d}$ is given by

$$p^S(x^S_{n,d}|\mathcal{L},\boldsymbol{\beta}) = [\beta_{\mathcal{L},d}]^{1-x^S_{n,d}} [1-\beta_{\mathcal{L},d}]^{x^S_{n,d}} \quad . \tag{9.6}$$

We emphasize that $\beta_{\mathcal{L},d}$ is only introduced for notational convenience and is always derived from the parameters of the single sources $\beta_{k,d}$.

9.1.2 Noise Models

Besides the structure, a separate noise process can influence the value of $x_{n,d}$. We discuss two different global noise models in detail and then present a unifying global noise model. Variants of the noise models with local noise processes are briefly discussed at the end of this section.

The term "noise" has a negative connotation, which might not be adequate in all applications. For example in role mining, some deviations from the structure are justified, as explained in Section 8.3. However, in order to underline their irregular emergence, we keep the term "noise" to describe such exceptions.

We assume that all random variables are independent for all $n = 1,\ldots,N$ and $d = 1,\ldots,D$. The probability of a data matrix $\mathbf{x}$ is thus given by the product over the probability of its elements. Formally we introduce $\mathfrak{m}$ as an indicator for the noise model and denote by $\boldsymbol{\theta}^{\mathfrak{m}}_N$ the parameters of the

noise model. The entire parameter tuple of the generative model including the noise model $\mathfrak{m}$ is denoted by $\boldsymbol{\theta}^{\mathfrak{m}} := (\boldsymbol{\beta}, \boldsymbol{\theta}_N^{\mathfrak{m}})$. Then, we obtain

$$p^{\mathfrak{m}}(\mathbf{x}|\boldsymbol{\mathcal{L}}, \boldsymbol{\theta}^{\mathfrak{m}}) = \prod_{n=1}^{N} \prod_{d=1}^{D} p^{\mathfrak{m}}(x_{n,d}|\mathcal{L}_n, \boldsymbol{\theta}^{\mathfrak{m}}) \ . \tag{9.7}$$

The probability distributions $p^{\mathfrak{m}}_{x_{n,d}}(x_{n,d}|\mathcal{L}_n, \boldsymbol{\theta}^{\mathfrak{m}})$ for the individual noise models are detailed below.

Mixture Noise Model

The mixture noise model assumes a separate global Bernoulli process which generates noisy bits $X^N_{n,d}$. The distribution of $X^N_{n,d}$ is parameterized by the *noise parameter* $r \in [0, 1]$ indicating the probability of a noise bit to be 1:

$$X^N_{n,d} \sim Ber(r) \ , \qquad p^{mix}(x^N_{n,d}) = r^{x^N_{n,d}} \cdot (1-r)^{1-x^N_{n,d}} \ . \tag{9.8}$$

We introduce a binary indicator $\upsilon_{n,d}$ to indicate whether $x_{n,d}$ is generated by the noise process ($\upsilon_{n,d} = 1$) or by the structure process ($\upsilon_{n,d} = 0$). The full generative process for $x_{n,d}$ is thus

$$x_{n,d} = \upsilon_{n,d} x^N_{n,d} + (1 - \upsilon_{n,d}) x^S_{n,d} \tag{9.9}$$

The indicators $\upsilon_{n,d}$ cannot be observed. We assume that they also follow a Bernoulli distribution. The parameter of the distribution is denoted by ϵ and called *noise fraction*, as it indicates the expected ratio of noisy bits. Marginalizing out $\upsilon_{n,d}$, the overall probability distribution of $X^N_{n,d}$ is given by

$$\begin{aligned} p^{mix}(x_{n,d}|\boldsymbol{\mathcal{L}}_n, \boldsymbol{\beta}, r, \epsilon) = \epsilon \cdot r^{x_{n,d}}(1-r)^{1-x_{n,d}} \\ + (1-\epsilon) \cdot [\beta_{\mathcal{L},d}]^{1-x_{n,d}} [1-\beta_{\mathcal{L},d}]^{x_{n,d}} \ . \end{aligned} \tag{9.10}$$

The parameters of the mixture noise model are thus $\boldsymbol{\theta}^{mix}_N = (\epsilon, r)$. The mixture noise model is illustrated in Figure 9.1.

In order to be formulated in the framework of Section 2.2, a probabilistic combination function $\mathrm{c}^M_{\kappa^M}(x^S_{n,d}, x^N_{n,d})$ is used to combine the structure and the noise component to the observed value $x_{n,d}$:

$$\mathrm{c}^M_{\kappa^M}(x^S_{n,d}, x^N_{n,d}) = \begin{cases} x^S_{n,d} & \text{with probability } 1-\epsilon \\ x^N_{n,d} & \text{with probability } \epsilon \end{cases} \ . \tag{9.11}$$

In the above derivation, the indicator variables $\upsilon_{n,d}$ to capture the probabilistic nature of the combination function.

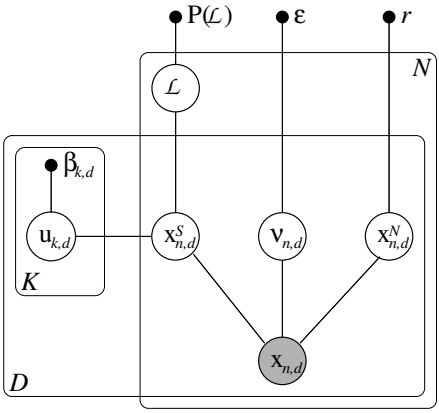


Figure 9.1: The generative model of multi-assignment clustering (MAC) with mixture noise model.

Flip Noise Model

In the flip noise model, we assume that a fraction of bits in $\mathbf{x}^S$ is flipped by the noise process. Introducing the binary flip indicator $\upsilon_{n,d}$ to specify whether $X^S_{n,d}$ is flipped ($\upsilon_{n,d} = 1$) or not ($\upsilon_{n,d} = 0$), the generative process for $x_{n,d}$ is described by

$$x_{n,d} = x^S_{n,d} \oplus \upsilon_{n,d} \ , \tag{9.12}$$

where $\oplus$ denotes the addition modulo 2, also known as the XOR gate.

In a general version, the probability of a bit flip depends on the original value of the bit. For example in the case of role-based access control, users call the help-desk if they are missing a permission. Lacking permissions are thus corrected, while unnecessary extra-permissions might remain undetected. Defining ϵ_0 (ϵ_1) as the probability of a bit flip from 0 to 1 (from 1 to 0), i.e. $\epsilon_0 := P(\upsilon_{n,d} = 1 | X^S_{n,d} = 0)$ and $\epsilon_1 := P(\upsilon_{n,d} = 1 | X^S_{n,d} = 1)$, the probability distribution for $\upsilon_{n,d}$ is given by

$$\begin{aligned} &p^{flip}(\upsilon_{n,d} | x^S_{n,d}, \epsilon_0, \epsilon_1) \\ &\quad = \left(\epsilon_1^{x^S_{n,d}} \cdot \epsilon_0^{1-x^S_{n,d}}\right)^{\upsilon_{n,d}} \cdot \left((1-\epsilon_1)^{x^S_{n,d}} \cdot (1-\epsilon_0)^{1-x^S_{n,d}}\right)^{1-\upsilon_{n,d}} \ . \end{aligned} \tag{9.13}$$

The joint probability distribution of $x_{n,d}$, $x^S_{n,d}$ and $\upsilon_{n,d}$ is then given by

$$p^{flip}(x_{n,d}, x^S_{n,d}, \upsilon_{n,d}|\mathcal{L}_n, \boldsymbol{\beta}, \epsilon_0, \epsilon_1) = p^{flip}(x_{n,d}|x^S_{n,d}, \upsilon_{n,d}) \cdot p^S(x^S_{n,d}|\mathcal{L}_n, \boldsymbol{\beta}) \cdot p^{flip}(\upsilon_{n,d}|x^S_{n,d}, \epsilon_0, \epsilon_1) \ .$$

Integrating out the unobserved variables $x^S_{n,d}$ and $\upsilon_{n,d}$, we get

$$\begin{aligned} p^{flip}(x_{n,d}|\mathcal{L}_n, \boldsymbol{\beta}, \epsilon_0, \epsilon_1) &= (1-\epsilon_0)\beta_{\mathcal{L}_n,d}(1-x_{n,d}) + (1-\epsilon_1)(1-\beta_{\mathcal{L}_n,d})x_{n,d} \\ &\quad + \epsilon_0\beta_{\mathcal{L}_n,d}x_{n,d} + \epsilon_1(1-\beta_{\mathcal{L}_n,d})(1-x_{n,d}) \ . \end{aligned} \tag{9.14}$$

The bit flip noise model is parameterized by $\boldsymbol{\theta}^{flip}_N = (\epsilon_0, \epsilon_1)$.

A special case of the bit flip model is the *symmetric bit flip model*, where a flip from 0 to 1 has the same probability as a flip from 1 to 0, i.e. $\epsilon_0 = \epsilon_1 =: \epsilon$. In this case, Eq. 9.14 simplifies to

$$\begin{aligned} p^{sflip}(x_{n,d}|\mathcal{L}_n, \boldsymbol{\beta}, \epsilon) &= (1-\epsilon)\left[\beta_{\mathcal{L}_n,d}(1-x_{n,d}) + (1-\beta_{\mathcal{L}_n,d})x_{n,d}\right] \\ &\quad + \epsilon\left[\beta_{\mathcal{L}_n,d}x_{n,d} + (1-\beta_{\mathcal{L}_n,d})(1-x_{n,d})\right] \ . \end{aligned} \tag{9.15}$$

The symmetric bit flip noise model has only one parameter: $\boldsymbol{\theta}^{sflip}_N = (\epsilon)$.

In the formulation of Section 2.2, we have $x_{n,d} = \upsilon_{n,d}$, and the combination function $\mathrm{c}^M_{\kappa^M}(x^S_{n,d}, x^N_{n,d})$ is given by the addition modulo 2 (Eq. 9.12).

Unified Noise Model

In order to compare the two noise models, we introduce the *bit set probability* $q^{\mathfrak{m}}_{\mathcal{L}_n,d}$ as the probability for $x_{n,d} = 1$ under noise model $\mathfrak{m}$, with $\mathfrak{m} \in \{mix, flip, sflip\}$. Reordering Eq. 9.10, and 9.14, we get

$$q^{mix}_{\mathcal{L}_n,d} = \beta_{\mathcal{L}_n,d}(\epsilon - 1) + (1-\epsilon) + \epsilon r \tag{9.16}$$

$$q^{flip}_{\mathcal{L}_n,d} = \beta_{\mathcal{L}_n,d}(\epsilon_0 + \epsilon_1 - 1) + 1 - \epsilon_1 \tag{9.17}$$

Equating the coefficients of $\beta_{\mathcal{L}_n,d}$ as well as the constant terms, we find the following parameter values for equal probabilities under the mixture and

the flip noise models:

$$\begin{pmatrix} \text{mixture noise model} \\ \text{with parameters} \\ \boldsymbol{\theta}_N^{mix} = (\epsilon, r) \end{pmatrix} \text{ is equivalent to } \begin{pmatrix} \text{bit flip noise model} \\ \text{with parameters} \\ \boldsymbol{\theta}_N^{flip} = (\epsilon r, \epsilon(1-r)) \end{pmatrix}$$

$$\begin{pmatrix} \text{bit flip noise model} \\ \text{with parameters} \\ \boldsymbol{\theta}_N^{flip} = (\epsilon_0, \epsilon_1) \end{pmatrix} \text{ is equivalent to } \begin{pmatrix} \text{mixture noise model} \\ \text{with parameters} \\ \boldsymbol{\theta}_N^{mix} = \left(\epsilon_0 + \epsilon_1, \frac{\epsilon_0}{\epsilon_0+\epsilon_1}\right) \end{pmatrix}$$

Note that the conversion is not possible for all parameter values of the models. In particular, in the noise-free setting, we have $\epsilon_0 = \epsilon_1 = 0$ for the flip noise model, while the value of r in the mixture noise model is undefined. However, since no samples are drawn from the noise source in this setting, the value of r is irrelevant for the data probabilities. In the opposite case of high flip probabilities, the noise fraction $\epsilon = \epsilon_0 + \epsilon_1$ could reach values higher than 1, which renders the probability of a noisy bit undefined. This is due to a particularity of the bit flip model: Flipping all bits does actually preserve the complete information. The highest uncertainty is introduced when both flip probabilities are equal to $\epsilon_0 = \epsilon_1 = 1/2$. In an information-theoretic setting, the bit flip noise model corresponds to the binary symmetric channel [29]. Given the equivalence and the anomalies of the flip noise model, we only consider the mixture noise model in the following.

Local Noise Models

The global noise models presented above can be extended to a noise process which depends either on the dimension d or on the data item n. Doing so, either the global noise fraction ϵ, the global noise parameter r or both of them are replaced by dimension- or data-item wise parameters. We give details for the case where both ϵ and r are local, the reduction to the case where one parameter remains global is straight-forward.

Dimension-Wise Noise Process. The probability of $x_{n,d}$ under the dimension-wise mixture noise model becomes

$$\begin{aligned} p_{x_{n,d}}^{d-mix}(x_{n,d}|\mathcal{L}_n, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\epsilon}) = \epsilon_d \cdot r_d^{x_{n,d}}(1-r_d)^{1-x_{n,d}} \\ + (1-\epsilon_d) \cdot [\beta_{\mathcal{L}_n,d}]^{1-x_{n,d}} [1-\beta_{\mathcal{L}_n,d}]^{x_{n,d}} \quad . \end{aligned}$$

Data-Item-Wise Noise Process. Eq. 9.10 is replaced by

$$p^{n-mix}_{x_{n,d}}(x_{n,d}|\mathcal{L}_n, \boldsymbol{\beta}, \boldsymbol{r}, \boldsymbol{\epsilon}) = \epsilon_n \cdot r_n^{x_{n,d}}(1-r_n)^{1-x_{n,d}} + (1-\epsilon_n)\cdot[\beta_{\mathcal{L}_n,d}]^{1-x_{n,d}}\,[1-\beta_{\mathcal{L}_n,d}]^{x_{n,d}} \quad .$$

These local noise models are very specific and able to describe particularities of the data. For example in role mining, it seems plausible that the error probability depends on the type of permission: The IT security group is probably more careful when granting root permissions to the central database than when allowing a user to change the background image on the desktop.

9.2 Inference

As stated earlier, we choose the parameters according to the principle of *maximum a posteriori*. As both the assignment sets as well as the parameter values are to be determined, we solve this optimization problem by alternating between an estimation step and a maximization step. In the estimation step (abbreviated as *E-step*), the assignments of data items to clusters are estimated. Hence, the hard assignments $\mathbf{z}$ of data items to single assignment sets are replaced by the posterior distribution over the assignment sets, given the parameters and the data item:

$$p^{\mathfrak{m}}(\mathcal{L}|x_{n,\cdot}, \boldsymbol{\theta}^{\mathfrak{m}}) = \frac{p^{\mathfrak{m}}(x_{n,\cdot}|\mathcal{L}, \boldsymbol{\theta}^{\mathfrak{m}})\cdot p(\mathcal{L})}{\sum_{\mathcal{L}'} p^{\mathfrak{m}}(x_{n,\cdot}|\mathcal{L}', \boldsymbol{\theta}^{\mathfrak{m}})\cdot p(\mathcal{L}')} \tag{9.18}$$

with

$$p^{\mathfrak{m}}(x_{n,\cdot}|\mathcal{L}, \boldsymbol{\theta}^{\mathfrak{m}}) = \prod_{d=1}^{D} p^{\mathfrak{m}}(x_{n,d}|\mathcal{L}, \boldsymbol{\theta}^{\mathfrak{m}}) \ . \tag{9.19}$$

$p(\mathcal{L})$ denotes the prior probability of assignment set $\mathcal{L}$. In the following, we assume a uniform prior, i.e. $p(\mathcal{L}) = 1/|\mathbb{L}|$ for all $\mathcal{L}$. The maximization step (*M-step*) consists of optimizing the parameters such that the data is most probably under the assignments computed in the E-step. This algorithm is called *estimation-maximization* (EM) algorithm [33].

The probability of the data matrix $\mathbf{x}$ given in Eq. 9.7 is highly non-convex in the parameters, and a direct maximization would most likely be trapped in a local optimum. We therefore modify the estimation step by

introducing a parameter T, called the *computational temperature*, to vary the width of the posterior distribution over the assignment sets:

$$\gamma^{\mathrm{m}}_{n,\mathcal{L}} := p(\mathcal{L}|x_{n,\cdot},\boldsymbol{\theta}^{\mathrm{m}}) = \frac{\left(p^{\mathrm{m}}(x_{n,\cdot}|\mathcal{L},\boldsymbol{\theta}^{\mathrm{m}})\cdot p(\mathcal{L})\right)^{1/T}}{\sum_{\mathcal{L}'}\left(p^{\mathrm{m}}(x_{n,\cdot}|\mathcal{L}',\boldsymbol{\theta}^{\mathrm{m}})\cdot p(\mathcal{L}')\right)^{1/T}} \tag{9.20}$$

The limit of $T \to \infty$ yields the uniform distribution over all assignment sets. Starting at a high value of T, the temperature is slowly decreased. Doing so, the influence of the posterior probability grows and the assignments become harder. For $T = 1$, we get back Eq. 9.18. This techniques is well-known as *deterministic annealing* [96, 21] and usually formulated in terms of a *risk* function R, which maps a clustering solution and the data to a real number. To formulate our approach in this setting, we define the risk of assigning data item n to assignment set $\mathcal{L}$ as the negative log-likelihood of the feature vector $x_{n,\cdot}$:

$$R^{\mathrm{m}}_{n,\mathcal{L}} := -\log\left(p^{\mathrm{m}}(x_{n,\cdot}|\mathcal{L},\boldsymbol{\theta}^{\mathrm{m}})\right) = \sum_{d=1}^{D}\log\left(p^{\mathrm{m}}(x_{n,d}|\mathcal{L},\boldsymbol{\theta}^{\mathrm{m}})\right) \tag{9.21}$$

Corresponding to Eq. 9.20, the posterior distribution of the label sets is computed as $\gamma^{\mathrm{m}}_{n,\mathcal{L}} = \exp(-R^{\mathrm{m}}_{n,\mathcal{L}}/T)/\sum_{\mathcal{L}'}\exp(-R^{\mathrm{m}}_{n,\mathcal{L}'}/T)$. Deterministic annealing minimizes the Lagrange functional

$$F := -T\log Z = \overline{R} - T\cdot H\ , \tag{9.22}$$

where Z is the *state sum*:

$$Z^{\mathrm{m}} := \prod_{n=1}^{N}\sum_{\mathcal{L}}\exp\left(-R^{\mathrm{m}}_{n,\mathcal{L}}/T\right) \tag{9.23}$$

In statistical physics, F is called the *free energy* of the system. Alternatively to Eq. 9.22, it can be computed as

$$F^{\mathrm{m}} := -T\log Z^{\mathrm{m}} = -T\sum_{n}\log\left(\sum_{\mathcal{L}}\exp\left(-R^{\mathrm{m}}_{n,\mathcal{L}}/T\right)\right) \tag{9.24}$$

The right-hand side of Eq. 9.22 shows how the computational temperature T determines the trade-off between minimizing the expected risk $\overline{R} := \sum_n\sum_{\mathcal{L}}\gamma^{\mathrm{m}}_{n,\mathcal{L}}R^{\mathrm{m}}_{n,\mathcal{L}}$ and the entropy H of the assignment probabilities $\gamma^{\mathrm{m}}_{n,\mathcal{L}}$.

Given the probabilistic assignments $\gamma^{\mathfrak{m}}_{n,\mathcal{L}}$ (Eq. 9.20) estimated in the E-step, F is minimized as a function of the model parameters in the M-step. We do this by setting the derivative of the free energy $F^{\mathfrak{m}}$ with respect to the generic parameter θ to zero:

$$\frac{\partial F^{\mathfrak{m}}}{\partial \theta} = \sum_n \sum_{\mathcal{L}} \gamma^{\mathfrak{m}}_{n,\mathcal{L}} \frac{\partial R^{\mathfrak{m}}_{n,\mathcal{L}}}{\partial \theta} \tag{9.25}$$

$$= \sum_n \sum_{\mathcal{L}} \gamma^{\mathfrak{m}}_{n,\mathcal{L}} \sum_d \frac{(1-2x_{n,d}) \frac{\partial q^{\mathfrak{m}}_{\mathcal{L},d}}{\partial \theta}}{x_{n,d}\left(1-q^{\mathfrak{m}}_{\mathcal{L},d}\right) + (1-x_{n,d}) q^{\mathfrak{m}}_{\mathcal{L},d}} \stackrel{!}{=} 0 \ . \tag{9.26}$$

As an explicit solution for the optimal parameter values only exists for some very small problems, we numerically determine the parameter values using Brent's method [15]. This elaborate root-finding algorithm combines bisection search, the secant method and inverse quadratic interpolation. Brent's method reaches the reliability of bisection search at the computational speed of the less reliable methods (bisection search and the secant method).

We first update the noise parameters $\boldsymbol{\theta}^{\mathfrak{m}}_N$ and then the centroid probabilities $\boldsymbol{\beta}$ independently for each centroid μ and each dimension ν. When updating $\beta_{\mu,\nu}$, we use the values of $\beta_{p,q}$, $p \neq \mu$ and $q \neq \nu$, of the previous iteration. Only at the end of the update procedure, the values of the matrix $\boldsymbol{\beta}$ are overwritten. This assumption is clearly a simplification but allows to drastically reduce the computation time.

In the following, we give the optimality conditions for the models presented above. For a compact notation, we introduce

$$g^{0,\mathfrak{m}}_{\mathcal{L},\nu} := \sum_{n:x_{n,\nu}=0} \gamma^{\mathfrak{m}}_{n,\mathcal{L}} \qquad g^{1,\mathfrak{m}}_{\mathcal{L},\nu} := \sum_{n:x_{n,\nu}=1} \gamma^{\mathfrak{m}}_{n,\mathcal{L}} \ .$$

Optimality Conditions for the Mixture Noise Model. The derivatives of the bit set probabilities $q^{mix}_{\mathcal{L},d}$ with respect to the parameters of the mixture noise model ($\theta \in \{\beta_{\mu,\nu}, \epsilon, r\}$) are as follows:

$$\frac{\partial q^{\text{mix}}_{\mathcal{L},d}}{\partial \beta_{\mu,\nu}} = (1-\epsilon)\, \beta_{\mathcal{L}\setminus\mu,d} 1_{\nu=d} 1_{\mu\in\mathcal{L}} \qquad \frac{\partial q^{\text{mix}}_{\mathcal{L},d}}{\partial \epsilon} = 1 - r - \beta_{\mathcal{L},d} \qquad \frac{\partial q^{\text{mix}}_{\mathcal{L},d}}{\partial r} = -\epsilon$$

With $w := \epsilon r + (1-\epsilon)(1-\beta_{\mathcal{L},\nu})$, this results in the following extremality conditions for the mixture noise model:

$$\frac{\partial F^{\text{mix}}}{\partial \beta_{\mu,\nu}} = (1-\epsilon) \sum_{\mathcal{L} \ni \mu} \beta_{\mathcal{L}\setminus\mu,\nu} \left\{ \frac{g^{1,\text{mix}}_{\mathcal{L},\nu}}{w} - \frac{g^{0,\text{mix}}_{\mathcal{L},\nu}}{1-w} \right\} = 0$$

$$\frac{\partial F^{\text{mix}}}{\partial \epsilon} = \sum_d \left\{ \sum_{\mathcal{L}} \frac{(1-r-\beta_{\mathcal{L},d}) g^{1,\text{mix}}_{\mathcal{L},d}}{w} - \sum_{\mathcal{L}} \frac{(1-r-\beta_{\mathcal{L},d}) g^{0,\text{mix}}_{\mathcal{L},d}}{1-w} \right\} = 0$$

$$\frac{\partial F^{\text{mix}}}{\partial r} = \epsilon \sum_d \left\{ \sum_{\mathcal{L}} \frac{g^{0,\text{mix}}_{\mathcal{L},d}}{1-w} - \sum_{\mathcal{L}} \frac{g^{1,\text{mix}}_{\mathcal{L},d}}{w} \right\} = 0$$

Again, numerical root finding methods are employed to determine the optimal values of the parameters $\beta_{\mu,\nu}$, ϵ and r.

