Variational Inference in Probabilistic Submodular Models
Josip Djolonga

Diss.-No. ETH 25182
VARIATIONAL INFEERENCE IN PROBABILISTIC SUBMODULAR MODELS

A dissertation submitted to attain the degree of DOCTOR OF SCIENCES OF ETH ZURICH (Dr. sc. ETH Zurich)

presented by

JOSIP DJOLONGA
MSc ETH CS

born on 7 August 1989
citizen of Republic of Macedonia

accepted on the recommendation of

Prof. Dr. A. Krause, examiner
Prof. Dr. F. Bach, co-examiner
Prof. Dr. J. Bilmes, co-examiner

2018
Josip Djolonga: Variational Inference in Probabilistic Submodular Models, © 2018

DOI: 10.3929/ethz-a-
ABSTRACT

Submodular functions are increasingly used in machine learning to capture notions of sparsity, regularity and diversity. Since both the data used to define them and the decisions made after optimizing them are subject to uncertainty and noise, it is natural to examine them from a probabilistic perspective. We thus have to go beyond the classically studied optimization problems that yield single solutions, and work towards quantifying the inherent statistical uncertainty. In this thesis, we define and analyze several classes of distributions, which we collectively refer to as probabilistic submodular models. These models are motivated by optimization problems that leverage submodularity in various ways, and have been used to great effect in machine learning and computer vision.

The main problem that we study is that of performing probabilistic inference in these models. This task is not only provably hard to approximately solve, but can be also challenging for classical approximate inference techniques. The distributions are typically complex and can include terms that model dependencies between thousands of variables. In this thesis we undertake a variational divergence minimization approach, and approximate these distributions with simpler ones that are as close as possible to them. As a measure of distributional distance we use the inclusive and exclusive infinite Rényi divergences, which we show how to efficiently minimize by leveraging the models’ submodularity. These distances not only have computational benefits, but it is also well-understood what kind of approximations they strive for, and what kind of trade-offs one makes when minimizing them. For specific model classes we not only prove approximation guarantees, but also characterize the best approximate distributions — for example, we relate them to classically studied problems and show that they share a mode with the target distribution. Moreover, throughout all chapters we experimentally showcase the efficacy of the proposed techniques on synthetic problems and real-world data. In particular, we apply our methods to semantic image segmentation and item set recommendation.

The first family we consider are the log-sub- and log-supermodular models, which are the natural Gibbs measures corresponding to the unconstrained submodular minimization and maximization problems. We then introduce the class of mixed submodular models, that can capture any dis-
crete distribution. For these models we develop inference algorithms using an expectation propagation framework. Finally, we discuss the family of cooperative graphical models, which are motivated by the cooperative cut problem in computer vision. To perform approximate inference we combine concepts from the theory of submodular functions with variational inference methods for exponential families.

In the last chapter we tackle the problem of learning conditional log-supermodular models. Due to the hardness of the exact inference problems, these models cannot be learned using the standard maximum likelihood approach. We hence learn them in a framework that leverages our approximation schemes. Specifically, we learn models such that the closest approximate distribution yields a high likelihood to the observed data. This results in a complex bi-level optimization problem, which we efficiently solve by carefully utilizing both the optimization and polyhedral properties of submodular functions.


Die erste von uns betrachtete Familie von Modellen, sind die log-sub- und log-supermodularen Modelle, die den Gibbs-Maßen der bedin-
gungslosen submodularen Minimierungs- und Maximierungsproblemen entsprechen. Die Inferenzmethoden, die wir für diese Modellklassen entwickeln, wenden wir dann auf die Klasse der gemischten submodularen Modelle an, die jede diskrete Verteilung erfassen können, in einem expectation-propagation Rahmen. Schließlich werden wir die Familie der Cooperative Graphical Models diskutieren, die durch das Cooperative Cut Problem in der Bildverarbeitung motiviert sind. Für ihre Optimierung werden wir Konzepte aus der Theorie der submodularen Funktionen mit Variationsmethoden für exponentielle Familien kombinieren.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my advisor Professor Andreas Krause for his excellent mentorship, for always great feedback and ideas, and for giving me the freedom to pursue my research interests. I would also like to thank Professors Francis Bach and Jeffrey Bilmes for agreeing to be on my thesis committee, and for providing me with invaluable feedback.