Optimality Conditions for the Bit Flip Noise Model. We define $\varepsilon := 1-\epsilon_0-\epsilon_1$ to simplify the notation. The derivatives for the asymmetric bit flip model ($\theta \in \{\beta_{\mu,\nu}, \epsilon_0, \epsilon_1\}$) are:

$$\frac{\partial q^{\text{asym}}_{\mathcal{L},d}}{\partial \beta_{\mu,\nu}} = \varepsilon \beta_{\mathcal{L}\setminus\mu,\nu} 1_{\{\nu=d\}} 1_{\{\mu\in\mathcal{L}\}} \qquad \frac{\partial q^{\text{asym}}_{\mathcal{L},d}}{\partial \epsilon_0} = -\beta_{\mathcal{L},d} \qquad \frac{\partial q^{\text{asym}}_{\mathcal{L},d}}{\partial \epsilon_1} = 1-\beta_{\mathcal{L},d}$$

Setting the derivatives of the free energy F with respect to the respective parameters to zero results in the following update conditions for the optimal parameter values:

$$\frac{\partial F^{\text{asym}}}{\partial \beta_{\mu,\nu}} = \varepsilon \sum_{\mathcal{L} \ni \mu} \beta_{\mathcal{L}\setminus\mu,\nu} \left\{ \frac{g^{1,\text{asym}}_{\mathcal{L},\nu}}{1-\epsilon_1-\varepsilon\beta_{\mathcal{L},\nu}} - \frac{g^{0,\text{asym}}_{\mathcal{L},\nu}}{\epsilon_1+\varepsilon\beta_{\mathcal{L},\nu}} \right\} = 0$$

$$\frac{\partial F^{\text{asym}}}{\partial \epsilon_0} = \sum_d \left\{ \sum_{\mathcal{L}} \frac{\beta_{\mathcal{L},d} g^{0,\text{asym}}_{\mathcal{L},d}}{\epsilon_1+\varepsilon\beta_{\mathcal{L},d}} - \sum_{\mathcal{L}} \frac{\beta_{\mathcal{L},d} g^{1,\text{asym}}_{\mathcal{L},d}}{1-\epsilon_1-\varepsilon\beta_{\mathcal{L},d}} \right\} = 0$$

$$\frac{\partial F^{\text{asym}}}{\partial \epsilon_1} = \sum_d \left\{ \sum_{\mathcal{L}} \frac{(1-\beta_{\mathcal{L},d}) g^{1,\text{asym}}_{\mathcal{L},d}}{1-\varepsilon\beta_{\mathcal{L},d}} - \sum_{\mathcal{L}} \frac{(1-\beta_{\mathcal{L},d}) g^{0,\text{asym}}_{\mathcal{L},d}}{\epsilon_1+\varepsilon\beta_{\mathcal{L},d}} \right\} = 0$$

Optimality Conditions for the Symmetric Flip Noise Model. With $\epsilon_0 = \epsilon_1 = \epsilon$ and setting $\varepsilon := 1-2\epsilon$, the above optimality conditions for the

symmetric flip noise model are:

$$\frac{\partial F^{\text{sym}}}{\partial \beta_{\mu,\nu}} = \varepsilon \sum_{\mathcal{L} \ni \mu} \beta_{\mathcal{L}\setminus\mu,\nu} \left\{ \frac{g^{1,\text{sym}}_{\mathcal{L},\nu}}{1-\epsilon-\varepsilon\beta_{\mathcal{L},\nu}} - \frac{g^{0,\text{sym}}_{\mathcal{L},\nu}}{\epsilon+\varepsilon\beta_{\mathcal{L},\nu}} \right\} = 0$$

$$\frac{\partial F^{\text{sym}}}{\partial \epsilon} = \sum_{d} \left\{ \sum_{\mathcal{L}} \frac{(1-2\beta_{\mathcal{L},d}) g^{1,\text{sym}}_{\mathcal{L},d}}{1-\epsilon-\varepsilon\beta_{\mathcal{L},d}} - \sum_{\mathcal{L}} \frac{(1-2\beta_{\mathcal{L},d}) g^{0,\text{sym}}_{\mathcal{L},d}}{\epsilon+\varepsilon\beta_{\mathcal{L},d}} \right\} = 0$$

Inference is done in the same way as for the global noise models.

Estimating the Centroids u. As stated above, we infer the probabilistic centroids $\hat{\boldsymbol{\beta}}$, while a Boolean description $\hat{\mathbf{u}}$ of the centroids is required to solve the problem of Boolean matrix factorization. We obtain $\hat{\mathbf{u}}$ by rounding $\hat{\boldsymbol{\beta}}$, formally

$$\hat{u}_{n,d} = \begin{cases} 1 & \text{if } \hat{\beta}_{n,d} \geq 0.5 \\ 0 & \text{if } \hat{\beta}_{n,d} < 0.5 \end{cases} . \tag{9.27}$$

In our experiments, we observe that the probabilistic centroids $\hat{\boldsymbol{\beta}}$ converge towards either 0 or 1 unless there is a model mismatch or the algorithm is severely stuck in a local optimum.

9.3 Equivalent Single-Assignment Clustering

In this section, we describe some of the theoretical properties of the multi-assignment clustering model. Recall that $\mathbb{L}$ is the set of all admissible assignment sets. $L := |\mathbb{L}|$ denotes the number of possible assignment sets. Assuming an arbitrary, but fixed numbering of the admissible assignment sets, $\mathbb{L}$ can be encoded as a binary matrix $\mathbf{z}^{\mathbb{L}} \in \{0,1\}^{L \times K}$. The l^{th} row $z^{\mathbb{L}}_{l,\cdot}$ denotes the clusters contained in the l^{th} assignment set. In this way, we can decompose the assignment matrix $\mathbf{z}$ as $\mathbf{z} = \mathbf{z}^{SAC} \otimes \mathbf{z}^{\mathbb{L}}$, where $\mathbf{z}^{SAC} \in \{0,1\}^{N \times L}$ denotes the exclusive assignments of data items to assignment sets. $\mathbf{z}^{\mathbb{L}}$ then decomposes the assignment set to individual sources.

Given the above, the decomposition $\mathbf{x}^S = \mathbf{z} \otimes \mathbf{u}$, which is searched for in the structure inference problem (Definition 9), can be written as

$$\mathbf{x}^S = \left(\mathbf{z}^{SAC} \otimes \mathbf{z}^{\mathbb{L}}\right) \otimes \mathbf{u} = \mathbf{z}^{SAC} \otimes \left(\mathbf{z}^{\mathbb{L}} \otimes \mathbf{u}\right) . \tag{9.28}$$

This reformulation shows how, for a given centroid matrix $\mathbf{u}$, an equivalent single-assignment clustering (SAC) can be obtained. The assignment matrix

$\mathbf{z}$ is replaced by assignments to a single cluster, $\mathbf{z}^{SAC}$. The centroids of the SAC clusters are given by $\mathbf{u}^{SAC} := \mathbf{z}^{\mathbb{L}} \otimes \mathbf{u}$.

The difference between single-assignment and multi-assignment clustering thus lies in the inference phase. SAC ignores the high dependency between the centroids of different clusters and therefore has to estimate a much larger number of parameters than MAC. Given a finite number of data, we thus expect SAC to yield less accurate parameter estimators than MAC. Experiments described in Section 9.4.1 confirm this conjecture. Furthermore, the additional assumptions underlying MAC reduce the number of possible solutions in comparison to SAC, which we conjecture to further improve the estimation accuracy.

9.4 Experiments

In this section, we present experimental results on both synthetic and real-world data which allows us to compare the performance of MAC and of previously presented methods under different scenarios.

9.4.1 Experiments on Synthetic Data

In experiments with synthetic data we assess the performance of different clustering techniques under controlled conditions. To generate the data, we use the two centroid sets depicted in Figure 9.2. The structure in the data is generated as described in Section 2 and then perturbed to a variable degree by noise. We vary the noise fraction ϵ to obtain data sets with different complexities. For all experiments, ten different data sets are sampled, each with a different noise fraction. The reported performance results are averages over these ten runs.

We start with a comparison of the estimator accuracy of different inference techniques and then discuss the performance of the variants of the multi-assignment clustering method. Finally, we investigate the influence of particular data set properties on the accuracy of the parameter estimators.

Comparing Clustering Techniques

We run the four clustering techniques MAC, BICA, DBPS and INO on synthetic data generated from the overlapping sources depicted in Figure 9.2(b). The structure is perturbed by a mixture noise process with variable noise fraction ϵ and fixed noise parameter $r = 0.5$. We consider label sets up

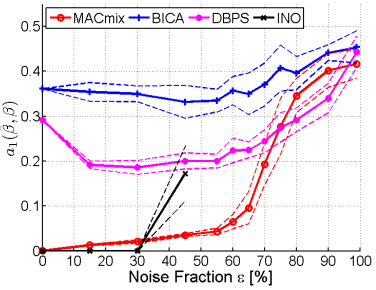
(a) Orthogonal Centroids (b) Overlapping Centroids

Figure 9.2: These three sources with 24 dimensions each are used to generate the synthetic data for the experiments. Black indicates a 1, white a 0 at the corresponding matrix element.

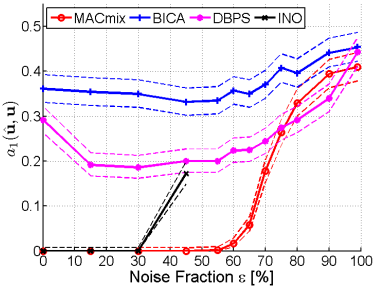
to degree 2, i.e. we have $\mathbb{L} = \{\{\}, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}\}$ and sample 50 data items per label set, yielding 350 data items in total. It is worth noting that we also allow the empty assignment set. The properties of data items which are assigned to no source are uniquely explained by the noise process.

Figures 9.3(a) and 9.3(b) show the parameter accuracy for the probabilistic parameter estimators $\hat{\boldsymbol{\beta}}$ and for the estimated binary parameters $\hat{\mathbf{u}}$, which we determine by rounding $\hat{\boldsymbol{\beta}}$. Note that only MAC uses probabilistic parameters, while the three other methods assume binary parameters. For noise levels above 45%, INO typically uses only one or two clusters to explain the data, identifying the rest of the data as noise. The accuracy of the parameter estimation or the stability of the clustering solutions can therefore not be measured for high noise fractions.

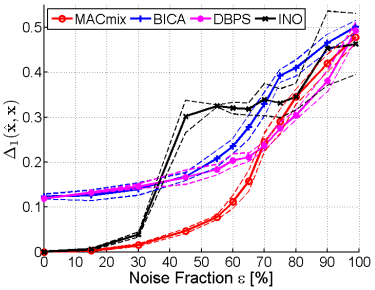
Concerning the estimators for the binary parameters, we observe that MAC perfectly retrieves $\mathbf{u}$ for noise fractions up to 55%. As the noise fraction further increases, the accuracy rapidly decreases and falls behind DBPS, but remains higher than the results obtained by BICA and INO. INO also yields perfect estimators for noise levels up to 30% but rapidly deteriorates for higher noise levels. Note that INO assumes a noisy-or noise process, i.e. $r = 1$, an assumption which is not fulfilled in this setting. An unmet assumption is also the reason why BICA yields poor estimators for all noise levels: This method assumes independent, i.e. orthogonal, sources, while the data is generated with overlapping data. Therefore, the restriction to orthogonal sources inhibits BICA from obtaining the correct centroids. Finally, the DBPS suffers from its greedy nature: The centroids are chosen from the candidate set such that the number of elements in the matrix $\mathbf{x}$ which can be explained with the new centroid is maximized. Namely for overlapping centroids, it can be favorable to choose a candidate



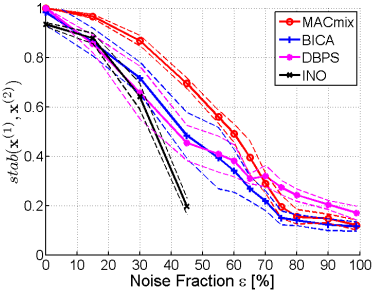
(a) Accuracy of Parameter Estimators



(b) Accuracy of rounded Parameter Estimators



(c) Reconstruction Error



(d) Stability of Clustering Solutions

Figure 9.3: Accuracy of parameter estimation, reconstruction error and stability for different inference techniques on synthetic data. The data is generated from overlapping sources with mixture noise with noise parameter $r = 0.5$. The noise fraction ϵ varies along the x-axis of the plots.

which corresponds to the proxy centroid of an assignment set, a phenomenon which we call *combination-singleton confusion*. These centroid estimators thus only poorly agree with the true centroids. Furthermore, the DBPS does not generatively model multiple assignments. For this reason, there is no strong incentive for overlapping sources, as the corresponding matrix elements are often already covered by a previously chosen centroid.

The accuracies of the probabilistic and the Boolean centroid estimators, $\hat{\boldsymbol{\beta}}$ and $\hat{\mathbf{u}}$, are depicted in Figure 9.3(a) and Figure 9.3(b), respectively. Comparing the two results shows that rounding is beneficial for MAC. The

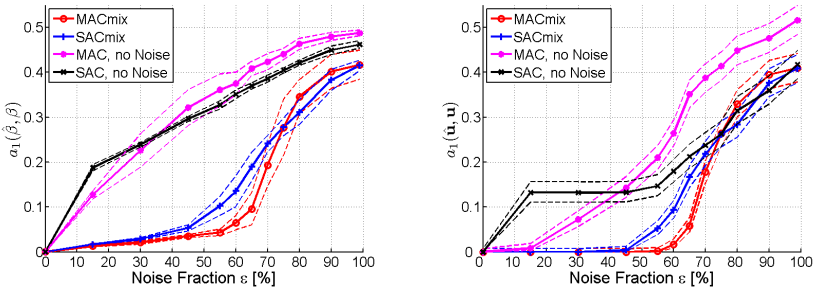
relatively small deviations in the unrounded centroids are due to the fact that the estimators make use of the relaxation from binary data in $\{0, 1\}$ to fractional values in the interval $[0, 1]$. While the differences between $\hat{\boldsymbol{\beta}}$ and the rounded centroids thus increases, the true values can be obtained by rounding. Only for noise values above 55%, the rounded estimators $\hat{\mathbf{u}}$ differ from the true binary centroids $\mathbf{u}$.

Regarding the error in the reconstruction of the signal part $\mathbf{x}^S$ (Figure 9.3(c)), we observe that both BICA and DBPS are able to partially compensate for the inaccurate centroid estimators. These two methods incur a significant reconstruction error even when noise-free data ($\epsilon = 0$) is available for parameter inference. As the noise level increases, the reconstruction error only slowly. INO obtains good reconstruction results for noise levels up to 30% but is limited by its inaccurate centroid estimators obtained for higher noise levels. MAC, on the other hand, benefits from the accurate estimators $\hat{\mathbf{u}}$ and clearly outperforms its competing methods for noise levels up to 65%.

Finally, for the instability of the cluster assignments (Figure 9.3(d)), we observe that MAC is able to find relatively stable cluster assignments and significantly outperforms its competitors for noise levels up to 50%. The decreasing stability even for low noise fractions explains why slightly erroneous reconstructions are obtained even if the parameters are perfectly retrieved. Reconstruction errors in these cases are due to cumulated noise effects on particular data items which case these data items to be assigned to a "wrong" assignment set. Furthermore, note that the stability does not decrease to zero in the case where the data is exclusively generated by noise. In this regime, we observe that BICA, DBPS and MAC assign most data items to a cluster that has only zeros in its structure and thus effectively explain the while data by noise. Only a small fraction of the data items is assigned to a cluster with non-empty centroid. Since this results in differing cluster sizes, the stability remains above zero.

Comparing MAC Variants

In this section, we investigate the influence of the noise model and compare the results obtained using multi-assignment clustering with the results of the corresponding single-assignment clustering. We analyze the accuracy of the estimators $\hat{\boldsymbol{\beta}}$ (Figure 9.4(a)) and of the rounded estimators $\hat{\mathbf{u}}$ (Figure 9.4(b)). With the noise model, both MAC and SAC are able to yield estimators which are, when rounded, fully correct for noise levels up to 45%.



(a) Accuracy of Estimators $\hat{\boldsymbol{\beta}}$ (b) Accuracy of Estimators $\hat{\mathbf{u}}$

Figure 9.4: Accuracy of original and rounded parameter estimators for multi-assignment and single-assignment clustering, both with and without noise model.

MAC is able to yield precise results even for higher noise levels, it breaks down at noise levels around 70%.

If no noise model is available, all matrix elements are to be explained by the structure. In this setting, both MAC and SAC perform dramatically worse than with noise model, and even for small noise levels, the unrounded estimators are imprecise. MAC is doing slightly better than SAC for noise levels up to 30% but then falls behind SAC. To our understanding, this is due to additional assumption on the structure in MAC: This technique interprets some data items as being generated by the disjunction of several sources. If noise is present but not modeled, this essential assumption is not fulfilled and leads to inaccurate estimators. Single-assignment clustering does not rely on this structure assumption and is therefore less affected by the unmodeled noise.

Comparison of Noise Model Variants. To investigate the performance of local noise models presented in Section 9.1.2, we generate data with a noise level ϵ linearly increasing from 0 to 2ϵ over the dimensions. The average noise fraction ϵ is varied between 0% and 50%. In this setting, an average noise ratio of 50% implies that the right-most columns are generated entirely by the noise process. The noise parameter r is set to 50% for all dimensions. Apart from this, the same experimental setting is used: 3 sources with at most two sources in each assignment set, and 350 objects.

We use the non-overlapping source centroids as depicted in Figure 9.2.

Inference on this data is carried out with the following variants of MAC:

MACmixNN: MAC without noise model

MACmixGG: MAC with mixture noise model and global noise model

MACmixDG: MAC with mixture noise model and dimension-wise noise intensity ϵ and global noise parameter r

MACmixDD: MAC with mixture noise model and dimension-wise noise intensity ϵ and noise parameter r

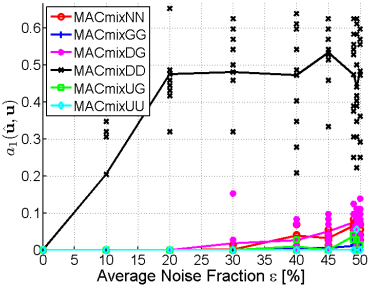
MACmixUG: MAC with mixture noise model, with object-wise noise intensity ϵ and global noise parameter r

MACmixUU: MAC with mixture noise model, with object-wise noise intensity ϵ and noise parameter r

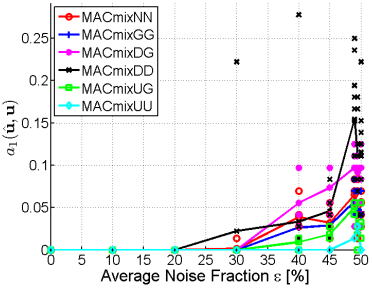
The estimators of the source parameters $\hat{\boldsymbol{\beta}}$ are depicted in Figure 9.5(a). Surprisingly, we see that MACmixDD yields by far the least accurate estimators and also has a high variation between the results obtained on several data set sampled from the same distribution. Figure 9.5(c) shows the estimated noise parameters $\hat{\epsilon}_d$, for $d = 1, \ldots, D = 24$ in the experiments where the true average noise fraction was 30%. Also this parameter is very inaccurate as soon as the true noise fraction per dimension exceeds roughly 20%. For dimensions with noise fraction above this critical value, we observe that the data is mainly explained by noise. The source accuracies are essentially random. Note that also the stability of the clustering solutions, depicted in Figure 9.5(e), rapidly decreases already for a small noise fraction of 10%.

All other MAC variants perform reasonably well: MACmixNN has by definition a noise intensity of $\epsilon = 0$. The global noise estimator obtained by MACmixGG determines the average noise intensity very accurately. Furthermore, it has a very small variance, as in the experiments presented above, e.g. in Figure 9.3(b) for this noise level. MACmixDG, the true model for these data, yields very accurate estimators for the noise intensity. The data-item-wise noise models MACmixUG and MACmixUU perform comparable to the global noise model MACmixGG.

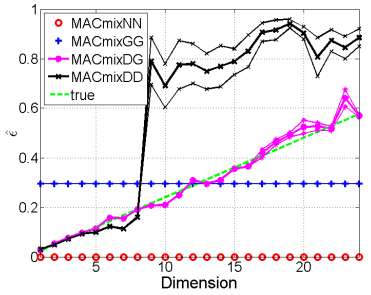
We interpret this unexpected poor performance of MACmixDD as follows: At high computational temperatures in the deterministic annealing scheme, the probabilistic source centroids $\hat{\boldsymbol{\beta}}$ take values strictly between 0 and 1. When determining the noise parameters, the values of $\hat{\boldsymbol{\beta}}$ are rounded.



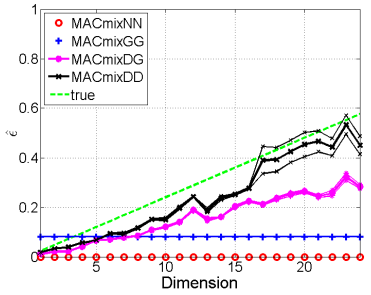
(a) Accuracy of $\hat{\mathbf{u}}$



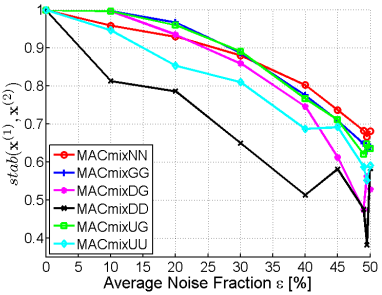
(b) Accuracy of $\hat{\mathbf{u}}$



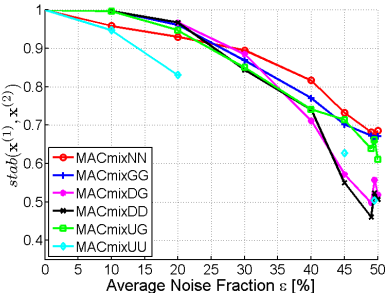
(c) Accuracy of $\hat{\epsilon}$



(d) Accuracy of $\hat{\epsilon}$



(e) Stability



(f) Stability

Figure 9.5: Parameter accuracy and stability for MAC variants with different noise models. The left column shows results obtained with the standard configuration where the intermediary values of $\hat{\beta}$ are rounded when estimating the parameters. In the right column, this rounding is omitted.

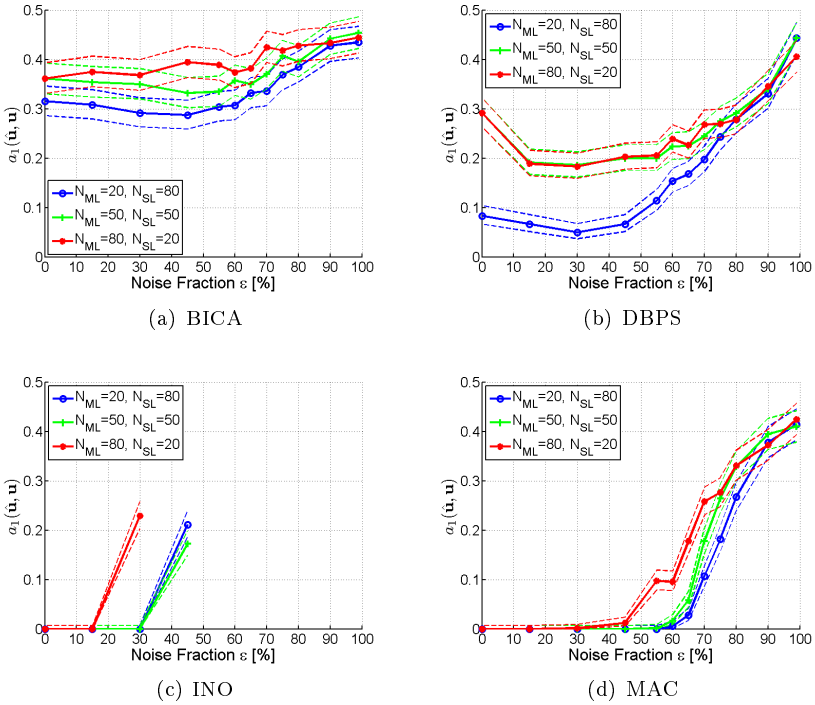
Doing so, the log-likelihood of explaining the observed noisy data at least partially by structure increases. The freedom to choose the noise parameter r dimension-wise enables the model to precisely adapt the estimated noise process to the observed data, and noise thus becomes the favorable explanation. MACmixDG does not have the freedom to choose dimension-wise noise parameters r. The noise values obtained by MACmixUG and MACmixUU have a high variance around the true value of the average noise fraction ϵ. However, since these variations are orthogonal to the true variation of the noise, they do not seem to disturb the inference of the centroids, as can be shown in Figure 9.5(a).

Rounding the intermediate estimators $\hat{\boldsymbol{\beta}}$ before estimating the noise fraction ϵ and the noise parameter r is a computational ruse that, in all cases studied so far, yields more accurate parameter estimators and speeds up computation. To determine the effect of this manipulation in the given setting, we run the same experiment without rounding $\hat{\boldsymbol{\beta}}$. The results are given in the right column of Figure 9.5. When the probabilistic parameter estimators are used, the accuracy of both the centroid and the noise fraction estimators obtained by MACmixDD drastically increases (Figures 9.5(b) and 9.5(d)). MACmixDG yields clearly less accurate noise fraction estimators than in the previous setting. Also the stability of the clustering solutions obtained by the data-item-wise noise models increases. However, MACmixUU now yields less stable results and does not use all possible assignment sets, which is why the stability can not be reported for average noise fractions above 20%.

When the average noise fraction increases, MACmixDD is the first one of the considered inference methods to incur errors on the estimated centroid parameters $\hat{\mathbf{u}}$, and also MACmixDG is outperformed by e.g. MACmixGG for higher noise fractions. We thus conclude that the local noise models carry with them the acute danger of over-parametrization. We recommend to compare results of the local noise models with the performance of a global noise model even in applications where there is evidence for a local noise model.

Influence of Data Set Properties

Fraction of Multi-Assignment Data. It seems reasonable to expect that the estimation of parameters is easier based on data generated by single sources than on data generated by multiple sources. To verify this intuition, we run BICA, DBPS, INO and MAC on three data sets with



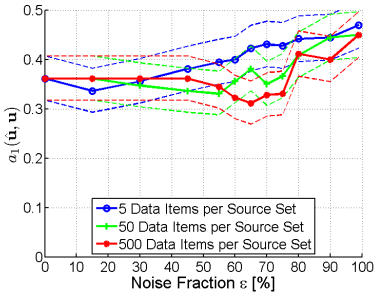
(a) BICA (b) DBPS

(c) INO (d) MAC

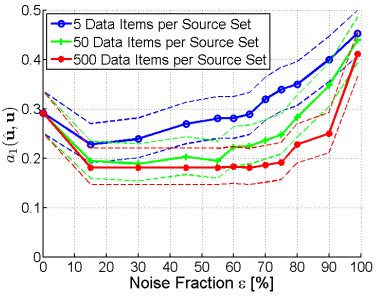
Figure 9.6: Accuracy of the rounded centroid estimators $\hat{\mathbf{u}}$ for low, medium and high ratio of data items from multiple sources.

varying numbers of data items per label degree. The first (second/third) data set, represented in blue (green/red), consists of 20 (50/80) samples per assignment set with more than one element, and of 80 (50/20) samples per single source.

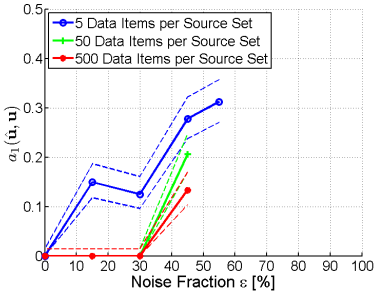
The results from this experiments are depicted in Figure 9.6 and confirm the intuition. Especially the DBPS (Figure 9.6(b)) is affected by the increased complexity as more data items are generated from multiple sources. The singlet-combination confusion is only a limited problem if the percentage of multi-assignment data is low, and error rates below 10% are obtained for noise levels up to 45%. For higher multi-assignment data ratios, however, at least 20% of the centroid bits are wrongly estimated at all noise levels.



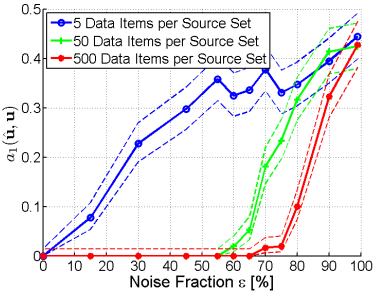
(a) BICA



(b) DBPS



(c) INO



(d) MAC

Figure 9.7: Accuracy of the rounded centroid estimators $\hat{\mathbf{u}}$ on data sets with 5, 50 and 500 data items per source set.

For INO as well as MAC, we observe that the noise level up to which fully accurate estimators can be obtained depends on the ratio of multi-assignment data. For both methods, this limiting noise ratio is higher if a lower ratio of data is generated by multiple sources. The largest difference is observed between the data with 20% and 50% multi-assignment data items. Further increasing the rate of such data items has a limited effect. For BICA, finally, the difference in estimation accuracy is less dramatic and results mainly in a loss of accuracy which is almost independent of the noise level.

Size of Data Set. As the size of the training data set increases, all inference methods studied in this thesis obtain, at least for some noise settings,

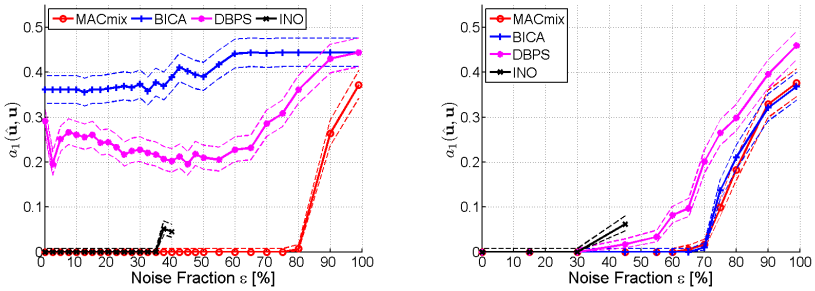
more accurate estimators. In Figure 9.7, we report the accuracy of the centroid estimator $\hat{\mathbf{u}}$ for three different data sets consisting of 5, 50 and 500 data items per assignment set. The overall data set thus consists of 35, 350 and 3500 data items, respectively.

For none of the methods, the accuracy on the noise-free data changes as more training data becomes available. However, for noisy data, some amelioration can be observed for both BICA and DBPS. For these two methods, the main cause for inaccurate parameter estimators seems to be the unmatched assumption of orthogonal centroids by BICA and the greedy optimization procedure followed by DBPS. Therefore, these methods can only slightly profit from the increased training data size.

A clear improvement is obtained by INO. As 50 or 500 data items are available per source set, this method yields perfect estimators for noise fractions up to 30%, while a high deviation is incurred already at the noise level of 15% when training on the small data set. For MAC, the estimators obtained on the small data set are very sensitive to noise. On larger training data sets, however, the Boolean centroids are perfectly recovered for noise fractions up to 55% (if $N = 350$) or even 65% (in the case where $N = 3500$). MAC is thus best able to profit from more training data.

Type of Noise Process. To investigate the influence of the noise process, we run experiments on data which is generated according to the noisy-or noise process (i.e. $r = 1$) and thus corresponds to the assumptions made by INO. The results, reported in Figure 9.8(a), show that INO now yields more precise parameter estimators than on data generated with $r = 0.5$ (Figure 9.3(b)). However, INO still yields no stable number of clusters for noise fractions above 40%. MAC also yields more accurate parameter estimators on noisy-or data: The parameters are perfectly retrieved for noise fractions up to 0.75, while the first deviations are already observed for $\epsilon = 0.6$ in the case of symmetric noise. Also note that the decay in performance is much sharper on noisy-or data. The performance of BICA and DBPS does not significantly change between the two noise models.

Source Geometry. The results on data generated from orthogonal centroids are depicted in Figure 9.8(b). This data set corresponds to the assumptions made by BICA, and this method can dramatically improve its estimator accuracy: The parameters are perfectly retrieved for noise fractions up to 65%. DBPS also profits from orthogonal centroids: due to the orthogonal centroids, the difference of data items generated by different



(a) Overlapping Centroids, noisy-or ($r = 1$)

(b) Orthogonal Centroids, symmetric noise ($r = 0.5$)

Figure 9.8: Dependence of the accuracy of the rounded centroid estimators $\hat{\mathbf{u}}$ on the source geometry and the noise process: The left panel shows experiments on data from overlapping centroids, in the right panel, the noise process is the noisy-or.

source sets is higher than in the setting of overlapping centroids. Therefore, combination-singleton confusion no longer occurs, and the parameter estimators are perfect for noise fractions up to 0.3 and then only slowly deteriorate. INO, again, chooses a varying number of clusters for noise fractions above 0.45. Summing up, we find that all considered inference methods improve their estimation accuracy when data is generated by orthogonal centroids.

9.4.2 Experiments on Real-World Data

To evaluate the performance of our algorithm on real data, we apply multi-assignment clustering to role-mining. We use a real-world dataset containing the user-permission assignment matrix of $N = 4900$ users and $D = 1300$ permissions. A section of this data matrix is depicted in Figure 9.9. The roles are inferred based on the permissions of the first 2000 users. The permissions of the remaining users are used to compute the generalization ability of the role set.

In order to evaluate the different methods on more complex data with higher noise level, we generate a modified data set $\bar{\mathbf{x}}$ as follows: The first and the second 500 permissions of the original matrix are combined by an

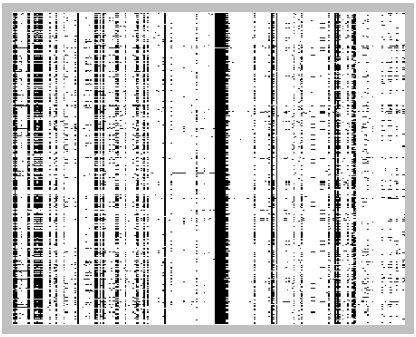


Figure 9.9: An excerpt of the real-data matrix. a black dot indicates a 1 at the corresponding matrix element, white indicates a 0. The full matrix has size 4900×1300.

element-wise OR operation to give the structure part of $\bar{\mathbf{x}}$, $\bar{\mathbf{x}}^S$:

$$\bar{\mathbf{x}}^S = \left[\bar{x}^S_{n,d}\right]_{2000\times 500} \qquad \text{with } \bar{x}^S_{n,d} = x_{n,d} \vee x_{n,d+500} \tag{9.29}$$

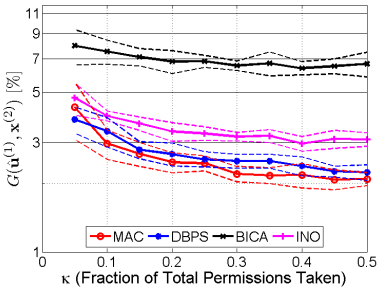
Furthermore, 33% of the entries of the matrix $\bar{\mathbf{x}}^S$ are replaced by random bits to yield the modified matrix $\bar{\mathbf{x}}$, which exhibits both a clearly higher structural complexity and a considerably increased noise level.

For the generalization experiment, we use again the permissions of unused users in the structure matrix $\bar{\mathbf{x}}^S$. Doing so, we are able to detect whether a method is able to infer a suitable role set even under conditions with high noise level.

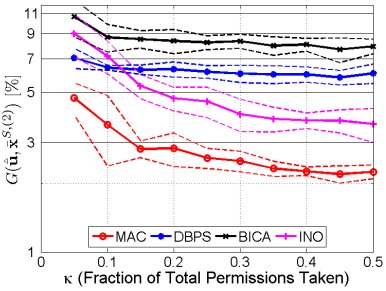
Results of Different Inference Techniques

The results of the generalization experiments are depicted in Figure 9.10 for the four methods MAC, DBPS, BICA and INO. The ratio κ of disclosed permissions is varied between 0.05 and 0.5. All models profit from an increased number of disclosed permissions.

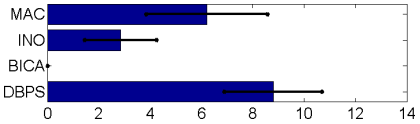
On the original dataset (Figure 9.10(a)), DBPS and MAC perform comparably for lower values of κ. As κ is increased, MAC is able to outperform DBPS, which we see as an indication for the more accurate parameter estimators obtained by the proposed method. INO also performs well for low κ, but is not able to improve as much as the other two methods when more



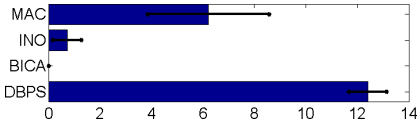
(a) Generalization Ability on Original Data



(b) Generalization Ability on Modified Data



(c) Average Role Overlap (%) on Original Data



(d) Average Role Overlap (%) on Modified Data

Figure 9.10: Results from the generalization experiment on real data. 30 roles are used in MAC, DBPS and BICA. INO selects 34 roles on the original data set (left column) and 25 on the modified data set (right column). The figures in the upper row show the generalization error obtained with the inferred roles, the average row overlap is displayed in the lower row.

dimensions are revealed. We assume that this is due to an unappropriate noise model. The noise parameters obtained by MAC with mixture noise model are $\epsilon \approx 6\%$ and $r \approx 20\%$, which is clearly different from a noisy-or noise process (which would correspond to $r = 1$). The performance of BICA is significantly behind the results obtained with the three other methods for all values of κ. As the average centroid overlap of the roles inferred by MAC is $6 - 7\%$, the assumption of independent, i.e. non-overlapping, centroids made by BICA seems therefore appropriate and causes the high generalization error. Note that roughly 13% of the matrix entries are 1. The most trivial role set containing one single role with no permission would thus yield a generalization error of 13%.

In the experiments on the modified dataset with more structure and higher noise level (Figure 9.10(b)), all methods incur higher prediction er-

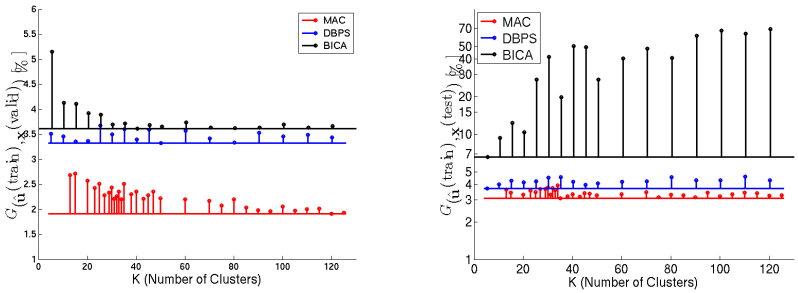
rors. MAC yields significantly lower prediction errors than all its competitors for all values of κ above 0.05. In comparison with the original dataset, DBPS looses more generalization ability than all other methods and is considerably behind MAC in the more difficult setting. Also INO is clearly behind MAC but still outperforms BICA and DBPS when a medium to large rate of permissions is revealed.

Furthermore, it is enlightening to follow the trend in the performance of different methods as more and more dimensions are revealed: While MAC and INO profit from the additional information and yield more precise predictions, the performance of both DBPS and BICA shows only minor changes. This indicates that these two methods have problems to infer the underlying structure, which would allow to predict the permissions of new users. As observed in the experiments on synthetic data, BICA suffers from the unmet assumption of orthogonal centroids, while for DBPS, the problem is its greedy nature, which yields combination-singleton confusion. Since these two methods do not have a noise model, the effect of the model mismatch is aggravated when the noise level increases.

The lower row of Figure 9.10 shows the average role overlap between the roles obtained with the different methods. This is the average number of permissions that the inferred roles have in common. The overlap between the roles obtained by MAC does not change as the complexity and the noise level of the data increases. By construction of the method, the average role overlap obtained by BICA is 0. INO yields roles which are almost orthogonal (overlap less than 1%, with 21 roles) for the more complicated case, while an overlap of some 3% and 34 roles is observed when doing inference on the original data set. DBPS, on the other hand, shows an opposite trend: The roles share a higher number of permissions as the structure get more fine-grained and the noise level increases. The construction of the more complex data set leads to a higher number of permissions in the data set and thus a higher overlap between the permissions of different users. Since DBPS derives the candidate roles from individual data items and their intersections, all candidate roles show an higher overlap.

Optimal Parameter Values

In the experiments reported in the previous section, we choose the number of roles by educated guess, and the other parameters of the models are set to their default values. INO uses non-parametric techniques to avoid a hard choice of the number of clusters, but for the other methods considered in



(a) Validation Data (b) Test Data

Figure 9.11: The generalization error (with $\kappa = 0.1$) for varying numbers of clusters K. The number of clusters is displayed, the other parameters are determined by exhaustive search (for MAC) over the parameter space of a discretisation thereof. The performance on the cross-validation test sets is shown on the left, the results on the hold-out set on the right. Note that namely BICA performs clearly worse on the test data than on the validation data, i.e. the roles obtained by BICA are not able to explain new data.

these experiments, namely the number of sources is a critical parameter. We therefore investigate the dependency of the clustering performance on the model order in the context of role mining. The experiments are based on a data set with similar statistics as the one used in the previous experiments. Roughly 1900 users are hold out as test set. The remaining 3000 users are randomly split into a training and a validation set. This splitting is repeated five times such that each user is once in the validation set and four times in the training set. The number of permissions used in this experiment is 500. In all experiments, the fraction κ of revealed permissions is $\kappa = 0.1$, with 10 random subsets of permissions revealed.

The number of roles is varied between 5 and 120. In order to determine the optimal values of the remaining parameters (such as the maximal degree of the assignments for MAC) for a given number of roles, the methods run on a training set for each of the possible values of these parameters if the number of possible parameter values is small. For continuous parameters, we discretize the parameter space into roughly 50 equally spaced parameter values spanning the whole range of possible parameter values. The prediction performance on a separate validation set is evaluated and the parameter

values for a given number of roles are set such that the average prediction performance (averaged over the five splits in training and validation data) on the evaluation set is maximized. The performance of the three methods MAC, DBPS and BICA on the validation data is depicted in Figure 9.11(a).

The prediction ability of the role set inferred by a method with a fixed number of classes is estimated on the separate test data set. The performance values are displayed in Figure 9.11(b). While both MAC and DBPS have a prediction error which is about 1% above the value obtained on the validation set, the performance of BICA dramatically deteriorates on the test data. Recall that BICA is the only method assuming orthogonal centroids. In the experiments with synthetic data, we have observed that its performance largely depends on this assumption being true. Furthermore, all three competing methods support role overlap and do find overlapping roles. These two observations are strong evidence that the actual underlying structure contains roles with significant overlap of permissions.

Chapter 10

Approximation Set Coding for Cluster Validation

In the experiments presented in Chapter 9, we compare different clustering models based on their performance with respect to a particular measure. Furthermore, except for the INO, we determined the number of clusters K either by knowing the true generative process (in the experiments with synthetic data) or by an educated guess based on the data matrix. However, selecting an appropriate model and the right model order, i.e. the right number of clusters, are fundamental issues in clustering problems [23]. Namely from a generative viewpoint, where selecting a model implies also selecting an explanation of the observed data, model selection and model order selection are fundamental tasks of scientific inquiry.

Several approaches to guide the selection of a clustering technique out of the large variety of models and algorithms have been proposed. Most of these approaches are based on criteria describing subjective ideas on the property of a "good" clustering solution. For example, the Gap statistic [106] prefers compact clusters around a centroid, while both the Akaike Information Criterion (AIC) [2] and the Bayesian Information Criterion (BIC) [99] assume that model errors are normally distributed. These model selection techniques are only reliable if the respective assumptions are fulfilled and should not be applied in scenarios where elongated structures or non-Gaussian errors can not be excluded.

Statistical learning theory [114, 116] advocates the generalization ability of models to measure the quality of a model. The measure of stability (Sec-

tion 2.4.3) has shown promising results for model order selection in practice [9, 40, 75]. However, its reliability is disputed, as in the limit of infinite data, the stability only depends on the objective function of the clustering methods, but not on its parameters [8]. More fundamentally, stability is only one aspect of an ideal clustering solution. The other criteria is the amount of information retrieved from the data. A small decrease in the stability might be compensated by a large increase in the information content of the clustering solution. The goal of statistical modeling is thus a trade-off between stability and informativeness. In the following, we briefly summarize an information-theoretic approach to weight up these two considerations [20] and detail the adaptation of the calculations to our setting. We then show that this framework captures the accuracy as well as the generalization ability of the inferred parameters.

10.1 Clustering As Communication

The idea underlying the information-theoretic model validation scheme is as follows: The solutions for an inference problem should generalize to a new data set. However, data sets differ from each other due to perturbations. For this reason, the single best solution is replaced by an approximation set, which contains a number of "good" solutions. The size of the approximation set is chosen such that the same clustering solutions are obtained on both data sets.

Formally, a clustering is a function c which assigns each data item $X_{n,\cdot}$, $n = 1, \ldots, N$, in the data set $\mathbf{X}$ to a set of clusters, i.e. $c(X_{n,\cdot}) = \mathcal{L}_n \in \mathbb{L}$. The hypothesis space $\mathcal{C}(\mathbf{X})$ contains all possible clustering solutions on $\mathbf{X}$. We assume a *risk function* $R(c, \mathbf{X})$ to measure how well a particular clustering c groups the objects in $\mathbf{X}$. Apart from the assignments of objects to sets of clusters, the clustering costs typically depend on parameters $\boldsymbol{\theta}$ describing e.g. the centroids of a cluster. To simplify the notation, these parameters are not explicitly listed as arguments of the clustering cost function R.

Given a data set $\mathbf{X}$, the optimal clustering solution $c^{\perp}(\mathbf{X})$ is computed such that the *empirical risk* on the data set $\mathbf{X}$ is minimized:

$$c^{\perp}(\mathbf{X}) := \arg \min_{c \in \mathcal{C}(\mathbf{X})} R(c, \mathbf{X}) \tag{10.1}$$

Assuming that $\mathbf{X} \sim P(\mathbf{X})$ is a random variable generated by a stochastic process, the empirical risk minimizer $c^{\perp}(\mathbf{X})$ is a random variable as well.

The optimal solution on the training data might therefore not be suitable for describing the clustering of a data set $\mathbf{X}$. Instead, we define the set $\mathcal{C}_\gamma(\mathbf{X})$ of empirical risk approximations:

$$\mathcal{C}_\gamma(\mathbf{X}) := \{c(\mathbf{X}) : R(c, \mathbf{X}) \leq R(c^\perp, \mathbf{X}) + \gamma\} \tag{10.2}$$

The set $\mathcal{C}_\gamma(\mathbf{X})$ describes the set of objects which are statistically indistinguishable with respect to their relevant properties. This set corresponds to the micro-canonical ensemble in statistical physics [82, 83]. In our setting, all solutions share the property of being γ-close to $c^\perp$ in terms of their costs.

Replacing the exact solutions with the approximation sets coarsens the hypothesis space. The key question of learning remains to determine the optimal resolution in the hypothesis space: The parameter γ has to be chosen such that the approximation sets $\mathcal{C}_\gamma(\mathbf{X})$ are still identifiable under the random variations of the data. Conversely, choosing γ too high yields a too coarse-grained resolution and does not capture the optimal amount of information contained in the data.

Approximation Set Coding and Approximation Capacity. To describe the identification of approximation sets formally and to derive an approximation capacity, we embed the problem of clustering in a communication framework. In this setting, the approximation sets represent the code words, and the problem instances $\mathbf{X}$ form the noisy channel. The approximation capacity of this "channel" ranks models according to their stability and informativeness. Good models are those which have a high capacity.

The communication process is organized in two stages:

Protocol Design The problem generator $\mathfrak{PG}$ generates a first data set $\mathbf{X}^{(1)} \sim P(\mathbf{X})$ and sends it to the sender $\mathfrak{S}$. $\mathfrak{S}$ then transforms the data by a set of permutations $\Sigma^{\mathfrak{S}} := \{\sigma_j, 1 \leq j \leq 2^{n\rho}\}$. Thereby, $\mathfrak{S}$ generates $2^{n\rho}$ data sets $\{\sigma_j \circ \mathbf{X}^{(1)}\big|_{j=1}^{2^{n\rho}}\}$ and consequently, $2^{n\rho}$ optimization problems with $2^{n\rho}$ approximation sets as solutions. These permutations $\Sigma^{\mathfrak{S}}$ are shared with the receiver $\mathfrak{R}$ and serve as codebook.

Communication During communication, the sender $\mathfrak{S}$ randomly selects a permutation $\sigma_s \in \Sigma^{\mathfrak{S}}$. This permutation is the message to be communicated. The problem generator $\mathfrak{PG}$ generates a new data set $\mathbf{X}^{(2)} \sim P(\mathbf{X})$ from the same distribution as $\mathbf{X}^{(1)}$. $\mathfrak{PG}$ obtains the

permutation σ_s and applies it to $\mathbf{X}^{(2)}$. $\mathfrak{PG}$ sends the resulting data $\tilde{\mathbf{X}}^{(2)} := \sigma_s \circ \mathbf{X}^{(2)}$ to the receiver $\mathfrak{R}$.