I am very grateful to the Swiss National Science Foundation and to Google for funding this research. I would also like to thank the Swiss government for the excellence scholarship during my Master’s program, to Jacobs University for funding me during my Bachelor’s degree, and for the scholarship from the Macedonian government during high school.

The research in this thesis would not have been possible without the input from numerous collaborators that I had the pleasure to work with over the years: Marwa El Halabi, Stefanie Jegelka, Sebastian Tschiatschek, Junyao Zhao and Jian Zhang. I would also like to thank Stefanie Jegelka for hosting me in her group and for the many research discussions we had.

Thanks to all members of the Learning and Adaptive Systems group for creating an excellent research environment. The many discussions we had have helped me grow both as a researcher and as a person. Special thanks to Rita Klute for helping with a myriad of administrative questions.

This endeavour would not have been possible without the support from my friends — thanks to all of you!

Finally, thanks to my parents Mihailo and Valentina for supporting me through all these years and for always encouraging me to follow my interests. Thanks to my wife Jana for constantly being by my side me and supporting me in my decisions despite the many sacrifices we had to make.
## CONTENTS

1 **INTRODUCTION** 1  
  1.1 Submodularity 3  
  1.2 Uncertainty and inference 4  
  1.3 From optima to distributions 4  
  1.4 Approximate inference 5  
  1.5 Problem statement and contributions 6  
  1.6 Thesis organization 7  
  1.7 Publications 7  
  1.8 Collaborators 8  

2 **BACKGROUND — SET FUNCTIONS AND SUBMODULARITY** 9  
  2.1 Set functions and optimization 9  
  2.2 Continuous extensions 11  
  2.3 Submodularity 15  
  2.3.1 Operations preserving submodularity 18  
  2.4 Submodular polyhedra 18  
  2.5 Matroids 22  
  2.6 Submodular minimization and the Lovász extension 24  
  2.6.1 The minimum norm problem 27  
  2.6.2 Graph-representable functions 31  
  2.7 Submodular maximization 32  
  2.7.1 The continuous greedy algorithm 33  
  2.7.2 $\mathcal{M}^\alpha$-Concavity 34  

3 **BACKGROUND — PROBABILISTIC INFERENCE** 35  
  3.1 The inference problems and their hardness 36  
  3.1.1 Computational hardness 38  
  3.2 Exponential family models 38  
  3.2.1 Examples 41  
  3.3 Approximate inference techniques 42  
  3.3.1 Variational inference for exponential families 42  
  3.3.2 Mean-field 45  
  3.3.3 Perturb-and-MAP 46  
  3.3.4 Sampling 47
4 VARIATIONAL INFERENCE USING INFINITE DIVERGENCES 49
4.1 Introduction 49
4.2 Background — Rényi divergences 51
4.3 The infinite divergence for discrete models 55
  4.3.1 Connection to global estimators 59
4.4 The fully factorized binary case 60
  4.4.1 As optimization over general polyhedra 60
  4.4.2 The entropic forms 62
4.5 The Markov Random Field case 64
4.6 Relationship to Perturb-and-MAP 66

5 INFERENCE IN LOG-SUBMODULAR AND LOG-SUPERMODULAR MODELS 69
5.1 Introduction 69
5.2 The model classes 72
  5.2.1 Log-supermodular models 72
  5.2.2 Log-submodular models 74
5.3 Inference in log-submodular models 77
5.4 Inference in log-supermodular models 83
5.5 Approximation guarantees 88
  5.5.1 Curvature-based bounds 89
  5.5.2 Constant factor approximation for log-polymatroids 94
5.6 Experiments 97
  5.6.1 Synthetic models 97
  5.6.2 Image segmentation 100
5.7 Conclusion 106

6 INFERENCE IN MIXED SUBMODULAR MODELS 107
6.1 Introduction 107
6.2 Inference in factorized models 109
  6.2.1 Dual decomposition and expectation propagation 109
  6.2.2 The polyhedral view 115
6.3 Inference in mixed submodular models 116
  6.3.1 Inference in factorized log-supermodular models 117
6.4 Experiments 121
6.5 Conclusion 122