On the receiver side, the task is to determine the permutation σ_s. The lack of knowledge about the permutation σ_s is mixed with the stochastic variability in the source generating the data sets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. To estimate the permutation $\hat{\sigma}$, $\mathfrak{R}$ determines the intersections between $\mathcal{C}_\gamma(\tilde{\mathbf{X}})$ and the approximation sets in the codebook of approximation problems:

$$\Delta\mathcal{C}_{\gamma,s} = \mathcal{C}_\gamma(\sigma_s \circ \mathbf{X}^{(1)}) \cap \mathcal{C}_\gamma(\tilde{\mathbf{X}}^{(2)}),\ \sigma_s \in \Sigma\ . \tag{10.3}$$

The estimator for the permutation $\hat{\sigma}$ is determined such that the intersection is maximized:

$$\hat{\sigma} := \arg\max_{\sigma\in\Sigma} |\Delta\mathcal{C}_{\gamma,s}|\ . \tag{10.4}$$

An error occurs whenever the estimated permutation $\hat{\sigma}$ is different from the permutation σ_s chosen by the sender. Analyzing the error probability $\mathbb{P}(\hat{\sigma} \neq \sigma_s|\sigma_s)$ of this communication protocol [20] shows that an asymptotically non-vanishing error rate is achievable for rates

$$\begin{aligned}\rho \leq \mathcal{I}_\gamma(\sigma_s,\hat{\sigma}) &= \frac{1}{n}\left(H(\sigma_s) + \log\left(\frac{|\Delta\mathcal{C}_{\gamma,s}|}{|\mathcal{C}_\gamma^{(1)}||\mathcal{C}_\gamma^{(2)}|}\right)\right)\\ &\overset{(a)}{=} \frac{1}{n}\log\left(|\Sigma^{\mathfrak{S}}|\cdot\frac{|\Delta\mathcal{C}_{\gamma,s}|}{|\mathcal{C}_\gamma^{(1)}||\mathcal{C}_\gamma^{(2)}|}\right)\ ,\end{aligned} \tag{10.5}$$

where $H(\sigma_s)$ denotes the entropy of the random permutation chosen by the sender. The transformation (a) assumes that the sender chooses this permutation uniformly from all members of the permutation set $\Sigma^{\mathfrak{S}}$, in this case we have $H(\sigma_s) = \log|\Sigma^{\mathfrak{S}}|$. Furthermore, to obtain a compact formulation, we have introduced $\mathcal{C}_\gamma^{(q)} := \mathcal{C}_\gamma(\mathbf{X}^{(q)})$ for $q = 1, 2$.

Note that the mutual information $\mathcal{I}_\gamma(\sigma_s,\hat{\sigma})$ (10.5) is not defined when the intersection set $\Delta\mathcal{C}_s$ between the two approximation sets is empty.

10.2 Calculating the Approximation Capacity

To evaluate the mutual information, we calculate the sizes of approximation sets. Assuming that the approximation sets are large, we use the canonical state sum as an approximation of the micro-canonical state sum [83]:

$$\forall\gamma, \exists\beta \text{ s.t. } |\mathcal{C}_\gamma(\mathbf{X}^{(q)})| = \sum_{c\in\mathcal{C}(\mathbf{X}^{(q)})} \exp(-\beta R(c,\mathbf{X}^{(q)})). \tag{10.6}$$

The weights $\exp(-\beta R(c, \mathbf{X}^{(q)}))$ are known as Boltzmann factors. Similarly, to approximate the cardinality of the joint approximation set $\Delta\mathcal{C}$, we use

$$\forall\gamma, \exists\beta \text{ s.t. } |\Delta\mathcal{C}_{\gamma,s}| = \sum_{c\in\mathcal{C}(\mathbf{X}^{(1,2)})} \exp(-\beta R(c, \mathbf{X}^{(1)})) \cdot \exp(-\beta R(c, \mathbf{X}^{(2)})) \ , \quad (10.7)$$

with $\mathbf{X}^{(1,2)} = (\mathbf{X}^{(1)}, \mathbf{X}^{(2)})$. It should be stressed that under these approximations, the calculation of $\mathcal{I}_\beta(\sigma_j, \hat{\sigma})$ requires only to calculate Boltzmann factors $\exp(-\beta R(c, \mathbf{X}^{(2)}))$. In the following, we will always use the partition sums to determine the size of the approximation sets. Doing so, we replace the set sizes in Eq. 10.5 by the state sums (Equations 10.6 and 10.7), and choose the parameter β instead of the micro-canonical parameter γ such that the approximation capacity ρ is maximized.

Keep in mind that not for all β in Eq. 10.6 there exists a value of γ such that the partition sum is equal to the size of the approximation set. In particular for large values of β, the sum of Boltzmann factors may be smaller than 1. In this pathological setting, the error of the approximation used to compute the mutual information is very high, and no conclusion can be drawn about the value of $\mathcal{I}_\beta(\sigma_j, \hat{\sigma})$. However, since we are primarily interested in the maximum of this information, around which the approximation sets necessarily have appropriate sizes, these limits are no handicap for the proposed approach to model and model order selection.

Simplifications in a Factorial Model. In a general setting, computing the state sum is still a very demanding task, which is mostly addressed using further approximations or sampling techniques. A drastic simplification is possible if the individual data items are mutually independent. In such a factorial model, the risk of a clustering solution is the sum of the risks of assigning a data item to a particular cluster:

$$R(c, \mathbf{X}^{(q)}) = \sum_{n=1}^{N} R(c_n, X_n^{(q)})$$

and the size of the approximation set is determined according to

$$|\mathcal{C}_\gamma^{(q)}| = \prod_{n=1}^{N} \sum_{c_n\in\mathcal{C}(\mathbf{X}_n^{(q)})} \exp\left(-\beta R(c_n, X_n^{(q)})\right) \ . \quad (10.8)$$

To simplify the state sum corresponding to the joint approximation set (Eq. 10.7) in this way, we have to ensure that the product of Boltzmann

factors goes over corresponding objects. To do so, we choose a permutation on the data items of the second data set $\mathbf{X}^{(2)}$ such that the overall ℓ_1 distance between data items in $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ is minimized:

$$\pi^{(2)} := \arg\min_{\pi \in P_N} \left\{ \sum_{n=1}^{N} \left| X^{(1)}_{n,\cdot} - X^{(2)}_{\pi(n),\cdot} \right| \right\} , \tag{10.9}$$

where P_N is the set of all permutations on N objects. Using the Hungarian algorithm [74], the permutation $\pi^{(2)}$ can be found in $\mathcal{O}\left(N^3\right)$ running time. With this permutation, the size of the joint approximation set is approximated as

$$|\Delta\mathcal{C}_{\gamma,s}| \approx \prod_{n=1}^{N} \sum_{c_n \in \mathcal{C}(\mathbf{X}^{(1)}_n)} \exp\left(-\beta R(c_n, \mathbf{X}^{(1)}_n)\right) \cdot \exp\left(-\beta R(c_n, \mathbf{X}^{(2)}_{\pi^{(2)}(n)})\right) .$$

The mentioned simplifications are applicable to sets of vectorial data where the solution space of the two data sets $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ are identical. For non-vectorial data (e.g. distance data), more involved mappings are required.

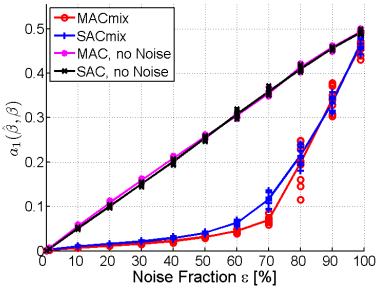
10.3 Experimental Evaluation

In this section, we present experiments indicating that the ranking of clustering methods according to the approximation capacity corresponds to the ranking according to specialized quality measures.

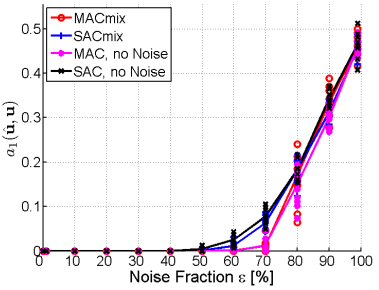
In the experiments, we restrict ourselves to single- and multi-assignment clustering of Boolean data, both with and without a noise model. For these models, we have a clearly defined risk function given by Eq. 9.21, while the risk function is not clearly given for the other clustering methods considered in Chapter 9. In the case of multi-assignment clustering, the cluster solution c_n for data item n might imply that n is assigned to several clusters. Using the assignment set $\mathcal{L}$ to describe this assignment, the risk $R(c, X_n)$ corresponds to $R^{mix}_{n,\mathcal{L}}$ as defined in Eq. 9.21.

10.3.1 Experiments on Synthetic Data

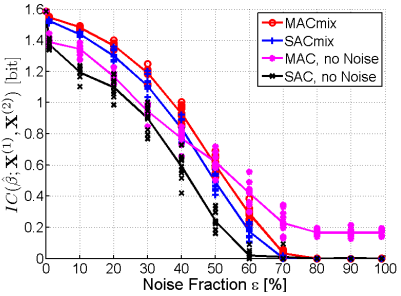
For the first experiment, we generate data from the first two overlapping sources depicted in Figure 9.2(b) and add a padding of 30 additional dimensions containing only zeros. The set of possible source sets is $\mathbb{L} =$



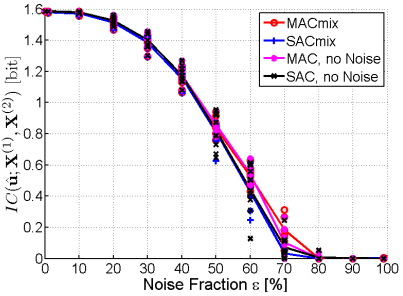
(a) Accuracy of $\hat{\boldsymbol{\beta}}$



(b) Accuracy of $\hat{\mathbf{u}}$



(c) Approximation Capacity computed based on $\hat{\boldsymbol{\beta}}$



(d) Approximation Capacity computed based on $\hat{\mathbf{u}}$

Figure 10.1: Experiments on data sets generated by two slightly overlapping sources. The parameter accuracy and the approximation capacity are shown both for the original estimators $\hat{\boldsymbol{\beta}}$ as well as for the rounded estimators $\hat{\mathbf{u}}$.

$\{\{1\}, \{2\}, \{1, 2\}\}$, and 100 data items are sampled from each source set. The noise process is symmetric, i.e. $r = 0.5$. Inference is carried out with both SAC and MAC, each with and without a noise model.

Figure 10.1(a) shows the accuracy of the probabilistic centroids $\hat{\boldsymbol{\beta}}$. As observed in previous experiments, the ℓ_1 distance between the true and the estimated parameters grows almost linearly for both the MAC and the SAC model without a noise model, while the two models with a noise component incur only slight errors for noise fractions up to 60%. If the probabilistic centroids are used to compute the approximation capacity, the inaccuracy in the parameter estimators directly translates into a smaller capacity (Fig-

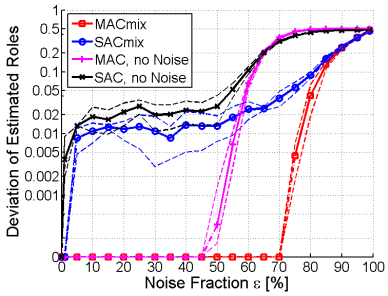
ure 10.1(c)).

Figure 10.1(b) shows the accuracy of the rounded estimators $\hat{\mathbf{u}}$. The rounding corrects most of the deviations in $\hat{\boldsymbol{\beta}}$ for noise fractions up to 60%. Due to the symmetric noise process, the effect of the noise on $\hat{\boldsymbol{\beta}}$ leads to a deviation from the true value that is, on the average, independent of the true value: Instead of 0 (1), the estimator $\hat{\beta}_{k,d}$ is $\hat{\beta}_{k,d} = 0+\eta$ ($\hat{\beta}_{k,d} = 1-\eta$). Even for relatively large values of η, rounding the estimator $\hat{\beta}_{k,d}$ returns the true value 0 (1) for $\hat{u}_{k,d}$. As a consequence, the approximation capacity obtained by all four inference techniques is comparable. Only for noise fractions between 0.6 and 0.7, the more accurate parameter estimators due to a noise model translate into a higher approximation capacity.

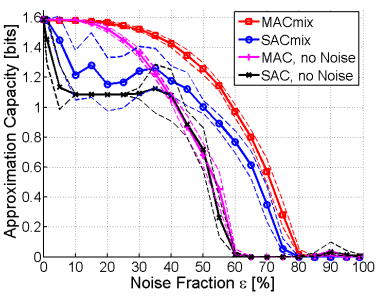
Considering Figure 10.1(d), we observe that the approximation capacity attains its maximum of $\log_2 3 \approx 1.6$ bits for noise-free data. As the noise fraction increases, the capacity first only slightly diminishes but then rapidly decreases for noise fractions above 20%. In the setting where $\epsilon = 0.5$, the capacity has fallen to half of the maximum, even though the parameter estimators $\hat{\mathbf{u}}$ are still perfect. The noise fraction of 50% implies that half of the matrix entries are random and the number of distinguishable permutations $|\Sigma^{\mathfrak{S}}|$ in Eq. 10.5 reduces to the square root of its value in the noise-free case. Accordingly, the approximation capacity decays in spite of the perfect parameter estimation.

To increase the differences between the four considered variants of MAC, we design an experimental setup where symmetries are broken: We generate data from two twenty-dimensional orthogonal centroids with unequal numbers of ones. Further dimensions are added to all emissions up to the total number of D dimensions. The padding dimensions contain as many values 0 as 1, the Hamming distance between the centroids thus remains constant. Furthermore, we consider an asymmetric noise process with noise parameter $r = 0.75$ such that the effects of noise do not cancel out in the average. The data items are mainly generated by single-label sources, more precisely we have $N_{\{1\}} = 150$, $N_{\{2\}} = 140$ and $N_{\{1,2\}} = 10$.

The results obtained in this setting with $D = 150$ are given in Figure 10.2. The parameter estimators of the models without noise component are clearly less accurate than the estimators obtained by MACmix and SACmix. In this setting, the effect of the noisy data onto the estimators is no longer symmetric but introduces a tendency towards 1. As a consequence, and in contrast to the previous setting, rounding is no longer as beneficial, and the rounded estimators are less precise. For the noisy



(a) Accuracy of $\hat{\mathbf{u}}$



(b) Approximation Capacity computed based on $\hat{\mathbf{u}}$

Figure 10.2: Estimator accuracy and approximation capacity on data generated from two asymmetric orthogonal sources. A total number of 300 data items is drawn mainly from single sources, the noise process is asymmetric with $r = 0.75$.

SAC model, the main problem is the small number of data items from the combination $\{1, 2\}$ that leads to erroneous estimators for this source.

The differences in the parameter accuracy are mirrored in the approximation capacity. For both the noise-free and the noisy SAC model, the approximation capacity clearly drops as the noise fraction becomes positive. Note that these two models incur small, but non-zero deviations in their parameter estimators. MAC without noise model first maintains a high approximation capacity, but then falls back to similar values as the noise-free SAC model as the parameter estimators of the two models become similar in accuracy. MAC with the mixture noise model maintains the highest approximation capacity, in accordance with the highest precision in the parameter estimators.

The high sensitivity of the approximation capacity to small deviations in the parameter estimators can be explained with the logarithmic form of this quality measure. Consider for example the binary symmetric channel (BSC) [29], where bits are flipped with probability p. The information capacity of this channel is given by $C(p) = 1 - h(p)$, where $h(p)$ denotes the binary entropy function, given by $h(p) = -p \log p - (1 - p) \log(1 - p)$. As seen in Figure 10.3, a minor variation of a small value of p leads to a large change in the information capacity. If p is around 0.5, the information capacity reaches its minimum and shows little sensitivity for variations of p.

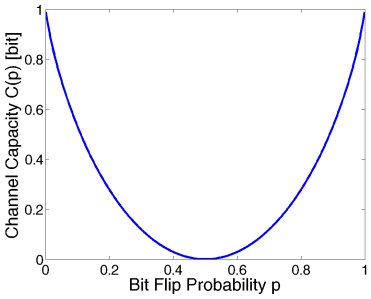


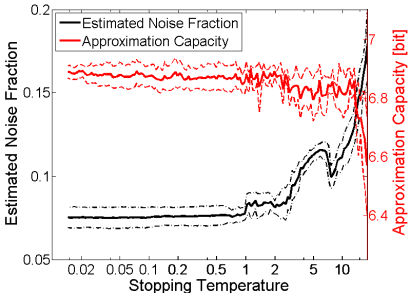
Figure 10.3: Capacity of the binary symmetric channel (BSC) as a function of the bit flip probability p.

10.3.2 Experiments on Real-World Data

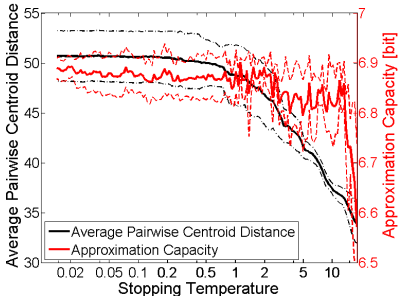
We conduct experiments on real-world data to determine the agreement of the approximation capacity with several specific quality measures for multi-assignment clustering. In these experiments, we trace the value of the mentioned measures as a function of the computational temperature T as it is gradually decreased in the deterministic annealing scheme. Note that namely around phase transitions, the value of estimators and thus also the value of the quality measures can rapidly (discontinuously) change as the computational temperature is reduced. This phenomenon implies fluctuations in all quality measures. Due to the high sensitivity of the information capacity, the fluctuations are particularly pronounced for this measure.

For these experiments we use the same data set as in Section 9.4.2. We randomly split the data set five times into training and test sets and report the averages and the standard deviations of the quality measures over these splits.

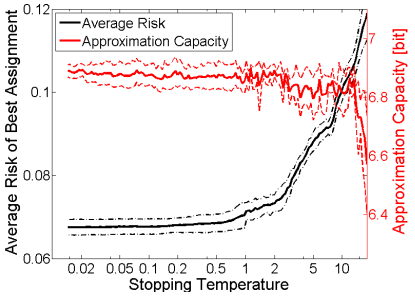
The value of the estimated noise fraction $\hat{\epsilon}$ and the average pairwise distance between centroids in $\hat{\mathbf{u}}$ are depicted in Figures 10.4(a) and 10.4(b) along with the approximation capacity. At high T, the estimated noise fraction is high — remember that in the data set we used, roughly 13% of the matrix elements are 1, all others are 0. The distance between different centroids is relatively low, indicating that the data is mostly explained as noise. Correspondingly, the approximation capacity is low. As the computational temperature decreases, the centroids become more and more dissimilar and



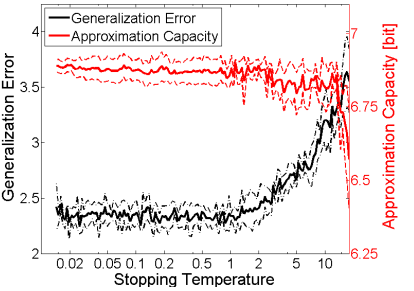
(a) Approximation capacity and Estimated Noise Fraction $\hat{\epsilon}$



(b) Approximation Capacity and Average Pairwise Centroid Distance in $\hat{\mathbf{u}}$



(c) Approximation Capacity and Average Risk of Optimal Assignment



(d) Approximation Capacity and Generalization Error

Figure 10.4: The development of the approximation capacity in comparison with several specific measures that characterize the evolution of the parameter estimators (upper row) and the clustering performance (lower row), as a function of the computational temperature T.

the estimated noise fraction decreases. At $T < 1$, the data is mostly explained by the structure, and the approximation capacity reaches its maximum. As the temperature further decreases, the estimators maintain their value, and also the approximation capacity stays at the same value.

With the responsibilities $\gamma_{n,\mathcal{L}}^{mix}$ given in Eq. 9.20, the average risk of the best assignment over all data items is computed as

$$\hat{\overline{R}} := \frac{1}{N}\sum_{n=1}^{N} R_{n,\hat{\mathcal{L}}_n} \qquad \text{with } \hat{\mathcal{L}}_n = \arg\max_{\mathcal{L}\in\mathbb{L}} \gamma_{n,\mathcal{L}}^{mix} \ .$$

The average risk measures the match between the data and the model and decreases as the model parameter become more adapted to the data (Figure 10.4(c)). Finally, for the generalization error in Figure 10.4(d), we observe the same behavior as for the risk on the training data. This is again an indication that the proposed multi-assignment clustering model is robust against overfitting.

The framework of approximation set coding offers a theoretically well-founded approach to model selection and model order selection. In our experiments on both synthetic and real-world data, we have observed that the generic criterion of approximation capacity agrees with specific criteria such as parameter accuracy, empirical risk and generalization ability. We advocate the use of this principle for all types of optimization problems. By applying the ASC for the first time to a factual clustering problem, we have bridged the gap between the theory [20] and application.

Chapter 11

Conclusion

This thesis investigates the challenging problem of learning from data where some data items are jointly generated by several sources. Starting from a general model for the generative process of such data, we develop models and algorithms for multi-label classification (supervised learning) as well as for multi-assignment clustering (unsupervised learning). These models are explored in the context of an acoustic data mining problem and a information security application.

Hearing instruments are widely used in western societies and allow its wearers pleasant social interactions in spite of a hearing impairment. The performance of these devices critically depends on the capability to detect the current acoustic situation and thus to adapt the signal processing. In this demanding setting, the proposed generative multi-label classifier yields lower classification errors than state-of-the-art algorithms. While the extent of the difference depends on the evaluation criterion, clear improvements are observed for all quality measures. The more complex parameter optimization is worthwhile namely in settings where only a small training data set is available. We expect an extended acoustic model including e.g. reverberation to further improve the hearing comfort for hearing instrument users.

In a theoretical study on the distribution of parameter estimators based on multi-label data, we confirm that the proposed generative method outperforms its competitors in the accuracy of the parameter estimators. Together with the fact that generative models with consistent parameter estimators yield asymptotically optimal classification results, this observation provides a theoretically well-based explanation for the superior classifica-

tion performance. At the same time, this finding confirms the power of appropriately complex statistical models compared to oversimplified modeling. Furthermore, we have proven that some of the commonly made assumptions in multi-label classification imply a model mismatch. Inference methods relying on these assumptions thus obtain inconsistent parameter estimators.

In the unsupervised learning scenario, we again observe a superior accuracy of parameter estimators for models which take the generative nature of the data into account, compared to methods without this additional knowledge. In addition, the obtained clustering solutions are more stable under resampling of the data, and the cluster representatives offer a more precise description of previously unseen data. Namely the second property is a key feature in the application of role mining for role-based access control: The set of roles inferred from a direct access-control matrix should endow new users with the required permissions. Furthermore, we again observe that some of the previously proposed models rely on unrealistic assumptions and are therefore systematically disadvantaged in this real-world problem.

Cluster validation is a hard problem, and most validation techniques rely on certain assumptions. The framework of approximation set coding allows us to compare different versions of our clustering algorithm without presuming particular characteristics of a clustering solution. In several experiments, we observe that model selection based on this universal cluster validation framework chooses the model which performs best in terms of specialized quality measures. Furthermore, this being the first application of approximation set coding to a real-world problem, we facilitate the use of assumption-free evaluation criteria for clustering.

In conclusion, this thesis demonstrates the importance of a sufficiently general and complex model for both classification and clustering. Algorithms derived from generative models are applied to two important real-world problems and outperform state-of-the-art methods which are based on overly simplifying or unrealistic assumptions. A general theoretical framework facilitates the extension of this work to further problems in machine learning.

Appendix A

Proofs

A.1 Asymptotic Distribution of Estimators

This section contains the proofs of the lemmata describing the asymptotic distribution of estimators obtained by the inference methods $\mathcal{M}_{ignore}$, $\mathcal{M}_{new}$ and $\mathcal{M}_{cross}$ in Section 6.3.

Proof. **Lemma 1.** $\mathcal{M}_{ignore}$ reduces the estimation problem to the standard single-label classification problem for K independent sources. The results of standard (single-label) asymptotic analysis are directly applicable: The estimators $\hat{\boldsymbol{\theta}}^{ignore}$ are consistent and converge to the true parameter $\boldsymbol{\theta}^G$.

As only single-label data is used in the estimation process, the estimators for different sources are independent and the asymptotic covariance matrix is block-diagonal, as stated in Eq. 6.41. The diagonal elements are given by Eq. 6.35, which yields the expression given in the Lemma. □

Proof. **Lemma 2.** $\mathcal{M}_{new}$ reduces the estimation problem to the standard single-label classification problem for $L := |\mathbb{L}|$ independent sources. The results of standard asymptotic analysis (Section 6.1.4) are therefore directly applicable: The parameter estimators $\hat{\boldsymbol{\theta}}new$ for all single-label sources (including the proxy-distributions) are consistent with the true parameter values $\boldsymbol{\theta}^G$ and asymptotically normally distributed, as stated in the lemma.

The covariance matrix of the estimators is block-diagonal as the parameters are estimated independently for each source. Using Eq. 6.35, we obtain the values for the diagonal elements as given in the lemma. □

Proof. **Lemma 3.** The parameters θ_k of source k are estimated independently for each source. Combining Eq. 6.24 and Eq. 6.52, the condition for θ_k is

$$\Psi_N^{cross}(\theta_k) := \sum_D \psi_{\theta_k}^{cross}(D) \stackrel{!}{=} 0$$

$\psi_{\theta_k}^{cross}(D) = 0$ in the case $k \notin \mathcal{L}$ thus implies that D has no an influence on the parameter estimation.

For simpler notation, we define the set of all label sets which contain k as $\mathbb{L}_k$, formally $\mathbb{L}_k := \{\mathcal{L} \in \mathbb{L} | k \in \mathcal{L}\}$. The asymptotic criterion function for θ_k is then given by

$$\Psi^{cross}(\theta_k) = \mathbb{E}_{D\sim P_{\boldsymbol{\theta}G}}\left[\mathbb{E}_{\Xi_k\sim P_{D,\theta_k}^{cross}}[\phi(\Xi_k)]\right] - \mathbb{E}_{\Xi_k\sim P_{\theta_k}}[\phi(\Xi_k)] \tag{A.1}$$

$$= \sum_{\mathcal{L}\in\mathbb{L}_k} \pi_{\mathcal{L}} \mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}G}}[\phi(X)] + \sum_{\mathcal{L}\notin\mathbb{L}_k} \pi_{\mathcal{L}} \mathbb{E}_{\Xi\sim P_{\theta_k}}[\phi(X)] - \mathbb{E}_{\Xi_k\sim P_{\theta_k}}[\phi(\Xi_k)] \tag{A.2}$$

Setting $\Psi^{cross}(\theta_k) = 0$ yields

$$\mathbb{E}_{X\sim P_{\hat{\theta}_k^{cross}}}[\boldsymbol{\phi}(X)] = \frac{1}{1 - \sum_{\mathcal{L}\notin\mathbb{L}_k} \pi_{\mathcal{L}}} \sum_{\mathcal{L}\in\mathbb{L}_k} \pi_{\mathcal{L}} \mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}G}}[\boldsymbol{\phi}(X)] \ . \tag{A.3}$$

The mismatch of $\hat{\theta}_k^{cross}$ thus grows as the fraction of multi-label data grows. Furthermore, the mismatch depends on the dissimilarity of the sufficient statistics of the partner labels from the sufficient statistics of source k. □

A.2 Lemma 4

Proof. **Lemma 4**. This proof consists mainly of computing summary statistics.

Ignore Training ($\mathcal{M}_{ignore}$)

Mean Value of the Mean Estimator. As derived in the general description of the method in Section 6.3.1, the ignore training yields consistent estimators for the single-label source distributions:

$$\hat{\theta}_{1,1} \to -\frac{a}{\sigma_1^2} \qquad \hat{\theta}_{2,1} \to \frac{a}{\sigma_2^2}$$

Variance of the Mean Estimator. Recall that we assume to have $\pi_{\mathcal{L}} N$ observations with label set $\mathcal{L}$, and the variance of the source emissions is assumed to be $\mathbb{V}_{\Xi \sim P_k}[\phi(\Xi)] = \sigma_k^2$. The variance of the estimator for the single-label source means based on a training set of size N is thus $\mathbb{V}[\hat{\mu}_k] = \sigma_k^2/(\pi_k N)$.

Mean-Squared Error of the Estimator. With the above, the mean square error, averaged over the two sources, is given by

$$MSE(\hat{\boldsymbol{\theta}}_{\mu}^{ignore}, \boldsymbol{\theta}) = \frac{1}{2}\left(\frac{{\sigma_1}^2}{\pi_1 N} + \frac{{\sigma_2}^2}{\pi_2 N}\right) .$$

Since the estimators obtained by $\mathcal{M}_{ignore}$ are consistent, the mean square error only depends on the variance of the estimator.

New Source Training ($\mathcal{M}_{new}$)

Mean Value of the Estimator. The new source training is based on single-label data items and therefore, according to Thm. 3 yields consistent estimators. Note that this method uses three sources to model the generative process in the given example:

$$\hat{\theta}_{1,1} \to -\frac{a}{\sigma_1^2} \qquad \hat{\theta}_{2,1} \to \frac{a}{\sigma_2^2} \qquad \hat{\theta}_{12,1} \to 0$$

Variance of the Mean Estimator. The variance is given by Eq. 6.49 and takes the following values in our setting:

$$\mathbb{V}[\hat{\mu}_1] = \frac{{\sigma_1}^2}{\pi_1 N} \qquad \mathbb{V}[\hat{\mu}_2] = \frac{{\sigma_2}^2}{\pi_2 N} \qquad \mathbb{V}[\hat{\mu}_{12}] = \frac{{\sigma_{12}}^2}{\pi_{12} N} = \frac{{\sigma_1}^2 + {\sigma_2}^2}{\pi_{12} N}$$

Since the observations with label set $\mathcal{L} = \{1, 2\}$ have a higher variance than single-label observations, the estimator $\hat{\mu}_{12}$ also has a higher variance than the estimators for single sources.

Mean-Squared Error of the Estimator. Given the above, the mean square error is given by

$$MSE(\hat{\boldsymbol{\theta}}_{\mu}^{new}, \boldsymbol{\theta}) = \frac{1}{3}\left(\frac{{\sigma_1}^2}{\pi_1 N} + \frac{{\sigma_2}^2}{\pi_2 N} + \frac{{\sigma_1}^2 + {\sigma_2}^2}{\pi_{12} N}\right) .$$

Cross-Training ($\mathcal{M}_{cross}$)

As described in Equation 6.50, the probability distributions of the source emissions given the observations are assumed to be mutually independent by $\mathcal{M}_{cross}$. The criterion function $\psi_{\theta_k}^{cross}(D)$ is given in Equation 6.52. The parameter θ_k is chosen according to Eq. A.3:

$$\mathbb{E}_{X\sim P_{\theta_k^{cross}}}[X] = \frac{1}{1-\sum_{\mathcal{L}\notin\mathbb{L}_k}\pi_{\mathcal{L}}} \sum_{\mathcal{L}\in\mathbb{L}_k} \pi_{\mathcal{L}}\mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}[X] \tag{A.4}$$

Mean Value of the Mean Estimator. With the conditional expectations of the observations given the labels (see Eq. 6.61), we have for the mean estimate of source 1:

$$\begin{aligned}\hat{\mu}_1 = \mathbb{E}_{X\sim P_{\theta_1^{cross}}}[X] &= \frac{1}{1-\pi_2}\left(\pi_1\mathbb{E}_{X\sim P_{\{1\},\theta^G}}[X] + \pi_{12}\mathbb{E}_{X\sim P_{\{1,2\},\boldsymbol{\theta}^G}}[X]\right) \\ &= -\frac{\pi_1\cdot a}{\pi_1+\pi_{12}} = -\frac{a}{1+\frac{\pi_{12}}{\pi_1}}\end{aligned} \tag{A.5}$$

and, similarly, for source 2,

$$\hat{\mu}_2 = \frac{\pi_2\cdot a}{\pi_2+\pi_{12}} = \frac{a}{1+\frac{\pi_{12}}{\pi_2}} \tag{A.6}$$

The deviation from the true value increases with the ratio of multi-labeled data items compared to the number of single-label data items from the corresponding source.

Mean Value of the Standard Deviation Estimator. According to the principle of maximum likelihood, the estimator for the source variance σ_k^2 is the empirical variance of all data items which contain k their label sets:

$$\begin{aligned}\hat{\sigma}_1^2 &= \frac{1}{|D_1\cup D_{12}|}\sum_{x\in(D_1\cup D_{12})}(x-\hat{\mu}_1)^2 \\ &= \frac{1}{N(\pi_1+\pi_{12})}\left(\sum_{x\in D_1}(x-\hat{\mu}_1)^2 + \sum_{x\in D_{12}}(x-\hat{\mu}_1)^2\right) \\ &= \frac{\pi_1\pi_{12}}{(\pi_1+\pi_{12})^2}a^2 + \frac{\pi_1\sigma_{G,1}^2+\pi_{12}\sigma_{G,12}^2}{\pi_1+\pi_{12}}\end{aligned} \tag{A.7}$$

Quantity	$\mathcal{L} = \{1\}$	$\mathcal{L} = \{2\}$	$\mathcal{L} = \{1, 2\}$
$\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} X \\ \hat{\theta}_{2,1} \end{pmatrix}$	$\begin{pmatrix} \hat{\theta}_{1,1} \\ x \end{pmatrix}$	$\begin{pmatrix} X \\ X \end{pmatrix}$
$\mathbb{E}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]$	$\begin{pmatrix} -a \\ \hat{\mu}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{\mu}_1 \\ a \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$\mathbb{V}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]$	$\begin{pmatrix} \sigma_1^2 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & \sigma_1^2 \end{pmatrix}$	$\begin{pmatrix} \sigma_{12}^2 & \sigma_{12}^2 \\ \sigma_{12}^2 & \sigma_{12}^2 \end{pmatrix}$

Table A.1: Quantities used to determine the asymptotic behavior of parameter estimators obtained by $\mathcal{M}_{cross}$ for a Gaussian distribution.

and similarly

$$\hat{\sigma}_2^2 = \frac{\pi_2 \pi_{12} a^2}{(\pi_2 + \pi_{12})^2} + \frac{\pi_2 \sigma_{G,2}^2 + \pi_{12} \sigma_{G,12}^2}{\pi_2 + \pi_{12}} \,. \tag{A.8}$$

The variance of the source emissions under the assumptions of method $\mathcal{M}_{cross}$ is given by $\mathbb{V}_{\boldsymbol{\Xi} \sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \operatorname{diag}\left(\hat{\sigma}_1^2, \hat{\sigma}_2^2\right)$.

Variance of the Mean Estimator. We use the decomposition derived in Section 6.2.6 to determine the variance. Using the expected values of the sufficient statistics conditioned on the label sets and the variances thereof, as given in Table A.1, we have

$$\mathbb{E}_{\mathcal{L}}\left[\mathbb{V}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] = \begin{pmatrix} \pi_1 \sigma_1^2 + \pi_{12} \sigma_{12}^2 & \pi_{12} \sigma_{12}^2 \\ \pi_{12} \sigma_{12}^2 & \pi_2 \sigma_2^2 + \pi_{12} \sigma_{12}^2 \end{pmatrix}.$$

Furthermore, the expected value of the sufficient statistics over all data items is

$$\mathbb{E}_{D \sim P_{\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{D,\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] = \begin{pmatrix} -\pi_1 a + \pi_2 \hat{\mu}_1 \\ \pi_1 \hat{\mu}_2 + \pi_2 a \end{pmatrix}$$

Hence

$$\mathbb{E}_{\mathcal{L} \sim P_\pi}\left[\left(\mathbb{E}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] - \mathbb{E}_{D' \sim P_{\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{cross}_{D',\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right]\right)^{\otimes}\right]$$
$$= \begin{pmatrix} \frac{\pi_1 \pi_{12}}{\pi_1 + \pi_{12}} a^2 & -\frac{\pi_1 \pi_{12} \pi_2}{(\pi_1 + \pi_{12})(\pi_2 + \pi_{12})} a^2 \\ -\frac{\pi_1 \pi_{12} \pi_2}{(\pi_1 + \pi_{12})(\pi_2 + \pi_{12})} a^2 & \frac{\pi_2 \pi_{12}}{\pi_2 + \pi_{12}} a^2 \end{pmatrix}$$

The variance of the sufficient statistics of the emissions of single sources and the generalized Fisher information matrices for each label set are thus given by

$$\mathbb{V}_{\boldsymbol{\Xi}\sim P^{cross}_{(X,\{1\}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \begin{pmatrix} 0 & 0 \\ 0 & \hat{\sigma}_2^2 \end{pmatrix} \qquad \mathcal{I}_{\{1\}} = -\begin{pmatrix} \hat{\sigma}_1^2 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbb{V}_{\boldsymbol{\Xi}\sim P^{cross}_{(X,\{2\}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \begin{pmatrix} \hat{\sigma}_1^2 & 0 \\ 0 & 0 \end{pmatrix} \qquad \mathcal{I}_{\{2\}} = -\begin{pmatrix} 0 & 0 \\ 0 & \hat{\sigma}_2^2 \end{pmatrix}$$

$$\mathbb{V}_{\boldsymbol{\Xi}\sim P^{cross}_{(X,\{1,2\}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \qquad \mathcal{I}_{\{1,2\}} = -\begin{pmatrix} \hat{\sigma}_1^2 & 0 \\ 0 & \hat{\sigma}_2^2 \end{pmatrix}$$

The expected value of the generalized Fisher information matrices over all label sets is

$$\mathbb{E}_{\mathcal{L}\sim P_{\mathcal{L}}}[\mathcal{I}_{\mathcal{L}}] = -\mathrm{diag}\left((\pi_1+\pi_{12})\hat{\sigma}_1^2, (\pi_2+\pi_{12})\hat{\sigma}_2^2\right)$$

where the values of $\hat{\sigma}_1$ and $\hat{\sigma}_2$ are given in Eq. A.7 and Eq. A.8.

Putting everything together, the diagonal entries of the covariance matrix of the estimator $\hat{\boldsymbol{\theta}}^{cross}$ is given by

$$\Sigma_{\boldsymbol{\theta}}^{cross} = \begin{pmatrix} v_{\theta,11} & v_{\theta,12} \\ v_{\theta,12} & v_{\theta,22} \end{pmatrix} \tag{A.9}$$

with diagonal elements

$$v_{\theta,11} = \frac{\pi_1+\pi_{12}}{\pi_1\pi_{12}a^2 + \pi_1\sigma_1^2 + \pi_{12}\sigma_{12}^2} \tag{A.10}$$

$$v_{\theta,22} = \frac{\pi_2+\pi_{12}}{\pi_2\pi_{12}a^2 + \pi_2\sigma_1^2 + \pi_{12}\sigma_{12}^2}\ . \tag{A.11}$$

To get the variance of the mean estimator, recall Eq. 6.59. The covariance matrix for the mean estimator is

$$\Sigma_{\boldsymbol{\mu}}^{cross} = \begin{pmatrix} v_{\mu,11} & v_{\mu,12} \\ v_{\mu,12} & v_{\mu,22} \end{pmatrix}$$

$$\text{with} \qquad v_{\mu,11} = \frac{1}{\pi_1+\pi_{12}} \cdot \left(\frac{\pi_1\pi_{12}}{(\pi_1+\pi_{12})^2} a^2 + \frac{\pi_1\sigma_1^2 + \pi_{12}\sigma_{12}^2}{\pi_1+\pi_{12}} \right)$$

$$v_{\mu,22} = \frac{1}{\pi_2+\pi_{12}} \cdot \left(\frac{\pi_2\pi_{12}}{(\pi_2+\pi_{12})^2} a^2 + \frac{\pi_2\sigma_2^2 + \pi_{12}\sigma_{12}^2}{\pi_2+\pi_{12}} \right)\ .$$

The first term in the brackets gives the variance of the means of the two true sources involved in generating the samples used to estimate the mean of the particular source. The second term is the average variance of the sources.

Mean-Squared Error of the Mean Estimator. Finally, the Mean Squared Error is given by:

$$
\begin{aligned}
&MSE(\hat{\boldsymbol{\mu}}^{cross}, \boldsymbol{\mu}) \\
&\quad = \frac{1}{2}\pi_{12}^2 \left(\frac{1}{(\pi_1+\pi_{12})^2} + \frac{1}{(\pi_2+\pi_{12})^2} \right) a^2 \\
&\quad + \frac{1}{2}\pi_{12} \left(\frac{1}{(\pi_1+\pi_{12})N} \frac{\pi_1}{(\pi_1+\pi_{12})^2} + \frac{1}{(\pi_2+\pi_{12})N} \frac{\pi_2}{(\pi_2+\pi_{12})^2} \right) a^2 \\
&\quad + \frac{1}{2} \left(\frac{1}{(\pi_1+\pi_{12})N} \frac{\pi_1\sigma_1^2 + \pi_{12}\sigma_{12}^2}{\pi_1+\pi_{12}} + \frac{1}{(\pi_2+\pi_{12})N} \frac{\pi_2\sigma_2^2 + \pi_{12}\sigma_{12}^2}{\pi_2+\pi_{12}} \right)
\end{aligned}
$$

This expression describes the three effects contributing to the estimation error incurred by $\mathcal{M}_{cross}$:

- The first line indicates the inconsistency of the estimator. This term grows with the mean of the true sources (a and $-a$, respectively) and with the ratio of multi-label data items. Note that this term is independent of the number of data items.
- The second line measures the variance of the observation x given the label set $\mathcal{L}$, averaged over all label sets and all sources. This term thus describes the excess variance of the estimator due to the inconsistency in the estimation procedure.
- The third line is the weighted average of the variance of the individual sources, as it is also found for consistent estimators.

The second and third line describe the variance of the observations according to the law of total variance:

$$
\mathbb{V}_X[X] = \underbrace{\mathbb{V}_{\mathcal{L}}[\mathbb{E}_X[X|\mathcal{L}]]}_{\text{second line}} + \underbrace{\mathbb{E}_{\mathcal{L}}[\mathbb{V}_X[X|\mathcal{L}]]}_{\text{third line}} \tag{A.12}
$$

Note that $(\pi_1+\pi_{12})N$ and $(\pi_2+\pi_{12})N$ is the number if data items used to infer the parameters of source 1 and 2, respectively.

Deconvolutive Training ($\mathcal{M}_{deconv}$)

Mean Value of the Mean Estimator. The conditional expectations of the sufficient statistics of the single-label data are:

$$
\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1\}),\boldsymbol{\theta}}(\boldsymbol{\Xi})}[\phi_1(\boldsymbol{\Xi})] = \begin{pmatrix} X \\ \hat{\mu}_2 \end{pmatrix} \qquad \mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{2\}),\boldsymbol{\theta}}(\boldsymbol{\Xi})}[\phi_1(\boldsymbol{\Xi})] = \begin{pmatrix} \hat{\mu}_1 \\ X \end{pmatrix}
$$

Observations X with label set $\mathcal{L} = \{1, 2\}$ are interpreted as the sum of the emissions from the two sources. Therefore, there is no unique expression for the conditional expectation of the source emissions given the data item $D = (X, \mathcal{L})$:

$$\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\{1\}),\boldsymbol{\theta}}(\boldsymbol{\Xi})}[\phi_1(\boldsymbol{\Xi})] = \begin{pmatrix} \hat{\mu}_1 \\ X - \hat{\mu}_1 \end{pmatrix} = \begin{pmatrix} X - \hat{\mu}_2 \\ \hat{\mu}_2 \end{pmatrix}$$

We use a the parameter $\lambda \in [0, 1]$ to parameterize the blending between these two extreme cases:

$$\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\{1\}),\boldsymbol{\theta}}(\boldsymbol{\Xi})}[\phi_1(\boldsymbol{\Xi})] = \lambda \begin{pmatrix} \hat{\mu}_1 \\ X - \hat{\mu}_1 \end{pmatrix} + (1 - \lambda) \begin{pmatrix} X - \hat{\mu}_2 \\ \hat{\mu}_2 \end{pmatrix}$$

Furthermore, we have

$$\mathbb{E}_{\boldsymbol{\Xi} \sim P_{\boldsymbol{\theta}}}[\phi_1(\boldsymbol{\Xi})] = \begin{pmatrix} \hat{\mu}_1 \\ \hat{\mu}_2 \end{pmatrix}$$

The criterion function $\Psi^{deconv}_{\boldsymbol{\theta}}(D)$ for the parameter vector $\boldsymbol{\theta}$ then implies the condition

$$\pi_1 \begin{pmatrix} \bar{X}_1 \\ \hat{\mu}_2 \end{pmatrix} + \pi_2 \begin{pmatrix} \hat{\mu}_1 \\ \bar{X}_2 \end{pmatrix} + \pi_{12} \begin{pmatrix} \lambda\hat{\mu}_1 + (1-\lambda)(\bar{X}_{12} - \hat{\mu}_2) \\ \lambda(\bar{X}_{12} - \hat{\mu}_1) + (1-\lambda)\hat{\mu}_2 \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} \hat{\mu}_1 \\ \hat{\mu}_2 \end{pmatrix} ,$$

where we have defined $\bar{X}_1$ and $\bar{X}_2$ as the average of the observations with label set $\{1\}$ and $\{2\}$, respectively, and $\bar{X}_{12}$ as the average of the observations with label set $\{1, 2\}$. Solving for $\hat{\boldsymbol{\mu}}$, we get

$$\hat{\mu}_1 = \frac{1}{2}\left((1+\lambda)\bar{X}_1 + (1-\lambda)\bar{X}_{12} - (1-\lambda)\bar{X}_2\right) \tag{A.13}$$

$$\hat{\mu}_2 = \frac{1}{2}\left(-\lambda\bar{X}_1 + \lambda\bar{X}_{12} + (2-\lambda)\bar{X}_2\right) \ . \tag{A.14}$$

Since

$$\mathbb{E}\left[\bar{X}_1\right] = -a \qquad \mathbb{E}\left[\bar{X}_{12}\right] = 0 \qquad \mathbb{E}\left[\bar{X}_2\right] = a \ ,$$

we find that the mean estimators are consistent:

$$\mathbb{E}[\mu_1] = -a \qquad \mathbb{E}[\mu_2] = a \ , \tag{A.15}$$

independent of the chosen value for λ. In particular, we have

$$\mathbb{E}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] = \mathbb{E}_{D' \sim P_{\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{D',\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]$$

for all $\mathcal{L}$.

Mean of the Variance Estimator. We compute the second component $\phi_2(\boldsymbol{\Xi})$ of the sufficient statistics vector $\boldsymbol{\phi}(\boldsymbol{\Xi})$ for the emissions given a data item. Since the estimators for the mean are consistent, we do not distinguish between the true and the estimated mean values any more. For single-label data items, we have

$$\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1\}),\hat{\boldsymbol{\theta}}}(\boldsymbol{\Xi})}[\phi_2(\boldsymbol{\Xi})] = \begin{pmatrix} X^2 \\ \mu_2^2 + \hat{\sigma}_2^2 \end{pmatrix}$$
$$\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{2\}),\hat{\boldsymbol{\theta}}}(\boldsymbol{\Xi})}[\phi_2(\boldsymbol{\Xi})] = \begin{pmatrix} \mu_1^2 + \hat{\sigma}_1^2 \\ X^2 \end{pmatrix}$$

For multi-label data items, the situation is again more involved. As when determining the estimator for the mean, we find again two extreme cases:

$$\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1,2\}),\hat{\boldsymbol{\theta}}}}[\phi_2(\boldsymbol{\Xi})] = \begin{pmatrix} X^2 - \mu_2^2 - \hat{\sigma}_2^2 \\ \mu_2^2 + \hat{\sigma}_2^2 \end{pmatrix} = \begin{pmatrix} \mu_1^2 + \hat{\sigma}_1^2 \\ X^2 - \mu_1^2 - \hat{\sigma}_1^2 \end{pmatrix}$$

We use again a parameter $\lambda \in [0,1]$ to parameterize the blending between the two extreme cases and write

$$\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1,2\}),\hat{\boldsymbol{\theta}}}}[\phi_2(\boldsymbol{\Xi})] = \lambda \begin{pmatrix} X^2 - \mu_2^2 - \hat{\sigma}_2^2 \\ \mu_2^2 + \hat{\sigma}_2^2 \end{pmatrix} + (1-\lambda) \begin{pmatrix} \mu_1^2 + \hat{\sigma}_1^2 \\ X^2 - \mu_1^2 - \hat{\sigma}_1^2 \end{pmatrix}$$

Using $\mathbb{E}_{X\sim P_{\{1\},\boldsymbol{\theta}^G}}\left[X^2\right] = \mu_1^2 + \sigma_1^2$, and similarly for other label sets, the criterion function implies the following condition for the standard deviation parameters

$$\pi_1 \begin{pmatrix} \mu_1^2 + \sigma_1^2 \\ \mu_2^2 + \hat{\sigma}_2^2 \end{pmatrix} + \pi_2 \begin{pmatrix} \mu_1^2 + \hat{\sigma}_1^2 \\ \mu_2^2 + \sigma_2^2 \end{pmatrix}$$
$$+ \pi_{12} \begin{pmatrix} \lambda\left(\mu_1^2 + \sigma_1^2 + \sigma_2^2 - \hat{\sigma}_2^2\right) + (1-\lambda)\left(\mu_1^2 + \hat{\sigma}_1^2\right) \\ \lambda\left(\mu_2^2 + \hat{\sigma}_2^2\right) + (1-\lambda)\left(\mu_2^2 + \sigma_1^2 + \sigma_2^2 - \hat{\sigma}_1^2\right) \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} \mu_1^2 + \hat{\sigma}_1^2 \\ \mu_2^2 + \hat{\sigma}_2^2 \end{pmatrix}$$

Solving for $\hat{\sigma}_1$ and $\hat{\sigma}_2$, we find

$$\hat{\sigma}_1 = \sigma_1 \qquad \hat{\sigma}_2 = \sigma_2 \ .$$

The estimators for the standard deviation are thus consistent as well.