7 COOPERATIVE GRAPHICAL MODELS 125
7.1 Motivation 125
7.2 Introduction 126
7.3 Inference in cooperative graphical models 128
7.4 Convex upper bounds 129
  7.4.1 Algorithms 131
  7.4.2 Convergence 132
  7.4.3 Joint optimization 133
7.5 Submodular lower bounds 134
7.6 Approximate inference via MAP perturbations 135
7.7 Experiments 136
7.8 Conclusion 137

8 LEARNING OF LOG-SUPERMODULAR MODELS 143
  8.1 Motivation 143
  8.2 Introduction 144
  8.3 Argmin-differentiation of isotonic regression 147
  8.4 Min-norm differentiation 151
  8.5 Analysis 154
    8.5.1 When will the approximate approach work? 155
    8.5.2 Exact computation 160
    8.5.3 Structured attention and constraints 165
  8.6 Experiments 165
  8.7 Conclusion 167

9 CONCLUSION AND FUTURE WORK 171
  9.1 Future work 172

BIBLIOGRAPHY 173

A APPENDIX 187
  A.1 Convex optimization 187
  A.2 Proof of Theorem 6.4 188
### NOTATION

<table>
<thead>
<tr>
<th>SYMBOL</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>[[T]]</td>
<td>Iverson bracket, 1 if T is correct, 0 otherwise</td>
</tr>
<tr>
<td>(B(F))</td>
<td>The base polytope of F</td>
</tr>
<tr>
<td>(\mathcal{P}(F))</td>
<td>The submodular polyhedron of F</td>
</tr>
<tr>
<td>(n)</td>
<td>The size of the ground set, or number of variables</td>
</tr>
<tr>
<td>(V)</td>
<td>The ground set, typically defined as ({1, 2, \ldots, n})</td>
</tr>
<tr>
<td>(A, B, C, D)</td>
<td>Sets, subsets of (V)</td>
</tr>
<tr>
<td>(\mu)</td>
<td>Marginals, moments under the distribution</td>
</tr>
<tr>
<td>(\mathcal{M})</td>
<td>The marginal polytope</td>
</tr>
<tr>
<td>(\mathcal{M})</td>
<td>An approximation to the marginal polytope</td>
</tr>
<tr>
<td>(H[Q])</td>
<td>The Shannon entropy</td>
</tr>
<tr>
<td>(\mathcal{H})</td>
<td>An approximation to the entropy</td>
</tr>
<tr>
<td>(\tau)</td>
<td>Pseudo-marginals</td>
</tr>
<tr>
<td>(q)</td>
<td>The vector holding the parameters of the approximation</td>
</tr>
<tr>
<td>(Q)</td>
<td>The considered family of approximate distributions</td>
</tr>
<tr>
<td>(Q)</td>
<td>Approximate distribution</td>
</tr>
<tr>
<td>(P)</td>
<td>The true distribution we are trying to approximate</td>
</tr>
<tr>
<td>(F)</td>
<td>The energy function of the true distribution</td>
</tr>
<tr>
<td>(Q)</td>
<td>The energy function of the approximate distribution</td>
</tr>
<tr>
<td>(A(-q))</td>
<td>The log-partition function</td>
</tr>
<tr>
<td>(A^*(\mu))</td>
<td>The entropy function, convex conjugate of (A)</td>
</tr>
<tr>
<td>(f_V)</td>
<td>The convex envelope of (F) on ({0, 1}^n)</td>
</tr>
<tr>
<td>(f_\wedge)</td>
<td>The concave envelope of (F) on ({0, 1}^n)</td>
</tr>
<tr>
<td>(f_L)</td>
<td>The Lovász extension of (F)</td>
</tr>
<tr>
<td>(\log Z)</td>
<td>The true partition function</td>
</tr>
<tr>
<td>(x, y, z, w)</td>
<td>Vectors in (\mathbb{R}^n)</td>
</tr>
<tr>
<td>(\mathcal{X})</td>
<td>The set of feasible configurations</td>
</tr>
<tr>
<td>(\text{Conv } X)</td>
<td>The convex hull of the vectors in (X)</td>
</tr>
</tbody>
</table>
INTRODUCTION

Discrete optimization problems are ubiquitous in machine learning and data mining. Namely, in many situations we have to make one of finitely many choices. We start this chapter by showcasing three diverse machine learning scenarios where we naturally arrive at such problems.

First, let us consider the problem of clustering, where we are given a set of points with the goal to group, or cluster, them in an unsupervised manner into $k$ groups. A common approach is to identify a set of $k$ points from the given data that would act as prototypical examples. If we then assign each point to the cluster whose prototype is most similar to it, we have to solve the following discrete problem (1): which subset of $k$ prototypes yields the best clustering?

The second instance that we present is the sensor placement problem of Krause, Singh & Guestrin [2]. Namely, we are given a set of finitely many spatial locations where we can place sensors that measure the air temperature. Moreover, the measurements that we obtain are not only noisy but also correlated with each other, as we would not expect the temperature to be fully independent of the location. Then, given a budget of say $k$ sensors, the problem is that of devising a placement strategy that would allocate the sensors in an optimal way. Informally, we want the measurements coming from these $k$ sensors to predict the temperature at all locations as accurately as possible. More precisely, it would be information theoretically sound to choose a subset of $k$ locations with a maximal joint entropy, or that have a high mutual information with the unobserved locations.

Another common setting in machine learning where discrete optimization naturally occurs is that of structured prediction (Tsochantaridis et al. [4]). As a concrete prominent instantiation, let us consider the problem of semantic image segmentation. Given an image of size $n \times m$ represented as an RGB array $x \in \mathbb{R}^{n \times m \times 3}$, the goal is to predict which of the $n \times m$ pixels contain an object. If we encode the presence of an object at some location pixel with a one, we can also state the problem as that of computing from $x$ a labelling $y \in \{0, 1\}^{n \times m}$. For example, given the image in Figure 1.1a, the ground truth segmentation is presented in Figure 1.1b. Note that, while there are a total of $2^{nm}$ possible segmentations, the output space admits a natural structure, which we can leverage instead of treating the problem as
that of classification with an exponentially large label space. For example, we could learn a function $\theta: \mathbb{R} \rightarrow \mathbb{R}$ that will output for each pixel $x_{i,j}$ a real number $\theta(x_{i,j}) \in \mathbb{R}$, that can be then thresholded to get a prediction. Formally, we predict the label of pixel $(i,j)$ to be $\hat{y}_{i,j} = \mathbb{1}_{\theta(x_{i,j}) \geq 0}$. However, due to its simplicity this approach might fail to achieve high accuracy, and moreover it might be easily susceptible to noise. To tackle both of these problems, we could also model the correlations between the labels. For example, it would be natural to assume that the set of foreground objects form several connected components, rather than being randomly scattered across the image. Hence, we can bias the segmentation $y$ towards those configurations with a small number of adjacent pixels with disagreeing labels, which we will denote by $\Omega(y)$. For example, we can produce a segmentation by solving $\arg\max_{y \in \{0,1\}^{n \times m}} y^\top \theta(x) - \Omega(y)$, which is a complex discrete optimization problem. This method, known as graph cut segmentation, has been successfully applied in computer vision ([5]).
1.1 SUBMODULARITY

The above problems can not be solved using exhaustive search — for example, in the first two examples there are a total of \( \binom{n}{k} \) candidate solutions. Fortunately, all three functions we considered share a common trait that renders the resulting discrete optimization tractable. If we take a closer look at them, we can note that they satisfy a diminishing returns property. For example, if we iteratively introduce new clusters, we expect the additional benefit of each additional one to decrease due to the fact that there will be less data variability to be explained. Similarly, connecting a new sensor to a larger installation can only result in smaller reduction in the entropy, than when adding it to configurations with only a few sensors. For the image segmentation example, a similar behaviour can be also observed. Specifically, the cost of changing the label of a pixel from one to zero should monotonically decrease with the number of neighbours already assigned to the background. Hence, we can also see that the neighbouring pixels exhibit a cooperative behaviour — they would rather all take on the same label.

This property of diminishing returns is known as submodularity. Informally, a function \( F \) is submodular if the gain of adding an item decreases as we increase the context in which it is added. For example, assume that we are buying electronics, and the function \( F(A) \) measures the utility of purchasing products \( A \). Then, as illustrated in Figure 1.2, for \( F \) to be submodular we require the benefit of purchasing say a smart phone to decrease as we own more items with similar capabilities, such as cameras and tablets.