Variance of the Mean Estimator. The variance of the conditional expectation values over observations X with label set $\mathcal{L}$, for the three possible

label sets, is given by

$$\begin{aligned}
\mathbb{V}_{X\sim P_{\{1\},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1\}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] &= \operatorname{diag}\left(\sigma_1^2, 0\right)\\
\mathbb{V}_{X\sim P_{\{2\},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{2\}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] &= \operatorname{diag}\left(0, \sigma_2^2\right)\\
\mathbb{V}_{X\sim P_{\{1,2\},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1,2\}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] &= \begin{pmatrix} (1-\lambda)^2 & \lambda(1-\lambda) \\ \lambda(1-\lambda) & \lambda^2 \end{pmatrix}\sigma_{12}^2
\end{aligned}$$

and thus

$$\begin{aligned}
&\mathbb{E}_{\mathcal{L}\sim P_\pi}\left[\mathbb{V}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\mathcal{L}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right]\\
&\quad= \begin{pmatrix} \pi_1\sigma_1^2 & 0 \\ 0 & \pi_2\sigma_2^2 \end{pmatrix} + \pi_{12}\begin{pmatrix} (1-\lambda)^2 & \lambda(1-\lambda) \\ \lambda(1-\lambda) & \lambda^2 \end{pmatrix}\sigma_{12}^2
\end{aligned}$$

The variance of the assumed source emissions are given by

$$\begin{aligned}
\mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1\},\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] &= \operatorname{diag}\left(0, \sigma_2^2\right)\\
\mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{2\},\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] &= \operatorname{diag}\left(\sigma_1^2, 0\right)\\
\mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1,2\},\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] &= \mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(X,\{1,2\},\boldsymbol{\theta}}}\left[\begin{pmatrix} \lambda\Xi_1 + (1-\lambda)(X-\Xi_2) \\ \lambda(X-\Xi_1) + (1-\lambda)\Xi_2 \end{pmatrix}\right]\\
&= \lambda^2\begin{pmatrix} \sigma_1^2 & -\sigma_1^2 \\ -\sigma_1^2 & \sigma_1^2 \end{pmatrix} + (1-\lambda)^2\begin{pmatrix} \sigma_2^2 & -\sigma_2^2 \\ -\sigma_2^2 & \sigma_2^2 \end{pmatrix}
\end{aligned}$$

With $\mathbb{V}_{\boldsymbol{\Xi}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \operatorname{diag}\left(\sigma_1^2, \sigma_2^2\right)$, the generalized Fisher information matrices for the single-label data are given by

$$\mathcal{I}_{\{1\}} = -\operatorname{diag}\left(\sigma_1^2, 0\right) \qquad \mathcal{I}_{\{2\}} = -\operatorname{diag}\left(0, \sigma_2^2\right)$$

For the label set $\mathcal{L} = \{1, 2\}$, we have

$$\mathcal{I}_{\{1,2\}} = \begin{pmatrix} (\lambda^2-1)\sigma_1^2 + (1-\lambda)^2\sigma_2^2 & -\lambda^2\sigma_1^2 - (1-\lambda)^2\sigma_2^2 \\ -\lambda^2\sigma_1^2 - (1-\lambda)^2\sigma_2^2 & \lambda^2\sigma_1^2 + \left((1-\lambda)^2 - 1\right)\sigma_2^2 \end{pmatrix}$$

Choosing λ such that the trace of the information matrix $\mathcal{I}_{\{1,2\}}$ is maximized yields $\lambda = \sigma_2^2/\left(\sigma_1^2 + \sigma_2^2\right)$ and the following value for the information matrix of label set $\{1, 2\}$:

$$\mathcal{I}_{\{1,2\}} = -\begin{pmatrix} \frac{\sigma_1^4}{\sigma_1^2+\sigma_2^2} & \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} \\ \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} & \frac{\sigma_2^4}{\sigma_1^2+\sigma_2^2} \end{pmatrix} = -\frac{1}{\sigma_1^2+\sigma_2^2}\begin{pmatrix} \sigma_1^4 & \sigma_1^2\sigma_2^2 \\ \sigma_1^2\sigma_2^2 & \sigma_2^4 \end{pmatrix}$$

The expected Fisher information matrix is then given by

$$\mathbb{E}_{\mathcal{L}\sim P_\pi}[\mathcal{I}_\mathcal{L}] = -\begin{pmatrix} \sigma_1^2\left(\pi_1 + \pi_{12}\frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\right) & \pi_{12}\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} \\ \pi_{12}\frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2} & \sigma_2^2\left(\pi_2 + \pi_{12}\frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\right) \end{pmatrix}$$

With this, $\Sigma_{\boldsymbol{\theta}}^{deconv}$ is

$$\Sigma_{\boldsymbol{\theta}}^{deconv} = \begin{pmatrix} v_{\theta,11}^2 & v_{\theta,12}^2 \\ v_{\theta,12}^2 & v_{\theta,22}^2 \end{pmatrix} ,$$

with the matrix elements given by

$$\begin{aligned} v_{\theta,11}^2 &= \frac{\pi_{12}^2\sigma_2^2 w_{12} + \pi_{12}\pi_2\left(\pi_2\sigma_1^2\sigma_{12}^2 + 2\pi_1\sigma_2^2 s_{12}\right) + \pi_1\pi_2^2 s_{12}^2}{\sigma_1^2\left(\pi_1\pi_2 s_{12} + \pi_{12}w_{12}\right)^2} \\ v_{\theta,12}^2 &= \frac{\pi_{12}^2 w_{12} + \pi_{12}\pi_1\pi_2(2s_{12} - \sigma_{12}^2)}{\left(\pi_1\pi_2 s_{12} + \pi_{12}w_{12}\right)^2} \\ v_{\theta,22}^2 &= \frac{\pi_{12}^2\sigma_1^2 w_{12} + \pi_{12}\pi_1(\pi_1\sigma_2^2\sigma_{12}^2 + 2\pi_2\sigma_1^2 s_{12}) + \pi_1^2\pi_2 s_{12}^2}{\sigma_2^2\left(\pi_1\pi_2 s_{12} + \pi_{12}w_{12}\right)^2} \end{aligned}$$

where, for simpler notation, we have defined

$$w_{12} := \pi_2\sigma_1^2 + \pi_1\sigma_2^2 \qquad s_{12} := \sigma_1^2 + \sigma_2^2 .$$

For the variance of the mean estimators, using Eq. 6.59, we get

$$\Sigma_{\boldsymbol{\mu}}^{deconv} = \begin{pmatrix} v_{\mu,11}^2 & v_{\mu,12}^2 \\ v_{\mu,12}^2 & v_{\mu,22}^2 \end{pmatrix} ,$$

with the matrix elements given by

$$v_{\mu,11}^2 = \frac{\pi_{12}^2\sigma_2^2 w_{12} + \pi_{12}\pi_2\left(\pi_2\sigma_1^2\sigma_{12}^2 + 2\pi_1\sigma_2^2 s_{12}\right) + \pi_1\pi_2^2 s_{12}^2}{\left(\pi_1\pi_2 s_{12} + \pi_{12}w_{12}\right)^2}\sigma_1^2 \tag{A.16}$$

$$v_{\mu,12}^2 = \frac{\pi_{12}^2 w_{12} + \pi_{12}\pi_1\pi_2(2s_{12} - \sigma_{12}^2)}{\left(\pi_1\pi_2 s_{12} + \pi_{12}w_{12}\right)^2}\sigma_1^2\sigma_2^2 \tag{A.17}$$

$$v_{\mu,22}^2 = \frac{\pi_{12}^2\sigma_1^2 w_{12} + \pi_{12}\pi_1(\pi_1\sigma_2^2\sigma_{12}^2 + 2\pi_2\sigma_1^2 s_{12}) + \pi_1^2\pi_2 s_{12}^2}{\left(\pi_1\pi_2 s_{12} + \pi_{12}w_{12}\right)^2}\sigma_2^2 . \tag{A.18}$$

Quantity	$\mathcal{L}=\{1\}$	$\mathcal{L}=\{2\}$	$\mathcal{L}=\{1,2\}$
$\mathbb{E}_{\boldsymbol{\Xi}\sim P^{new}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} X \\ \hat{e}_2 \\ \hat{e}_{12} \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ X \\ \hat{e}_{12} \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \\ X \end{pmatrix}$
$\mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{new}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]$	$\begin{pmatrix} e_1 \\ \hat{e}_2 \\ \hat{e}_{12} \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ e_2 \\ \hat{e}_{12} \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \\ e_{12} \end{pmatrix}$

Table A.2: Expectation values used to determine the asymptotic behavior of the $\mathcal{M}_{new}$ inference technique.

Mean-Squared Error of the Mean Estimator. Given that the estimators $\boldsymbol{\mu}^{deconv}$ are consistent, the mean squared error of the estimator is given by the average of the diagonal elements of $\Sigma^{deconv}_{\boldsymbol{\mu}}$:

$$MSE^{deconv}_{\boldsymbol{\mu}} = \frac{1}{2}\mathrm{tr}\left(\Sigma^{deconv}_{\boldsymbol{\mu}}\right) = \frac{v^2_{\mu,11}+v^2_{\mu,22}}{2} \tag{A.19}$$

Inserting the expressions in Eq. A.16 and A.18 yields the expression given in the theorem. □

A.3 Lemma 5

We compute the mean and variance of the mean estimator for each of the considered inference techniques.

Proof. **Lemma 5**.

New Training

Mean Value of the Estimator. Using the expectation values as given in Table A.2 and setting

$$\mathbb{E}_{\mathcal{L}\sim P_{\boldsymbol{\theta}G}}\left[\mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{new}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] = \mathbb{E}_{\mathcal{L}\sim P_{\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \ ,$$

we get the conditions

$$\begin{aligned}
\pi_1 e_1 + (\pi_2+\pi_{12})\hat{e}_1 &= \hat{e}_1 \\
\pi_2 e_2 + (\pi_1+\pi_{12})\hat{e}_2 &= \hat{e}_2 \\
\pi_{12} e_{12} + (\pi_1+\pi_2)\hat{e}_{12} &= \hat{e}_{12}
\end{aligned}$$

Quantity	$\mathcal{L} = \{1\}$	$\mathcal{L} = \{2\}$	$\mathcal{L} = \{1,2\}$
$\mathbb{V}_X[\mathbb{E}_{\boldsymbol{\Xi}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]]$	$\begin{pmatrix} \hat{v}_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \hat{v}_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \hat{v}_{12} \end{pmatrix}$
$\mathbb{V}_{\boldsymbol{\Xi}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \hat{v}_2 & 0 \\ 0 & 0 & \hat{v}_{12} \end{pmatrix}$	$\begin{pmatrix} \hat{v}_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \hat{v}_{12} \end{pmatrix}$	$\begin{pmatrix} \hat{v}_1 & 0 & 0 \\ 0 & \hat{v}_2 & 0 \\ 0 & 0 & 0 \end{pmatrix}$
$\mathcal{I}_{\mathcal{L}}$	$-\begin{pmatrix} \hat{v}_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$-\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$	$-\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \hat{v}_{12} \end{pmatrix}$

Table A.3: Variances and information matrices used to determine the asymptotic behavior of the $\mathcal{M}_{new}$ inference technique. Since $\mathcal{M}_{new}$ is consistent, we set $\boldsymbol{\theta}^G = \boldsymbol{\theta}$. The random variables X and $\boldsymbol{\Xi}$ are distributed as $X \sim P_{\mathcal{L},\boldsymbol{\theta}^G}$ and $\boldsymbol{\Xi} \sim P^{new}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}$

Using $\pi_1 + \pi_2 + \pi_{12} = 1$, we find that all estimators are consistent:

$$\hat{\theta}_1 = \theta_{G,1} \qquad \hat{\theta}_2 = \theta_{G,2} \qquad \hat{\theta}_{12} = \theta_{G,12} \ .$$

This implies that $\hat{e}_k = e_k$ and $\hat{v}_k = v_k$ for $k = 1, 2, 12$. In the analysis of the variance of the estimator, we do not distinguish between $\boldsymbol{\theta}^G$ and $\hat{\boldsymbol{\theta}}^{new}$ and denote both the true and the inferred value by $\boldsymbol{\theta}$.

Variance of the Estimator. With the values of Table A.3, we have

$$\begin{aligned} \mathbb{E}_{\mathcal{L}}\left[\mathbb{V}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{new}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] &= \operatorname{diag}\left(\pi_1\hat{v}_1, \pi_2\hat{v}_2, \pi_{12}\hat{v}_{12}\right) \\ \mathbb{V}_{\boldsymbol{\Xi}\sim P_{\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] &= \operatorname{diag}\left(\hat{v}_1, \hat{v}_2, \hat{v}_{12}\right) \\ \mathbb{E}_{\mathcal{L}\sim P_{\hat{\boldsymbol{\theta}}}}[\mathcal{I}_{\mathcal{L}}] &= -\operatorname{diag}\left(\pi_1\hat{v}_1, \pi_2\hat{v}_2, \pi_{12}\hat{v}_{12}\right) \ . \end{aligned}$$

The variation of the expectations is thus equivalent to the expected generalized Fisher information matrix. The reduction of the multi-label learning problem to the single-label problem is thus done in a consistent way by $\mathcal{M}_{new}$. The variance of the source estimators is thus given by

$$\begin{aligned} \Sigma^{new}_{\boldsymbol{\theta}} &= \operatorname{diag}\left(\pi_1\hat{v}_1, \pi_2\hat{v}_2, \pi_{12}\hat{v}_{12}\right)^{-1} \\ &= \operatorname{diag}\left(\frac{1}{\pi_1}\frac{(1+\exp\theta_1)^2}{\exp(\theta_1)}, \frac{1}{\pi_2}\frac{(1+\exp\theta_2)^2}{\exp(\theta_2)}, \frac{1}{\pi_{12}}\frac{(1+\exp\theta_{12})^2}{\exp(\theta_{12})}\right) \end{aligned}$$

Note that this is the variance of the estimator $\hat{\boldsymbol{\theta}}^{new}$. To get the variance for the estimator $\hat{\boldsymbol{\beta}}^{new}$, we write β as a function of $\boldsymbol{\theta}$ and do a Taylor expansion. This yields

$$\mathbb{V}[\beta(\boldsymbol{\theta})] \approx \left(\frac{\partial\beta(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)^2 \mathbb{V}[\boldsymbol{\theta}] \tag{A.20}$$

With

$$\frac{\partial\hat{\beta}_k(\theta_k)}{\partial\theta_k} = \frac{\exp\theta_k}{(1+\exp\theta_k)^2}$$

this yields

$$\begin{aligned}\Sigma_{\boldsymbol{\beta}}^{new} &\approx \operatorname{diag}\left(\frac{1}{\pi_1}\frac{\exp\theta_1}{(1+\exp\theta_1)^2}, \frac{1}{\pi_2}\frac{\exp\theta_2}{(1+\exp\theta_2)^2}, \frac{1}{\pi_{12}}\frac{\exp\theta_{12}}{(1+\exp\theta_{12})^2}\right) \\ &= \operatorname{diag}\left(\frac{1}{\pi_1}\hat{\beta}_1(1-\hat{\beta}_1), \frac{1}{\pi_2}\hat{\beta}_2(1-\hat{\beta}_2), \frac{1}{\pi_{12}}\hat{\beta}_{12}(1-\hat{\beta}_{12})\right).\end{aligned}$$

We thus get the well-known result back.

Mean-Squared Error of the Estimator. Given that the estimators $\boldsymbol{\beta}^{new}$ are consistent, the mean squared error is determined by the variances of the estimators:

$$\begin{aligned}&MSE(\hat{\boldsymbol{\beta}}^{new}, \boldsymbol{\beta}) \\ &\quad= \frac{1}{3}\left(\frac{1}{\pi_1 N}\beta_1(1-\beta_1) + \frac{1}{\pi_2 N}\beta_2(1-\beta_2) + \frac{1}{\pi_{12}N}\beta_{12}(1-\beta_{12})\right) .\end{aligned}$$

Note that $\pi_1 N$ is the number of data items with label set $\{1\}$, i.e. the number of data items based on which $\hat{\beta}_1^{new}$ is estimated.

Ignore Training $\mathcal{M}_{ignore}$

The ignore training trains only a subset of the parameters, the estimation of the single parameters is based on the same data subsets as in $\mathcal{M}_{new}$. We therefore abbreviate the derivation of the properties for the concrete setting.

Mean Value of the Estimator. The estimator is consistent, i.e.

$$\hat{\theta}_1 = \theta_1 \qquad \hat{\theta}_2 = \theta_2$$

Again, we will therefore no longer distinguish between the estimators and the true values.

Variance of the Estimator. Using the results from the previous section on $\mathcal{M}_{new}$, we have

$$\begin{aligned}\Sigma_{\boldsymbol{\theta}}^{ignore} &= \operatorname{diag}\left(\pi_1\hat{v}_1, \pi_2\hat{v}_2\right)^{-1} \\ &= \operatorname{diag}\left(\frac{1}{\pi_1}\frac{(1+\exp\theta_1)^2}{\exp(\theta_1)}, \frac{1}{\pi_2}\frac{(1+\exp\theta_2)^2}{\exp(\theta_2)}\right) ,\end{aligned}$$

and

$$\begin{aligned}\Sigma_{\boldsymbol{\beta}}^{ignore} &\approx \operatorname{diag}\left(\frac{1}{\pi_1}\frac{\exp\theta_1}{(1+\exp\theta_1)^2}, \frac{1}{\pi_2}\frac{\exp\theta_2}{(1+\exp\theta_2)^2}\right) \\ &= \operatorname{diag}\left(\frac{1}{\pi_1}\hat{\beta}_1(1-\hat{\beta}_1), \frac{1}{\pi_2}\hat{\beta}_2(1-\hat{\beta}_2)\right) .\end{aligned}$$

Mean-Squared Error of the Estimator. With the above, we have

$$MSE(\hat{\boldsymbol{\beta}}^{ignore}, \boldsymbol{\beta}) = \frac{1}{2}\left(\frac{1}{\pi_1 N}\beta_1(1-\beta_1) + \frac{1}{\pi_2 N}\beta_2(1-\beta_2)\right) .$$

Again, $\pi_1 N$ is the number of data items with label set $\{1\}$, i.e. the number of data items based on which $\hat{\beta}_1^{ignore}$ is estimated. Since $\mathcal{M}_{ignore}$ ignores data with multiple labels, not all data items are used in the training, and we have $\pi_1 N + \pi_2 N < N$ whenever multi-label data items occur in the training data set.

Cross-Training $\mathcal{M}_{cross}$

Mean Value of the Estimator. Using the results from Table A.4, we get

$$\begin{aligned}\mathbb{E}_{\mathcal{L}}\left[\mathbb{E}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}G}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] &= \begin{pmatrix} \pi_1 e_1 + \pi_2\hat{e}_1 + \pi_{12}e_{12} \\ \pi_1\hat{e}_2 + \pi_2 e_2 + \pi_{12}e_{12} \end{pmatrix} \\ \mathbb{E}_{\boldsymbol{\Xi}\sim P_{\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] &= \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \end{pmatrix}\end{aligned}$$

Equating the two expected values, we get

$$\hat{e}_1 = \frac{\pi_1 e_1 + \pi_{12}e_{12}}{\pi_1 + \pi_{12}} \qquad \hat{e}_2 = \frac{\pi_2 e_2 + \pi_{12}e_{12}}{\pi_2 + \pi_{12}} . \tag{A.21}$$

Note that these estimators are not consistent, with the bias depending on π_{12}, the proportion of data items with label set $\mathcal{L} = \{1, 2\}$.

Quantity	$\mathcal{L}=\{1\}$	$\mathcal{L}=\{2\}$	$\mathcal{L}=\{1,2\}$
$\mathbb{E}_{\boldsymbol{\Xi}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} X \\ \hat{e}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ X \end{pmatrix}$	$\begin{pmatrix} X \\ X \end{pmatrix}$
$\mathbb{E}_X[\mathbb{E}_{\boldsymbol{\Xi}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]]$	$\begin{pmatrix} e_1 \\ \hat{e}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ e_2 \end{pmatrix}$	$\begin{pmatrix} e_{12} \\ e_{12} \end{pmatrix}$
$\mathbb{V}_X[\mathbb{E}_{\boldsymbol{\Xi}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]]$	$\begin{pmatrix} v_1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & v_2 \end{pmatrix}$	$\begin{pmatrix} v_{12} & v_{12} \\ v_{12} & v_{12} \end{pmatrix}$
$\mathbb{V}_{\boldsymbol{\Xi}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} 0 & 0 \\ 0 & \hat{v}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{v}_1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$
$\mathcal{I}_{\mathcal{L}}$	$-\begin{pmatrix} \hat{v}_1 & 0 \\ 0 & 0 \end{pmatrix}$	$-\begin{pmatrix} 0 & 0 \\ 0 & \hat{v}_2 \end{pmatrix}$	$-\begin{pmatrix} \hat{v}_1 & 0 \\ 0 & \hat{v}_2 \end{pmatrix}$

Table A.4: Quantities used to determine the asymptotic behavior of the $\mathcal{M}_{cross}$ inference technique. The distributions of the random variables are omitted for lack of space, they are $X \sim P_{\mathcal{L},\boldsymbol{\theta}^G}$ and $\boldsymbol{\Xi} \sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}$.

Variance of the Estimator. With the derived mean values, the expectation deviances (Eq. 6.33) for the three label set are as follows:

$$\Delta\mathbb{E}_{\{1\}} = \begin{pmatrix} \frac{\pi_{12}(e_1-e_{12})}{(\pi_1+\pi_{12})} \\ 0 \end{pmatrix}$$

$$\Delta\mathbb{E}_{\{2\}} = \begin{pmatrix} 0 \\ \frac{\pi_{12}(e_2-e_{12})}{\pi_2+\pi_{12}} \end{pmatrix}$$

$$\Delta\mathbb{E}_{\{1,2\}} = \begin{pmatrix} \frac{\pi_{12}(e_1-e_{12})}{\pi_1+\pi_{12}} \\ \frac{\pi_{12}(e_2-e_{12})}{\pi_2+\pi_{12}} \end{pmatrix}$$

The average expectation deviation over all label sets is thus

$$\mathbb{E}_{\mathcal{L}\sim P_\pi}[\Delta\mathbb{E}_{\mathcal{L}}] = \begin{pmatrix} \frac{\pi_{12}^2(e_1-e_{12})^2}{\pi_1+\pi_{12}} & -\frac{\pi_{12}^3(e_1-e_{12})(-e_2+e_{12})}{(\pi_1+\pi_{12})(\pi_2+\pi_{12})} \\ -\frac{\pi_{12}^3(e_1-e_{12})(-e_2+e_{12})}{(\pi_1+\pi_{12})(\pi_2+\pi_{12})} & \frac{\pi_{12}^2(-e_2+e_{12})^2}{\pi_2+\pi_{12}} \end{pmatrix},$$

and the expected variance of the sufficient statistics of the source emissions is

$$\mathbb{E}_{\mathcal{L}\sim P_\pi}\Big[\mathbb{V}_{X\sim P_{\mathcal{L},\boldsymbol{\theta}^G}}\Big[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{cross}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\Big]\Big] = \begin{pmatrix} \pi_1 v_1 + \pi_{12} v_{12} & \pi_{12} v_{12} \\ \pi_{12} v_{12} & \pi_2 v_2 + \pi_{12} v_{12} \end{pmatrix}.$$

With the values given in Table A.4, the expected generalized Fisher information matrix over all label sets is given by

$$\mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}] = \begin{pmatrix} (\pi_1 + \pi_{12})\hat{v}_1 & 0 \\ 0 & (\pi_2 + \pi_{12})\hat{v}_2 \end{pmatrix}$$

Hence, according to Theorem 5, the covariance matrix for the estimator $\hat{\boldsymbol{\theta}}^{cross}$ is given by

$$\Sigma_{\boldsymbol{\theta}}^{cross} = \begin{pmatrix} v_{\theta,11} & v_{\theta,12} \\ v_{\theta,12} & v_{\theta,22} \end{pmatrix} \tag{A.22}$$

with

$$\begin{aligned} v_{\theta,11} &= \frac{\pi_{12}^2 \left(e_1 - e_{12}\right)^2}{\left(\pi_1 + \pi_{12}\right)^3 \hat{v}_1^2} + \frac{\pi_1 v_1 + \pi_{12} v_{12}}{\left(\pi_1 + \pi_{12}\right)^2 \hat{v}_1^2} \\ v_{\theta,11} &= \frac{\pi_{12}^3 \left(e_1 - e_{12}\right)\left(e_2 - e_{12}\right)}{\left(\pi_1 + \pi_{12}\right)^2 \hat{v}_1 \left(\pi_2 + \pi_{12}\right)^2 \hat{v}_2} + \frac{\pi_{12} v_{12}}{(\pi_1 + \pi_{12})\hat{v}_1(\pi_2 + \pi_{12})\hat{v}_2} \\ v_{\theta,22} &= \frac{\pi_{12}^2 \left(e_2 - e_{12}\right)^2}{(\pi_2 + \pi_{12})^3 \hat{v}_2^2} + \frac{\pi_2 v_2 + \pi_{12} v_{12}}{(\pi_2 + \pi_{12})^2 \hat{v}_2^2} \ , \end{aligned}$$

where the estimated variances $\hat{v}_1$ and $\hat{v}_2$ are defined in Eq. 6.75.