Submodular functions have very compelling algorithmic properties. Despite being defined using only this natural condition, they can be exactly minimized in polynomial time, and maximized using very simple algorithms within provably multiplicative approximation factors. Hence, all three examples that we have outlined in the previous section can be solved (nearly) optimally in polynomial time.

---

1 Icons from [www.github.com/google/material-design-icons](http://www.github.com/google/material-design-icons).
1.2 Uncertainty and Inference

In all three examples above, the functions depend directly on noisy sampled data. Namely, the clustering that we end up with depends on which points we have sampled, the sensor placement depends on the estimated correlations between the sites, and the segmentation might change as a function of both the noise levels of the pixels and the behaviour of the function $\theta$ under such perturbations. It might thus seem troubling to focus on a single solution, when the objective, and as a result also possibly the optima, change with the statistical fluctuations in the data. Moreover, we sometimes make complex decisions with the optima we have computed. In that case, reducing ourselves to one optimum might result in taking sub-optimal actions, as we ignore the inherent problem uncertainty. Thus, we argue for a probabilistic approach, which would let us handle the whole decision process in a principled manner.

As a concrete example that illustrates the possible pitfalls of focusing on one configuration, let us consider an instance where image segmentation can be applied. Specifically, assume that we want to help doctors by automatically segmenting suspicious lesions from an MRI scan. Then, based on the segmentation that we compute, the doctor has to take one of finitely many actions $A$. If perturbations in the input can significantly increase or decrease the volume of the segmented area, more tests might be necessary. Unfortunately, this will be impossible to conclude from a single segmentation $y$, which might over- or under-estimate the true volume of the lesion. If we, however, instead compute a distribution over the possible segmentations $P(Y = y)$, we can solve this problem using the framework of Bayesian decision theory. Formally, if $\ell(a, y)$ is the incurred loss by taking action $a \in A$ when the true segmentation is $y \in \{0, 1\}^{n \times m}$, we should choose the action that minimizes the expected risk $E_y[\ell(a, y)]$.

1.3 From Optima to Distributions

There are two problems that we have to solve if we want to introduce a probabilistic framework. First, we need to come up with a principle of defining distributions, and second, we need computationally efficient algorithms that can perform inference in them.

Let us consider the segmentation example, which maximized the function $F(y) = \theta(x)^\top y - \Omega(y)$ over $y \in \{0, 1\}^n$. We want to design a distribution $P(y)$ that assigns higher likelihoods to those configurations $y$ for
which $F$ is large. It thus makes sense to consider distributions of the form $P(y) \propto h(F(y))$ for some monotonically increasing function $h: \mathbb{R} \to [0, \infty)$. Perhaps the most common choice is $h(u) = \exp(u/\lambda)$ for some $\lambda > 0$, in which case the resulting distribution is called a Gibbs distribution. The number $\lambda$, also called the temperature, acts as a concentration parameter — as $\lambda \to 0$ the distribution will focus on the maxima, while it will converge to the uniform distribution as $\lambda \to \infty$. Throughout the thesis we undertake the above approach, and analyze the corresponding Gibbs distributions for a wide family of problems leveraging submodularity.

Alas, once we move from optimization to probabilistic inference, the computational landscape drastically changes. Many problems that admit efficient optimization algorithms unfortunately result in intractable probabilistic models. In the example above, while we know how to find the best labelling due to the submodularity of the model, working with the Gibbs distribution is provably hard. Namely, despite the importance of the resulting distribution, also known as the Ising model, in statistical physics and the amount of attention it has received, there exist only a few cases where we can efficiently work with it. We thus have to in general resort to approximate inference techniques.

1.4 approximate inference

Considering the importance of the problem, several frameworks have been proposed for the development of approximate inference methods targeting discrete models. In this thesis we focus on variational techniques — the inference problem is posed as that of optimization, which can be then relaxed and efficiently optimized.