To compute the variance of the estimator $\hat{\boldsymbol{\beta}}^{cross}$, we apply a Taylor approximation as in Eq. A.20. Hence we get

$$\Sigma_{\beta}^{cross} = \begin{pmatrix} v_{\beta,11} & v_{\beta,12} \\ v_{\beta,12} & v_{\beta,22} \end{pmatrix} \tag{A.23}$$

with

$$\begin{aligned} v_{\beta,11} &= \frac{v_1^2}{\hat{v}_1^2} \left(\frac{\pi_{12}^2 \left(\beta_1 - \beta_{12}\right)^2}{\left(\pi_1 + \pi_{12}\right)^3} + \frac{\pi_1 \beta_1 \left(1 - \beta_1\right) + \pi_{12} \beta_{12} \left(1 - \beta_{12}\right)}{\left(\pi_1 + \pi_{12}\right)^2} \right) \\ v_{\beta,12} &= \frac{v_1 v_2}{\hat{v}_1 \hat{v}_2} \left(\frac{\pi_{12}^3 \left(\beta_1 - \beta_{12}\right)\left(\beta_2 - \beta_{12}\right)}{\left(\pi_1 + \pi_{12}\right)^2 \left(\pi_2 + \pi_{12}\right)^2} + \frac{\pi_{12} \beta_{12} \left(1 - \beta_{12}\right)}{\left(\pi_1 + \pi_{12}\right)\left(\pi_2 + \pi_{12}\right)} \right) \\ v_{\beta,22} &= \frac{v_2^2}{\hat{v}_2^2} \left(\frac{\pi_{12}^2 \left(\beta_2 - \beta_{12}\right)^2}{\left(\pi_2 + \pi_{12}\right)^3} + \frac{\pi_2 \beta_2 \left(1 - \beta_2\right) + \pi_{12} \beta_{12} \left(1 - \beta_{12}\right)}{\left(\pi_2 + \pi_{12}\right)^2} \right) \end{aligned}$$

Mean-Squared Error of the Estimator. Since the estimator obtained by $\mathcal{M}_{cross}$ is not consistent, the mean squared error of the estimator consists of the bias and the variance. The bias of the estimators for the first and second source are given by

$$\Delta_1 := \hat{e}_1 - e_1 = \frac{\pi_{12}}{\pi_1 + \pi_{12}} \beta_2 (1 - \beta_1)$$
$$\Delta_2 := \hat{e}_2 - e_2 = \frac{\pi_{12}}{\pi_2 + \pi_{12}} \beta_1 (1 - \beta_2) \ .$$

Inserting these bias terms and the average variance into the bias-variance decomposition (Eq. 2.13) yields the expression given in the lemma.

Deconvolutive Training

Mean Value of the Estimator. The expectation values for the source emissions given single-label observations are given in Table A.5. For the label set $\mathcal{L} = \{1, 2\}$, the setting of Boolean random variables with an OR combination function is particular insofar as the observation $X = 0$ reveals total information about the emissions of all sources in the label set:

$$\boldsymbol{\Xi}|_{X=0,\mathcal{L}=\{1,2\}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(0,\{1,2\}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

If $X = 1$, at least one of the source emissions must be 1, while no information can be obtained about the other one. Using Eq. 6.74, the expected value of the sufficient statistics in the case $X = 1$ and $\mathcal{L} = \{1, 2\}$ is

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(1,\{1,2\}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \\ &= \frac{1}{\hat{e}_{12}} \left((1 - \hat{\beta}_1)\hat{\beta}_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \hat{\beta}_1 (1 - \hat{\beta}_2) \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \hat{\beta}_1 \hat{\beta}_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) \\ &= \frac{1}{\hat{e}_{12}} \begin{pmatrix} \hat{\beta}_1 \\ \hat{\beta}_2 \end{pmatrix} \end{aligned}$$

Combining the two cases, we get the values indicated in Table A.5.

The criterion function states that the parameters are chosen such that

$$\pi_1 \begin{pmatrix} e_1 \\ \hat{e}_2 \end{pmatrix} + \pi_2 \begin{pmatrix} \hat{e}_1 \\ e_2 \end{pmatrix} + \pi_{12} \frac{e_{12}}{\hat{e}_{12}} \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \end{pmatrix}$$

Solving for $\hat{e}_1$ and $\hat{e}_2$ and using $e_{12} = e_1 + e_2 - e_1 e_2$, $\hat{e}_{12} = \hat{e}_1 + \hat{e}_2 - \hat{e}_1 \hat{e}_2$ and $\pi_1 + \pi_2 + \pi_{12} = 1$, we find that the estimators are consistent:

$$\hat{e}_1 = e_1 \qquad \hat{e}_2 = e_2 \ . \tag{A.24}$$

Quantity	$\mathcal{L} = \{1\}$	$\mathcal{L} = \{2\}$	$\mathcal{L} = \{1, 2\}$
$\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} X \\ \hat{e}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ X \end{pmatrix}$	$X \frac{1}{\hat{e}_{12}} \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \end{pmatrix}$
$\mathbb{E}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}G}} \left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\mathcal{L}),\hat{\boldsymbol{\theta}}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \right]$	$\begin{pmatrix} e_1 \\ \hat{e}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{e}_1 \\ e_2 \end{pmatrix}$	$\frac{e_{12}}{\hat{e}_{12}} \begin{pmatrix} \hat{e}_1 \\ \hat{e}_2 \end{pmatrix}$

Table A.5: Quantities used to determine the asymptotic behavior of the $\mathcal{M}_{deconv}$ inference technique.

Quantity	$\mathcal{L} = \{1\}$	$\mathcal{L} = \{2\}$
$\mathbb{V}_{X \sim P_{\mathcal{L},\boldsymbol{\theta}}} \left[\mathbb{E}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\mathcal{L}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] \right]$	$\begin{pmatrix} \hat{v}_1 & 0 \\ 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 \\ 0 & \hat{v}_2 \end{pmatrix}$
$\mathbb{V}_{\boldsymbol{\Xi} \sim P^{deconv}_{(X,\mathcal{L}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]$	$\begin{pmatrix} 0 & 0 \\ 0 & \hat{v}_2 \end{pmatrix}$	$\begin{pmatrix} \hat{v}_1 & 0 \\ 0 & 0 \end{pmatrix}$
$\mathcal{I}_{\mathcal{L}}$	$-\begin{pmatrix} \hat{v}_1 & 0 \\ 0 & 0 \end{pmatrix}$	$-\begin{pmatrix} 0 & 0 \\ 0 & \hat{v}_2 \end{pmatrix}$

Table A.6: Variances and generalized Fisher information matrices for single label observations used to determine the asymptotic behavior of the $\mathcal{M}_{deconv}$ inference technique. The consistency of the estimators, i.e. $\hat{\theta} = \theta$, as found in Eq. A.24, is used to simplify the notation. The corresponding values for observations with label set $\mathcal{L} = \{1, 2\}$ are derived in the text.

As a consequence, the short-hands e_{12}, v_1, v_2 and v_{12} are consistent as well. In the remainder of this analysis, we therefore no longer distinguish between estimators and their true value.

Variance of the Estimator. For single-label data, the computations of the variances is straight forward, the values are given in Table A.6. For observations X with multiple label $\mathcal{L} = \{1, 2\}$, the variance is found by

enumerating all possible values of the emission vector $\boldsymbol{\Xi}$:

$$\mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(0,\{1,2\}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(1,\{1,2\}),\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})] = \frac{1}{(e_1+e_2-e_1e_2)^2}\begin{pmatrix} e_2v_1 & -v_1v_2 \\ -v_1v_2 & e_1v_2 \end{pmatrix}$$

This result again captures the particularities of the Boolean OR as combination function. Consider the variance of the estimator for Ξ_1:

$$\mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(0,\{1,2\}),\boldsymbol{\theta}}}[\Xi_1] = 0 \qquad \mathbb{V}_{\boldsymbol{\Xi}\sim P^{deconv}_{(1,\{1,2\}),\boldsymbol{\theta}}}[\Xi_1] = \frac{e_2v_1}{(e_1+e_2-e_1e_2)^2} \ .$$

The observation $X = 0$ reveals complete information about both source emissions. If $X = 1$, the expected emission Ξ_2 of the second source highly influences the amount of information that can be derived from the observation about Ξ_1: If the second source has a high probability to emit a 1, i.e. if e_2 is high, the variance of Ξ_1 is only slightly reduced as compared to the case where we have no observation, as information about Ξ_1 can only be obtained in the rare case when $\Xi_2 = 0$. In contrast, if the second source emits a 0 in most of the cases, e_2 is low, the variance of Ξ_1 is low, as Ξ_1 must be 1 in most of the cases when $X = 1$ is observed — otherwise, $X = \Xi_1 \vee \Xi_2$ would not be 1. The same is true for the variance of Ξ_2.

For the generalized Fisher information matrix of observations with label set $\mathcal{L} = \{1, 2\}$, we get

$$\mathcal{I}_{\{1,2\}} = \begin{pmatrix} v_1\left(\frac{e_2}{e_{12}} - 1\right) & -\frac{v_1v_2}{e_{12}} \\ -\frac{v_1v_2}{e_{12}} & v_2\left(\frac{e_1}{e_{12}} - 1\right) \end{pmatrix}$$

The expected Fisher information matrix over all label sets is

$$\mathbb{E}_{\mathcal{L}}[\mathcal{I}_{\mathcal{L}}] = -\begin{pmatrix} v_1\left(\pi_1 + \pi_{12}\left(1 - \frac{e_2}{e_{12}}\right)\right) & \pi_{12}\frac{v_1v_2}{e_{12}} \\ \pi_{12}\frac{v_1v_2}{e_{12}} & v_2\left(\pi_2 + \pi_{12}\left(1 - \frac{e_1}{e_{12}}\right)\right) \end{pmatrix}$$

To compute the variance of the expectation values over different observations with label set $\mathcal{L} = \{1, 2\}$, we enumerate all possible situations and get

$$\mathbb{V}_{X\sim P^{deconv}_{\{1,2\}),\boldsymbol{\theta}}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right] = \begin{pmatrix} v_1(1-e_2)\frac{e_1}{e_{12}} & \frac{v_1v_2}{e_{12}} \\ \frac{v_1v_2}{e_{12}} & v_2(1-e_1)\frac{e_2}{e_{12}} \end{pmatrix}$$

Note that, if either $e_1 = 1$ or $e_2 = 1$, the variance vanishes, as in such a setting, the observation is $X = 1$ for sure.

Using the values in Table A.6, the variance of the expected values is

$$\mathbb{E}_{\mathcal{L}}\left[\mathbb{V}_{X\sim P^{deconv}_{\mathcal{L},\boldsymbol{\theta}}}\left[\mathbb{E}_{\boldsymbol{\Xi}\sim P^{deconv}_{D,\boldsymbol{\theta}}}[\boldsymbol{\phi}(\boldsymbol{\Xi})]\right]\right] = \begin{pmatrix} \left(\pi_1 + \frac{e_1(1-e_2)}{e_{12}}\right) v_1 & \pi_{12} v_1 v_2 \\ \pi_{12} v_1 v_2 & \left(\pi_2 + \frac{e_2(1-e_1)}{e_{12}}\right) v_2 \end{pmatrix} .$$

Finally, the covariance matrix $\Sigma_{\boldsymbol{\theta}}$ for the estimator $\boldsymbol{\theta}$ is given by

$$\Sigma^{deconv}_{\boldsymbol{\theta}} = \begin{pmatrix} v_{\theta,11} & v_{\theta,12} \\ v_{\theta,12} & v_{\theta,22} \end{pmatrix}$$

$$\text{with} \quad v_{\theta,11} = \frac{\pi_2 e_{12} + \pi_{12} e_2(1-e_1)}{v_1\left(\pi_{12}(e_1\pi_2(1-e_2) + \pi_1 e_2(1-e_1)) + \pi_1\pi_2 e_{12}\right)}$$

$$v_{\theta,12} = -\frac{\pi_{12}}{\pi_{12}(\pi_1 e_2(1-e_1) + e_1\pi_2(1-e_2)) + \pi_1\pi_2 e_{12}}$$

$$v_{\theta,22} = \frac{\pi_1 e_{12} + \pi_{12} e_1(1-e_2)}{v_2\left(\pi_{12}(e_1\pi_2(1-e_2) + \pi_1 e_2(1-e_1)) + \pi_1\pi_2 e_{12}\right)}$$

For the variance of the estimator $\boldsymbol{\beta}$, we use again Eq. A.20 and get

$$\Sigma^{deconv}_{\boldsymbol{\beta}} = \begin{pmatrix} v_{\beta,11} & v_{\beta,12} \\ v_{\beta,12} & v_{\beta,22} \end{pmatrix} \tag{A.25}$$

$$\text{with} \quad v_{\beta,11} = \frac{\pi_2 e_{12} + \pi_{12} e_2(1-e_1)}{\left(\pi_{12}(e_1\pi_2(1-e_2) + \pi_1 e_2(1-e_1)) + \pi_1\pi_2 e_{12}\right)} v_1$$

$$v_{\beta,12} = -\frac{\pi_{12}}{\pi_{12}(\pi_1 e_2(1-e_1) + e_1\pi_2(1-e_2)) + \pi_1\pi_2 e_{12}} v_1 v_2$$

$$v_{\beta,22} = \frac{\pi_1 e_{12} + \pi_{12} e_1(1-e_2)}{\left(\pi_{12}(e_1\pi_2(1-e_2) + \pi_1 e_2(1-e_1)) + \pi_1\pi_2 e_{12}\right)} v_2$$

Mean-Squared Error of the Estimator. The mean squared error is the average of the diagonal elements of the covariance matrix. Using the values in Eq. A.25, we get the expression given in the theorem. □

A.4 Lemmas in Chapter 7

Proof. **Lemma 6.** The independence implies

$$p_{12}(\xi_1, \xi_2|\theta_1, \theta_2) = p_1(\xi_1|\theta_1) \cdot p_2(\xi_2|\theta_2) ,$$

thus

$$\frac{\partial^2 p_{12}(\xi_1,\xi_2|\theta_1,\theta_2)}{\partial\theta_1\partial\theta_2} = \frac{\partial \dot{p_1}(\xi_1|\theta_1)}{\partial\theta_2} p_2(\xi_2|\theta_2) + \dot{p_1}(\xi_1|\theta_1)\dot{p_2}(\xi_2|\theta_2) \ . \quad \text{(A.26)}$$

Since $p_1(\cdot)$ is independent of θ_2, the first summand is zero. If θ_1 has the maximum likelihood value, $\dot{p_1}(\xi_1|\theta_1) = 0$, and the sum is equal to zero. □

Proof. **Lemma 7.** Using the notation of Section 7.1.3, the log-likelihood of a data item $D_n = (x_n, \mathcal{L}_n)$ with binary label set $\mathcal{L}_n = \{s_1, s_2\}$ can be written as

$$\ell(\Theta; D_n) = \log P(\mathcal{L}_n) + \log\left(\int p_1(\xi) p_2(\mathrm{d}_\kappa(\xi, x_n))\,\mathrm{d}\xi\right) \ . \quad \text{(A.27)}$$

Recall that the likelihood of the parameter vector θ_1 is assumed to depend only on data items which contain s_1 in their label set, and that sources are assumed to emit i.i.d. samples. Thus, the derivative of the log-likelihood of the data set $\mathbf{D}$ with respect to θ_{1,c_1} can be written as

$$\frac{\partial \ell(\Theta;\mathbf{D})}{\partial\theta_{1,c_1}} = \sum_{\substack{n=1 \\ \mathcal{L}_n=\{s_1\}}}^{N} \frac{\partial \ell(\Theta, D_n)}{\partial\theta_1, c_1} + \sum_{\substack{n=1 \\ \mathcal{L}_n=\{s_1,\nu_n\}}}^{N} \frac{\partial \ell(\Theta, D_n)}{\partial\theta_{1,c_1}} \ . \quad \text{(A.28)}$$

The first term on the right hand side of Equation A.28 accounts for single label data. By the definition of the inference procedure $\mathcal{M}$, the parameter vector θ_2 and the sum are independent. The second term describes the influence of data items with two labels. Again due to the assumption of $\mathcal{M}$, its derivative with respect to θ_{2,c_2} vanishes for all n with $\nu_n \neq s_2$. For the remaining n with $\mathcal{L}_n = \{s_1, s_2\}$, we have, using Lemma 6:

$$\sum_{\substack{n=1 \\ \mathcal{L}_n=\{s_1,s_2\}}}^{N} \frac{\partial}{\partial\theta_{2,c_2}} \left\{ \frac{\partial \ell(\Theta;\mathbf{D})}{\partial\theta_{1,c_1}} \right\} = 0 \ .$$

Deriving $\ell(\Theta; D_n)$ as defined in Eq. A.27 with respect to θ_{1,c_1} and interchanging the derivation and the integration yields

$$\sum_{\substack{n=1 \\ \mathcal{L}_n=\{s_1,s_2\}}}^{N} \frac{\partial}{\partial\theta_{2,c_2}} \left\{ \frac{\int \left(\frac{\partial p_1(\xi)}{\partial\theta_{1,c_1}}\Big|_{\theta_{1,c_1}=\hat{\theta}_{1,c_1}} \cdot p_2(\mathrm{d}_\kappa(\xi,x)) \right) \mathrm{d}\xi}{\int p_1(\xi) p_2(\mathrm{d}_\kappa(\xi,x))\,\mathrm{d}\xi} \right\} = 0 \ .$$

Applying the derivation with respect to θ_{2,c_2} and using the introduced notation, one gets Eq. 7.1. □

Proof. **Lemma 8.** The proof mainly consists of computing the Taylor series of all integrands around $\xi = 0$. The resulting polynomials are then integrated and the coefficients reordered.

The Taylor series of a function $f(x)$ around x_0 is defined as

$$T(f, x_0, x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k \ ,$$

where $f^{(k)}(x_0)$ is the k^{th} derivative of f evaluated at x_0. The k^{th} derivative of a product $f(x) = f_1(x) \cdot f_2(x)$ is given by the *Leibnitz's law* as

$$f^{(k)}(x) = \sum_{j=0}^{k} \binom{k}{j} f_1^{(j)}(x) \cdot f_2^{(k-j)}(x) \ .$$

The generalization of the chain rule for derivatives of higher order is given by *Faà di Bruno's formula*:

$$\frac{\partial^m}{\partial x^m} \left\{ f(\mathrm{d}_n(0)) \right\} = S_m(f, n) \ ,$$

with $S_m(f, n)$ defined in Eq. 7.8. The Taylor series of the four integrands are thus:

$$\begin{aligned}
\dot{p}_1(\xi) \cdot \dot{p}_2(\mathrm{d}_n(\xi)) &= \sum_{j=0}^{\infty} \frac{\xi^j}{j!} \sum_{m=0}^{j} \left\{ \binom{j}{m} \dot{p}_1^{(j-m)}(0) \cdot S_m(\dot{p}_2, n) \right\} \\
p_1(\xi) \cdot p_2(\mathrm{d}_n(\xi)) &= \sum_{j=0}^{\infty} \frac{\xi^j}{j!} \sum_{m=0}^{j} \left\{ \binom{j}{m} p_1^{(j-m)}(0) \cdot S_m(p_2, n) \right\} \\
p_1(\xi) \cdot \dot{p}_2(\mathrm{d}_n(\xi)) &= \sum_{j=0}^{\infty} \frac{\xi^j}{j!} \sum_{m=0}^{j} \left\{ \binom{j}{m} p_1^{(j-m)}(0) \cdot S_m(\dot{p}_2, n) \right\} \\
\dot{p}_1(\xi) \cdot p_2(\mathrm{d}_n(\xi)) &= \sum_{j=0}^{\infty} \frac{\xi^j}{j!} \sum_{m=0}^{j} \left\{ \binom{j}{m} \dot{p}_1^{(j-m)}(0) \cdot S_m(p_2, n) \right\} \ .
\end{aligned} \tag{A.29}$$

After integrating each of the two polynomials separately, multiplying and re-arranging terms, we get the expressions presented in the Lemma. □

Proof. **Lemma 9.** The proof is done by induction over the order of the derivative.

Base case: Equations 7.3 and A.29 show that setting $C^0_{lhs} = C^0_{rhs}$ and $C^1_{lhs} = C^1_{rhs}$ implies that all integration constants are equal to zero (unless we are willing to accept constraints on the probability densities and their derivatives). With this, the non-zero terms of C^i_{lhs} and C^i_{rhs} for $i = 2, 3$ are identical and thus do not allow to draw any conclusions on the value of the derivatives of the inverse combination function $\mathrm{d}_n(\cdot)$. $C^4_{lhs} - C^4_{rhs}$ is the first non-vanishing difference between the coefficients that contains a derivative of $\mathrm{d}_n(\cdot)$:

$$C^4_{lhs} - C^4_{rhs} = \frac{1}{12} \cdot c^{(1)}(0) \cdot \left(\dot{p}_1(0) \cdot p_1^{(1)}(0) - p_1(0) \cdot \dot{p}_1^{(1)}(0) \right) \cdot \sum_n \frac{\dot{p}_2(\mathrm{d}_n(0)) \cdot p_2^{(1)}(\mathrm{d}_n(0)) - \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \cdot p_2(\mathrm{d}_n(0))}{p(x_n)^2}. \tag{A.30}$$

Requiring the left-hand side to be zero implies that at least one factor on the right-hand side has to be zero. Requiring one of the two factors containing probability densities to be zero might be impossible or at least a very hard constraint on the family of distributions and its parameters[1]. The only factor in Equation A.30 which is independent of probability densities is $c^{(1)}(0)$. Therefore $C^4_{lhs} = C^4_{rhs}$ implies $c^{(1)}(0) = 0$.

Inductive Step: Assume that, for some integer $z \geq 1$, we have used the conditions $C^{i+3}_{lhs} = C^{i+3}_{rhs}$ for $i = 1, \ldots, z$ to derive $c^{(1)}(0) = \ldots = c^{(z)}(0) = 0$. Using this inductive claim, we show that $c^{(z+1)}(0) = 0$ follows from the condition $C^{z+3+1}_{lhs} = C^{z+3+1}_{rhs}$.

The expression for C^{z+4}_{lhs} can be derived from Eq. 7.3 with $\alpha = lhs$ and $i = z + 4$. Consider $C^l_{lhs,1}$ as defined in Eq. 7.4: Since $c^{(i)}(0) = 0$ for $i \leq z$ by the induction claim, $b_m(t, \dot{p}_2, n)$ is zero whenever there is a $i \leq z$ with $n_i > 0$. Nonzero contributions to $S_m(\dot{p}_2, n)$ are therefore only possible if either $m = 0$ or $m > z$. If $m = 0$, we have $b_0(t, \dot{p}_2, n) = \dot{p}_2(\mathrm{d}_n(0))$. In the second case, $z < m \leq l - 1$ and $l \leq z + 3$ limit the possible values of m to $m = z + 1$, $m = z + 2$ and $m = z + 3$. Hence, the only $t \in T_m$ leading to a

[1] Consider e.g. the univariate Gaussian distributions $p_1(z) = \mathcal{N}(\mu_1, \sigma_1^2)$. With $\dot{p}_1(z) = p_1(z) \cdot \frac{z-\mu_1}{\sigma_1^2}$, $p_1^{(1)}(z) = -p_1(z) \cdot \frac{z-\mu_1}{\sigma_1^2}$ and $\dot{p}_1^{(1)}(z) = p_1(z) \cdot \frac{(z-\mu_1)^2 - \sigma^2}{\sigma_1^4}$, we get $\dot{p}_1(0) \cdot p_1^{(1)}(0) - p_1(0) \cdot \dot{p}_1^{(1)}(0) = -[p_1(z)]^2 \frac{1}{\sigma_1^2}$. This expression is different from 0 for all parameters and z. A similar reasoning is applicable for the sum over n.

nonzero contributions are $t_m = 1$ and $t_i = 0$ for all $i \neq m$. Therefore,

$$S_m(p_2, n) = \begin{cases} \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \cdot c^{(m)}(0) & \text{if } z < m \leq z+3 \\ 0 & \text{otherwise.} \end{cases}$$

Now consider $C^l_{lhs,2}$, defined in Eq. 7.5: With the same argumentation as above, we see that all terms with $0 < m < z+1$ are zero. Similar reasonings allow to filter out non-zero contributions to C^{z+4}_{rhs}. Thus, elementary but lengthy calculations and separating different derivatives of $\mathrm{d}_n(\cdot)$ lead to:

$$\begin{aligned} C^{z+4}_{lhs} =& \sigma^{z+4}_{lhs} + \sum_n \frac{1}{p(x_n)^2} \Big(\dot{p}_1(0) \cdot \dot{p}_2(\mathrm{d}_n(0)) \cdot p_1^{(1)}(0) \cdot p_2^{(1)}(\mathrm{d}_n(0)) \\ & + p_1(0) \cdot p_2(\mathrm{d}_n(0)) \cdot \dot{p}_1^{(1)}(0) \cdot \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \Big) \cdot \frac{z+2}{(z+3)!} \cdot \mathrm{d}_n^{(z+1)}(0) \\ & + \sum_n \frac{1}{p(x_n)^2} \Big(\dot{p}_1(0) \cdot \dot{p}_2(\mathrm{d}_n(0)) \cdot p_1(0) \cdot p_2^{(1)}(\mathrm{d}_n(0)) \\ & + p_1(0) \cdot p_2(\mathrm{d}_n(0)) \cdot \dot{p}_1(0) \cdot \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \Big) \cdot \frac{1}{(z+3)!} \cdot \mathrm{d}_n^{(z+2)}(0) \\ C^{z+4}_{rhs} =& \sigma^{z+4}_{rhs} + \sum_n \frac{1}{p(x_n)^2} \Big(p_1(0) \cdot \dot{p}_2(\mathrm{d}_n(0)) \cdot \dot{p}_1^{(1)}(0) \cdot p_2^{(1)}(\mathrm{d}_n(0)) \\ & + \dot{p}_1(0) \cdot p_2(\mathrm{d}_n(0)) \cdot p_1^{(1)}(0) \cdot \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \Big) \cdot \frac{z+2}{(z+3)!} \cdot \mathrm{d}_n^{(z+1)}(0) \\ & + \sum_n \frac{1}{p(x_n)^2} \Big(p_1(0) \cdot \dot{p}_2(\mathrm{d}_n(0)) \dot{p}_1(0) \cdot p_2^{(1)}(\mathrm{d}_n(0)) \\ & + \dot{p}_1(0) \cdot p_2(\mathrm{d}_n(0)) \cdot p_1(0) \cdot \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \Big) \cdot \frac{1}{(z+3)!} \cdot \mathrm{d}_n^{(z+2)}(0) \ . \end{aligned}$$

σ^{z+4}_{lhs} and σ^{z+4}_{rhs} are sums over terms that do not contain any derivatives of $\mathrm{d}_n(\cdot)$. Re-arranging terms and changing summation orders, we get $\sigma^{z+4}_{lhs} = \sigma^{z+4}_{rhs}$.