Many traditional variational inference techniques are only applicable to models that are of a very low order — specifically, only those that can be written as a product of simpler terms, each of them defined over a small subset of the domain. However, not all submodular functions can be written in such a form. In the introductory examples, neither the clustering loss, nor the entropy used to measure the quality of a sensor placement admit such a decomposition. While there do exist method that can be used for higher order models, they typically work only for specific families, or might require repeated sampling to perform the necessary updates.

Instead, we would like to construct methods that can be applied to models arising from any submodular function, as long as we have an oracle that evaluates it. To this end, we will identify pairs of approximation families
Figure 1.3: Approximate inference by divergence minimization. The rectangle denotes the set of all distributions, while the blue ellipsoid contains the tractable measures $Q$. We approximate the target $P$ using the distribution $Q$ that is inside the blue region and is as close as possible to $P$, as measured by the divergence $D(P \| Q)$.

$Q$ and corresponding divergences, which can be combined to perform approximate inference. Specifically, we will find the closest distribution from $Q$ as measured by the divergence, as illustrated in Figure 1.3. While inference using divergence minimization has been used before in the literature, most approaches suffer from the aforementioned issues.

1.5 Problem Statement and Contributions

The primary goal of this thesis is to both introduce rich probabilistic models that leverage submodularity, and create a framework for variational inference in them. Specifically, we would like to develop an approach that would enable us to devise efficient inference schemes for distributions that use submodular functions as their integral component.

To this end, we first identify a set of discrepancy measures, the infinite Rényi divergences, that are well-tailored for inference in submodular models. Then, we introduce several model classes, and show how to combine these divergences together with the properties of submodular functions into efficient inference algorithms. In the last chapter, we show how to use these divergences to not only perform inference, but to also learn a specific class of models.
1.6 Thesis Organization

The chapters of this thesis are organized as follows.

- **Chapter 2** We start the thesis with an overview on discrete functions, submodularity and optimization.

- **Chapter 3** Then, we provide the necessary background on approximate inference in discrete models.

- **Chapter 4** In the fourth chapter we discuss divergence measures and their properties, with a special focus on the infinite Rényi divergences.

- **Chapter 5** The first class of models we analyze are the log-submodular and log-supermodular distributions.

- **Chapter 6** We then move on to the family of mixed submodular models, which can represent any distribution over sets.

- **Chapter 7** Cooperative graphical models, inspired by the cooperative cut problem in computer vision, is the third family that we examine. We will perform inference in them by combining classical variational methods with submodularity.

- **Chapter 8** In the penultimate chapter we show how to learn log-supermodular models using our approximate inference techniques.

- **Chapter 9** We finally conclude the thesis with an overview of the results, and by presenting several possible directions for future work.

1.7 Publications

Most of the material in this thesis has been already published in the following conference proceedings.


The following publications and technical reports are also relevant to this thesis, but have not been directly incorporated.


• Zhao, J., Djolonga, J., Tschiatschek, S. & Krause, A. *Improving Optimization-Based Approximate Inference by Clamping Variables* in *Conference on Uncertainty in Artificial Intelligence (UAI)* (2017)


1.8 Collaborators

This thesis would not have been possible without the contributions of several collaborators that I had worked with over the past years. The idea to pursue the topic has been suggested by my advisor Prof. Dr. Andreas Krause, whose input had a large impact on every contribution of this thesis. With Dr. Sebastian Tschiatschek we have worked on several projects together, two of which have been included in the thesis: the mixed models in Chapter 6, and the cooperative graphical models from Chapter 7. The work in Chapter 7 contains also contributions from Prof. Dr. Stefanie Jegelka, with whom I have also worked on a second project on inference under constraints, which has not been directly included in the thesis.
When developing inference algorithms for discrete distributions, we will constantly work with certain functions associated with them. Namely, we will work with set functions, which take as arguments subsets of some ground set $V$, which we will w.l.o.g. always assume to be $V = \{1, 2, \ldots, n\}$.

Formally, we say that $F: 2^V \to \mathbb{R}$ is a set function with ground set $V$, where we denote by $2^V$ the collection of all $2^n$ subsets of $V$. Instead of thinking of set functions as being defined over $2^V$, we can also treat them as acting on the corners of the unit cube $\{0, 1\}^n$. A formal connection can be easily made, if we define for each $A \subseteq V$ its characteristic Boolean vector $\mathbf{1}_A$ with ones in the positions corresponding to $A$. Formally, we can establish the following bijection $\{0, 1\}^n \ni \mathbf{x} = \mathbf{1}_A \leftrightarrow A = \{i \in V: x_i = 1\} \in 2^V$.

Throughout the thesis we will interchangeably treat set functions as being defined over $2^V$ or $\{0, 1\}^n$. To make sure that no confusion arises, we will denote vectors by lower-case boldface letters (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots$), and sets by upper-case letters (e.g., $A, B, C, \ldots$). Another assumption that we will make, is that set functions are normalized, so that $F(\emptyset) = F(0) = 0$. This is of course without any loss of generality, as we can always work instead with the shifted function $F'(A) = F(A) - F(\emptyset)$, which is normalized.

**Example 2.1 (Modular functions).** A function $F: 2^V \to \mathbb{R}$ is called (normalized) modular if it is equal to $F(A) = \sum_{i \in A} F(\{i\}) = \sum_{i \in A} \theta_i$ for some real numbers $\theta_i = F(\{i\}) \in \mathbb{R}$. Hence, they are uniquely defined using $n$ numbers $\theta_i$, which we can collect in a vector $\mathbf{\theta} \in \mathbb{R}^n$, so that we can write the function in vector form as $F(\mathbf{x}) = \mathbf{x}^\top \mathbf{\theta}$. They can be thus seen as playing the role of linear functions in the discrete setting. As commonly done in combinatorial optimization, for any vector $\mathbf{\theta} \in \mathbb{R}^n$ and any subset of its coordinates $A \subseteq V$, we define the corresponding modular function as $\mathbf{\theta}(A) = \sum_{i \in A} \theta_i = \mathbf{\theta}^\top \mathbf{1}_A$.

**Example 2.2 (Graph cuts).** A very important class of functions are the classical graph cuts. Given a weighted (either directed or undirected) graph $G = (V, E, w)$...
with vertex set $V$, edge set $E$, and a non-negative edge weighting function $w: E \rightarrow [0, \infty)$, the graph cut function $F: 2^V \rightarrow \mathbb{R}$ is defined as

$$F(A) = \sum_{e \in E} w(e) \mathbb{1}[A \text{ cuts } e],$$

where $\mathbb{1}[]$, called the Iverson bracket, evaluates to one if its argument is true and to zero otherwise. A subset of the vertices $A$ is said to cut an undirected edge if exactly one of its end-points is in $A$, and a directed edge $u \rightarrow v$ only if $u$ is in $A$. As a concrete example, consider the directed graph on Figure 2.1a, and the set $A = \{1, 3\}$. Note that the edges that are cut between $A$ and $V - A$ are $\{1, 4\}$ and $\{3, 2\}$, with a total edge weight of $F(A) = 4 + 8 = 12$.

**Example 2.3** (Set coverage). Assume that we are given a collection of $n$ sets $U_1, U_2, \ldots, U_n$ — for example these could be different regions in the plane. The set cover function is defined as $F(A) = |\bigcup_{i \in A} U_i|$. As a concrete example, consider the three sets $U_1 = \{a, b\}$, $U_2 = \{b, c, d\}$, and $U_3 = \{a, c, d, e\}$, as illustrated in Figure 2.1b. Then, we have for example that $F(\{1, 2\}) = |U_1 \cup U_2| = |\{a, b, c, d\}| = 4$, while $F(\{1, 3\}) = |U_1 \cup U_3| = |\{a, b, c, d, e\}| = 5$.

**Optimization** Typically, we design set functions $F: 2^V \rightarrow \mathbb{R}$ to measure the utility that we would like to maximize, or some cost that we want to minimize, of the sets $A \subseteq V$. For example, if $V$ is the inventory of some shop, $F(A)$ could be the utility we get from buying items $A$. Thus, a very natural problem is that of optimizing $F$, i.e., selecting the best subset $A \subseteq V$ that maximizes, or minimizes, the specified function $F$. As an additional requirement, there might be some constraints on which configurations from $2^V$ we are allowed to select. In the example above, we might want to only consider item sets whose total cost does not exceed some fixed budget. We are thus in general considering only some collection $C \subseteq 2^V$ of allowed subsets, so that resulting problems can be written as

$$\max_{A \in C} F(A) \quad \text{and} \quad \min_{A \in C} F(A).$$

Let us consider two classical examples.