Finally, the difference $C^{z+4}_{lhs} - C^{z+4}_{rhs}$ is:

$$\begin{aligned} & \frac{z+1}{2 \cdot (z+3)!} \cdot c^{(z+1)}(0) \\ & \cdot \sum_n \frac{1}{p(x_n)^2} \Big(\Big(p_1^{(1)}(0) \cdot \dot{p}_1(0) - p_1(0) \cdot \dot{p}_1^{(1)}(0) \Big) \cdot p_2^{(1)}(\mathrm{d}_n(0)) \cdot \dot{p}_2(\mathrm{d}_n(0)) \\ & + \Big(p_1(0) \cdot \dot{p}_1^{(1)}(0) - p_1^{(1)}(0) \cdot \dot{p}_1(0) \Big) \cdot p_2(\mathrm{d}_n(0)) \cdot \dot{p}_2^{(1)}(\mathrm{d}_n(0)) \Big) \ . \end{aligned}$$

As we do not want do put constraints on the source distributions, $c^{(z+1)}(0) = 0$ follows from $C_{lhs}^{z+4} = C_{rhs}^{z+4}$. This proves the induction step and concludes the proof of the Lemma. □

Proof. **Corollary 1.** The log-likelihood of the parameters Θ given D_n is

$$l(\Theta; D_n) = \log P(\mathcal{L}) + \log\left(\int \prod_{i=1}^{d_n} p_{k_n^{(i)}}(\xi_{k_n^{(i)}}|\theta_{k_n^{(i)}}) \cdot \delta_{\mathrm{c}_\kappa(\boldsymbol{\xi})=x_n} \,\mathrm{d}\boldsymbol{\xi}\right),$$

and the derivative of the parameter likelihood given the training set $\mathbf{D}$ is

$$\frac{\partial \ell(\Theta; \mathbf{D})}{\partial \theta_{1,c_1}} = \sum_{d=1}^{K} \sum_{\substack{n: d_n = d \\ s_1 \in \mathcal{L}_n}} \frac{\partial \ell(\Theta; D_n)}{\partial \theta_{1,c_1}} \ .$$

Proceeding as in the proof of Lemma 7, we get

$$\begin{aligned} &\sum_{d=1}^{K} \sum_{\substack{n: d_n = d \\ s_1 \in \mathcal{L}_n}} \frac{\int p(\xi_n^{(-1,2)}) \dot{p}_{s_1}(\xi_n^{(1)}) \dot{p}_{s_2}(\xi_n^{(2)}) \,\mathrm{d}\boldsymbol{\xi}_n}{p(x_n)^2} \\ &\quad = \sum_{d=1}^{K} \sum_{\substack{n: d_n = d \\ s_1 \in \mathcal{L}_n}} \frac{\int p(\xi_n^{(-1)}) \dot{p}_{s_1}(\xi_n^{(1)}) \,\mathrm{d}\boldsymbol{\xi}_n p(x_n) \cdot \int p(\xi_n^{(-2)}) \dot{p}_{s_2}(\xi_n^{(2)}) \,\mathrm{d}\boldsymbol{\xi}_n}{p(x_n)^2}, \end{aligned} \tag{A.31}$$

with the following definitions for a more compact notation:

$$\begin{aligned} p(\xi_n^{(-q)}) &:= \prod_{s \in \mathcal{L}_n \setminus \{s_q\}} p_s(\xi_n^{(s)}) \quad q = 1, 2 \\ p(\xi_n^{(-1,2)}) &:= \prod_{s \in \mathcal{L}_n \setminus \{s_1, s_2\}} p_s(\xi_n^{(s)}) \ . \end{aligned}$$

The factors $p(\xi_n^{(-1)})$, $p(\xi_n^{(-2)})$, and $p(\xi_n^{(-1,2)})$ are independent of ξ_1 and ξ_2 and carry through the integration (with respect to ξ_1 and ξ_2) and the Taylor series. The equality of the coefficients of the Taylor series implies again that the combination function is constant with respect to one argument, thus contradicting the assumption that the combination function is a bijection. □

Bibliography

[1] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 1994.

[2] Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716 – 723, 1974.

[3] Erin L. Allwein, Robert E. Schapire, and Yoram Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2001.

[4] Barry Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12:35–50, 1992.

[5] Jonathan Baxter. A model of inductive bias learning. *J. Artif. Int. Res.*, 12(1):149–198, 2000.

[6] Thomas Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans.*, 53:370–418, 1763.

[7] Shai Ben-David and Reba Schuller. Exploiting task relatedness for mulitple task learning. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Porceedings of the 16th Annual Conference on Computational Learning Theory*, volume 2777 of *Lecture Notes in Computer Science*, pages 567–580. Springer, 2003.

[8] Shai Ben-David, Ulrike von Luxburg, and Dávid Pál. A sober look at clustering stability. In G. Lugosi and H.U. Simon, editors, *Learning Theory, Proceedings of 19th Annual Conference on Learning Theory,*

COLT 2006, Pittsburgh, PA, USA, volume 4005 of *LNAI*, pages 5–19. Springer-Verlag, 2006.

[9] Asa Ben-Hur, Andre Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, 2002.

[10] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, Berlin, 2007.

[11] Matthew Boutell, Jiebo Luo, Xipeng Shen, and Christopher Brown. Learning multi-label scene classification. *Pattern Recognition*, pages 1757–1771, 2004.

[12] Alessandra R. Brazzale, Anthony Christopher Davison, and Nancy Reid. *Applied asymptotics: case studies in small-sample statistics*. Cambridge series on statistical and probabilistic mathematics. Cambridge University Press, 2007.

[13] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. The MIT Press, 1994.

[14] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, 2001.

[15] Richard P. Brent. *Algorithms for Minimisation without Derivatives*. Prentice Hall, 1972.

[16] Klaus Brinker, Johannes Fürnkranz, and Eyke Hüllermeier. A unified model for multilabel classification and ranking. In *Proceedings of the 17th European Conference on Artificial Intelligence*, pages 489–493, 2006.

[17] Michael Büchler. How good are automatic program selection features? *Hearing Review*, 9(84):50–54, 2001.

[18] Michael Büchler. *Algorithms for Sound Classification in Hearing Instruments*. PhD thesis, ETH Zurich, 2002.

[19] Michael Büchler, Silvia Allegro, Stefan Launer, and Norbert Dillier. Sound classification in hearing aids inspired by auditory scene analysis. *EURASIP J. Appl. Signal Process.*, 2005(1):2991–3002, 2005.

[20] Joachim M. Buhmann. Information theoretic model validation for clustering. In *International Symposium on Information Theory, Austin Texas*. IEEE, 2010.

[21] Joachim M. Buhmann and Hans Kühnel. Vector quantization with complexity costs. In *IEEE Trans on Information Theory*, volume 39, pages 1133–1145, 1993.

[22] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining*, 2(2), 1998.

[23] Kenneth P. Burnham and David R. Anderson. *Model selection and inference: a practical information-theoretic approach, 2nd ed.* Springer, New York, 2002.

[24] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[25] Amanda Clare and Ross D. King. Knowledge discovery in multi-label phenotype data. *Lecture Notes in Computer Science*, 2168:42–53, 2001.

[26] Alessandro Colantonio, Roberto Di Pietro, and Alberto Ocello. A cost-driven approach to role engineering. In *Proceedings of the 23rd ACM Symposium on Applied Computing, SAC '08*, volume 3, pages 2129–2136, Fortaleza, Ceará, Brazil, 2008.

[27] Alessandro Colantonio, Roberto Di Pietro, Alberto Ocello, and Nino Vincenzo Verde. A formal framework to elicit roles with business meaning in RBAC systems. In *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, SACMAT '09*, 2009.

[28] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287 – 314, 1994. Higher Order Statistics.

[29] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley Interscience, 2006.

[30] Harald Cramér. Contributions to the theory of statistical estimation. *Skand. Aktuarietids*, 29:85–94, 1946.

[31] Harald Cramér. *Mathematical methods of statistics.* Princeton University Press, 1999.

[32] Scott Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and L. Beck. Improving information retrieval with latent semantic indexing. In *Proceedings of 1988 annual meeting of the American Society for Information Science*, 1988.

[33] Arthur Dempster, Nan Laird, and Donald Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Statist. Soc. B*, 39(1):1–38, 1977.

[34] Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition.* Stochastic Modelling and Applied Probability. Springer, corrected edition, 1996.

[35] Thomas G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *J. of Articificial Intelligence Research*, 2:263–286, 1995.

[36] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[37] Tomas Dikk. Features for multi-class auitory scene analysis. Semester Thesis, ETH Zurich, 2006.

[38] Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support Vector Regression Machines. In *Advances in Neural Information Processing Systems 9, NIPS 1996*, pages 155–161. MIT Press, 1997.

[39] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition).* Wiley-Interscience, 2000.

[40] Sandrine Dudoit and Jane Fridlyand. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology*, 3(7), 2002.

[41] Anthony W. F. Edwards. *Likelihood.* The Johns Hopkins University Press, 1992.

[42] Andre Elisseeff and Jason Weston. Kernel methods for multi-labelled classification and categorical regression problems. In *Proceedings of NIPS 2002*, 2002.

[43] Alina Ene, William Horne, Nikola Milosavljevic, Prasad Rao, Robert Schreiber, and Robert E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *Symp on Access Control Models and Technologies*, pages 1–10, 2008.

[44] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, and Ramaswamy Chandramouli. Proposed NIST standard for role-based access control. *ACM Trans Inf Syst Secur*, 4(3):224–274, 2001.

[45] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals Eugen.*, 7:179–188, 1936.

[46] Mario Frank, Joachim M. Buhmann, and David Basin. On the definition of role mining. In *SACMAT '10: Proceeding of the 15th ACM symposium on Access control models and technologies*, pages 35–44, New York, NY, USA, 2010. ACM.

[47] Mario Frank, Andreas P. Streich, David Basin, and Joachim M. Buhmann. A probabilistic approach to hybrid role mining. In *16th ACM Conference on Computer and Communications Security (CCS 2009)*. ACM, 2009.

[48] Brendan J. Frey, Li Deng, Alex Acero, and Trausti Kristjansson. Algonquin: Iterating Laplace's method to remove multiple types of acoustic distortion for robust speech recognition. In *EUROSPEECH, 2001*, pages 901–904, 2001.

[49] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo L. Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2):133–153, 2008.

[50] Zoubin Ghahramani and Michael I. Jordan. Factorial hidden markov models. *Machine Learning*, 29(2-3):245–275, 1997.

[51] Nadia Ghamrawi and Andrew McCallum. Collective multi-label classification. In *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM)*, pages 195–200, 2005.

[52] Thomas L. Griffiths and Zoubin Ghahramani. Infinite latent feature models and the indian buffet process. In *Conf on Neural Information Processing Systems*, pages 475–482, 2005.

[53] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[54] Trevor Hastie, Andreas Buja, and Robert Tibshirani. Penalized discriminant analysis. *Annals of Statistics*, 23:73–102, 1995.

[55] Trevor Hastie and Robert Tibshirani. Discriminant analysis by gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:155–176, 1996.

[56] Trevor Hastie, Robert Tibshirani, and Andreas Buja. Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association*, 89:1255–1270, 1993.

[57] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Second Edition.* Springer, 2009.

[58] John R. Hershey, Steven J. Rennie, Peder A. Olsen, and Trausti T. Kristjansson. Super-human multi-talker speech recognition: A graphical modeling approach. *Comput. Speech Lang.*, 24(1):45–66, 2010.

[59] Thomas Hofmann. *Data Clustering and Beyond - A Deterministic Annealing Framework for Exploratory Data Analysis.* PhD thesis, University of Bonn, Germany, 1997.

[60] Daniel Hsu, Sham M. Kakade, John Langford, and Tong Zhang. Multi-label prediction via compressed sensing. `http://arxiv.org/abs/0902.1284`, 2009.

[61] Eyke Hüllermeier, Johannes Fürnkranz, Weiwei Cheng, and Klaus Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172:1897–1917, 2008.

[62] Anil K. Jain, M. N. Murty, and Patrick J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[63] Edwin T. Jaynes. Information Theory and Statistical Mechanics. *Phys. Rev.*, 106(4):620–630, 1957.

[64] Edwin T. Jaynes. Information Theory and Statistical Mechanics. II. *Phys. Rev.*, 108(2):171–190, 1957.

[65] Edwin T. Jaynes. Prior probabilities. *IEEE Transactions on Systems Science and Cybernetics*, 4(3):227 –241, 1968.

[66] Rong Jin and Zoubin Ghahramani. Learning with multiple labels. In *Proceedings of the Sixteenth Conference on Neural Information Processing Systems (NIPS)*, pages 897–904, 2002.

[67] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML*, 1998.

[68] Ata Kabán and Ella Bingham. Factorisation and denoising of 0-1 data: A variational approach. *Neurocomputing*, 71(10-12):2291 – 2308, 2008.

[69] Kentaro Kawai and Yoshimasa Takahashi. Identification of the dual action antihypertensive drugs using tfs-based support vector machines. *Chem-Bio Informatics Journal*, pages 41–51, 2009.

[70] Paul Kendrick, Trevor J. Cox, Francis F. Li, Yonggang Zhang, and Jonathon A. Chambers. Monaural room acoustic parameters from music and speech. *The Journal of the Acoustical Society of America*, 124(1):278–287, 2008.

[71] Paul Kendrick, Francis F Li, Trevor J. Cox, Yonggang Zhang, and Jonathon A. Chambers. Blind estimation of reverberation parameters for non-diffuse rooms. *Acta Acustica united with Acustica*, 93(11):760–770, 2007.

[72] Trausti Kristjansson, Hagai Attias, and John Hershey. Single microphone source separation using high resolution signal reconstruction. In *Proceedings of ICASSP'04*, pages 817–820, 2004.

[73] Martin Kuhlmann, Dalia Shohat, and Gerhard Schimpf. Role mining - revealing business roles for security administration using data mining technology. In *Symp on Access Control Models and Technologies*, pages 179–186, 2003.

[74] Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[75] Tilman Lange, Mikio Braun, Volker Roth, and Joachim M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299–1323, 2004.

[76] Erich L. Lehmann and George Casella. *Theory of point estimation*. Springer, 1998.

[77] Tao Li and Mitsunori Ogihara. Detecting emotion in music. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 239–240, Washington D.C., USA, 2003.

[78] Percy Liang and Michael I. Jordan. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 584–591, New York, NY, USA, 2008. ACM.

[79] David J.C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[80] Oded Maron and Aparna Lakshmi Ratan. Multiple-instance learning for natural scene classification. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 341–349, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[81] Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by EM. In *Proceedings of NIPS*, 1999.

[82] Neri Merhav, Dongning Guo, and Shlomo Shamai. Statistical physics of signal estimation in Gaussian noise: Theory and examples of phase transitions. *IEEE Transactions on Information Theory*, 56(3):1400–1416, 2010.

[83] Marc Mézard and Andrea Montanari. *Information, Physics and Computation*. Oxford University Press, Oxford, 2008.

[84] Pauli Miettinen, Taneli Mielikäinen, Aris Gionis, Gautam Das, and Heikki Mannila. The Discrete Basis Problem. In *Proc of Principles and Practice of Knowledge Discovery in Databases*, pages 335–346, 2006.

[85] Ian Molloy, Ninghui Li, Yuan (Alan) Qi, Jorge Lobo, and Luke Dickens. Mining roles with noisy data. In *SACMAT '10: Proceeding of the 15th ACM symposium on Access control models and technologies*, pages 45–54, New York, NY, USA, 2010. ACM.

[86] Alan V. Oppenheim and Roland W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, 1998.

[87] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[88] Louis C.W. Pols. *Spectral analysis and identification of Dutch vowels in monosyllabic words*. PhD thesis, Free University of Amsterdam, 1966.

[89] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Readings in speech recognition*, pages 267–296. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[90] Calyampudi R. Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bull. Calcutta Math. Soc.*, 37:81–91, 1945.

[91] Calyampudi R. Rao. *Linear Statistical Inference and Its Applications*. John Wiley & Sons Inc, 1973.

[92] Rama Ratnam, Douglas L. Jones, Bruce C. Wheeler, William D. O'Brien Jr, Charissa R. Lansing, and Albert S. Feng. Blind estimation of reverberation time. *The Journal of the Acoustical Society of America*, 114(5):2877–2892, 2003.

[93] Jesse Read, Bernhard Pfahringer, and Geoff Holmes. Multi-label classification using ensembles of pruned sets. In *ICDM '08: Eighth IEEE International Conference on Data Mining*, pages 995 –1000, 2008.

[94] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases*, pages 254–269, 2009.

[95] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.

[96] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proc. of the IEEE*, pages 2210–2239, 1998.

[97] Volker Roth and Bernd Fischer. Improved functional prediction of proteins by learning kernel combinations in multilabel settings. *BMC Bioinformatics*, 8(2):12, 2007.

[98] Robert E. Schapire and Yoram Singer. Boostexter A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[99] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[100] Malcolm Slaney. Auditory toolbox. version 2, technical report, 1998.

[101] Steven T. Smith. Covariance, subspace, and intrinsic cramér-rao bounds. *Signal Processing, IEEE Transactions on*, 53(5):1610 – 1630, 2005.

[102] Andreas P. Streich and Joachim M. Buhmann. Classification of multi-labeled data: A generative approach. In *Europ Conf on Machine Learning*, pages 390–405, 2008.

[103] Andreas P. Streich and Joachim M. Buhmann. Ignoring co-occurring sources in learning from multi-labeled data leads to model mismatch. In *MLD09: ECML/PKDD 2009 Workshop on Learning from Multi-Label Data*, 2009.

[104] Andreas P. Streich, Manuela Feilner, Alfred Stirnemann, and Joachim M. Buhmann. Sound field indicators for hearing activity and reverberation time estimation in hearing instruments. In *AES 12th Convention*, 2010.

[105] Andreas P. Streich, Mario Frank, David Basin, and Joachim M. Buhmann. Multi-assignment clustering for boolean data. In *Proceedings of the 26th International Conference on Machine Learning*, pages 969–976. Omnipress, 2009.

[106] Robert Tibshirani, Guenther Walther, and Trevor Hastie. Estimating the number of clusters in a dataset via the gap statistic. *Journal of the Royal Statistical Society series B*, 63:411–423, 2000.

[107] Grigorios Tsoumakas and I. Katakis. Multi label classification: An Overview. *International Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.

[108] Grigorios Tsoumakas and Ioannis Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European Conference on Machine Learning (ECML 2007)*, pages 406–417, Warsaw, Poland, 2007.

[109] George Tzanetakis and Perry Cook. Marsyas: a framework for audio analysis. *Organised Sound*, 4(03):169–175, 2000.

[110] George Tzanetakis and Georg Essl. Automatic musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, pages 293–302, 2001.

[111] Naonori Ueda and Kazumi Saito. Parametric mixture model for multitopic text. *Systems and Computers in Japan*, 37(2):56–66, 2006.

[112] Jaideep Vaidya, Vijay Atluri, and Qi Guo. The Role Mining Problem: Finding a minimal descriptive set of roles. In *Symp on Access Control Models and Technologies*, pages 175–184, 2007.

[113] Jaideep Vaidya, Vijay Atluri, and Qi Guo. The Role Mining Problem: Finding a minimal descriptive set of roles. In *The Twelth ACM Symposium on Access Control Models and Technologies*, pages 175–184, Sophia Antipolis, France, 2007. ACM.

[114] Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[115] Aad W. van der Vaart. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1998.

[116] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, Hoboken, NJ, USA, 1998.

[117] Alexander Vezhnevets and Joachim M. Buhmann. Towards weakly supervised semantic segmentation by means of multiple instance and multitask learning. In *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[118] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305, 2008.

[119] Frank Wood. A non-parametric Bayesian method for inferring hidden causes. In *Conf on Uncertainty in Artificial Intelligence*, pages 536–543, 2006.

[120] Kai Yu, Shipeng Yu, and Volker Tresp. Multi-label informed latent semantic indexing. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–265, New York, NY, USA, 2005. ACM.

[121] Min-Ling Zhang and Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. *Proceedings of the IEEE International Conference on Granular Computing*, 2:718–721, 2005.

[122] Shenghuo Zhu, Xiang Ji, Wei Xu, and Yihong Gong. Multi-labelled classification using maximum entropy method. In *Proceedings of SIGIR*, 2005.

Curriculum Vitae

Name	Andreas Peter Streich
Date of birth	April 29, 1980, in Berne, Switzerland
08/1996 - 06/2000	High school: Gymnasium Interlaken
06/2000	**Matura** at Gymnasium Interlaken
06/2000 - 10/2000	Military Service
10/2000 - 10/2002	Basic studies (∼B.Sc.) in computer science ETH Zurich
10/2002 - 10/2005	Advanced studies in computer science ETH Zurich
09/2003 - 04/2004	Advanced studies in theoretical physics École polytechnique, France
10/2005	**Diplom (M.Sc.) in Computer Science** ETH Zurich
02/2006 - 08/2010	**Doctoral studies at ETH Zürich**

Acknowledgements

The four years I was working on this thesis have been a stimulating journey of discovery. I was very fortunate to be accompanied by many persons who have inspired, criticized and motivated me during that time.

First of all, I'm much obliged to my thesis supervisor Joachim Buhmann. His suggestions have been the starting point for several sub-projects. With his critical comments, he taught me to ask the right questions and to make my statements as precise as possible.

I thank our industrial partner Phonak for financial support and for providing me with some of the data sets used in this thesis. The discussions with our contact persons Sascha Korl and Manuela Feilner as well as with Stefan Launer have lead to new inspiration and have given a valuable focus onto the practical application. The development of new features with Alfred Stirnemann was a formidable, application-driven task.

Yvonne Moh and Christian Sigg who have been working in the same cooperation with Phonak. Namely in the early phase of this thesis, the joint discussions were very helpful to identify relevant research problems. Tomas Dikk has joined the audio group later. I would like to thank all three for their valuable input. A thank you also to Volker Roth who coached us in the early phase of the project.

I thank Mario Frank for the fruitful collaboration in the later phase of my thesis. I enjoyed working with him, our discussions and mutual enrichment were a big benefit to both of us.

Furthermore, I would like to thank all members of the machine learning laboratory for helpful questions and suggestions on my work. Cheng Soo Ong helped me to concretize research questions and has contributed a lot to an open-minded and cooperative spirit within the research group. Thomas Fuchs supported me with useful hints on the use of the programming language R, and Peter Schüffler helped with all sorts of computer problems. My

gratitude goes also to Patrick Pletscher and Sharon Wulff for their company in the "Super-Kondi", and to Verena Kaynig and Ludwig Busse for encouraging coffee breaks. Rita Klute has taken care of all administrative matters in a very kind way.

Furthermore, I would like to thank all my friends for their support over the last years. I thank Oliver von Rotz, Nils Weidmann and Martin Hood from the Academic Alpine Club Zurich, Thomas Reinthaler, and the Gaswerk climbing crew with Basil Gysin and Niculin Saratz for enjoyable hours and weekends of mountain sports. I also thank my colleagues from the diploma studies, in particular Tobias Gysi, for entertaining hours outside academia.

Last but not least, a very special thank you goes to my family for their constant support through all ups and downs during the time of my studies.