**Example 2.4** (Maximum set cover [15]). The maximum set cover problem is that of selecting at most $k$ elements that maximize the coverage function from Example 2.3, i.e.,

$$\max_{A \subseteq V: |A| \leq k} |\bigcup_{i \in A} U_i|.$$
The constraint in this case is also called a cardinality constraint. For example, if \( k = 1 \) we should pick the index of the set in with the largest cardinality, which is \{3\} in Figure 2.1b. The \( k = 2 \) case in Figure 2.1b has as solution \{2, 3\}, as that would cover all elements in \( U_1 \cup U_2 \cup U_3 \).

Example 2.5 (Minimum graph cut (see e.g. Cormen [16])). On the other hand, cut functions are typically used to model costs. As one typical example, assume that we have a set \( V \) of \( n \) points that we want to cluster into two clusters \((A, V - A)\), and we moreover have for each pair of points in \( V \) some measure of similarity. Then, as a very simple criterion (see e.g. Flake, Tarjan \\& Tsoutsouliklis [17]) of the quality of some clustering \((A, V - A)\) we can use the graph cut function \( F(A) \) on the complete graph with the similarity measures as weights. As we want neither partition to be empty, we want to optimize

\[
\min_{A \in 2^V - \{\emptyset, V\}} F(A).
\]

In the concrete example in Figure 2.1a, the optimum is \( A = \{4\} \) with a corresponding value of 7.

2.2 CONTINUOUS EXTENSIONS

The first problem that we address is that of computing continuous extensions of set functions. Namely, given a set function \( F: \{0,1\}^n \to \mathbb{R} \), the
goal is to design a continuous function \( f: [0,1]^n \rightarrow \mathbb{R} \) that agrees with \( F \) on \( \{0,1\}^n \), i.e., \( \forall x \in \{0,1\}^n: F(x) = f(x) \). In other words, we have a function \( F \) that is defined only on the corners of the unit cube, and we want to extend it over the complete unit cube by filling the missing part in a continuous manner. To illustrate the task, we show in Figure 2.2 three such extensions for the function \( F(x_1, x_2) = [x_1 = x_2] \). While we can easily construct such functions in the two dimensional case, it is not immediately clear how to do it for large \( n \). In this section we will present three principled approaches of deriving such extensions, and show how they are connected to the minimization and maximization problems we have introduced. The results that we state here mainly follow the survey of Dughmi [18], even though they are well-known in the literature and have been stated at multiple places.

The first extension that we discuss is the so-called convex envelope, or convex closure, of \( F \). Namely, given that convex functions are closed under point-wise supremum, we consider the tightest convex function by taking the supremum of all convex functions that are extensions of \( F \).

**Definition 2.1** ([18, Definition 3.1]). The convex extension of a set function \( F: \{0,1\}^n \rightarrow \mathbb{R} \) is the function \( f_\vee: [0,1]^n \rightarrow \mathbb{R} \) defined as

\[
f_\vee(x) = \sup_{\text{convex } f: [0,1]^n \rightarrow \mathbb{R}, \ f \leq F \text{ on } \{0,1\}^n} f(x).
\]

Similarly, we can also define a concave envelope, by considering all concave majorants of the discrete function and take their point-wise infimum.

\[
|1 - x_1 - x_2| \quad (1 - x_1 - x_2)^2 \quad 1 - (x - y)^2
\]

**Figure 2.2:** Different extensions of the set function \( F(x_1, x_2) = [x_1 = x_2] \).
2.2 Continuous Extensions

Definition 2.2 ([18, Definition 3.9]). The concave extension of a set function $F: \{0,1\}^n \rightarrow \mathbb{R}$ is the function $f_\land: [0,1]^n \rightarrow \mathbb{R}$ defined as

$$f_\land(x) = \inf_{\text{concave } f: [0,1]^n \rightarrow \mathbb{R}} f(x).$$

It turns out that we do not have to consider all lower- and upper-bounds in the above definitions, and we can only focus on affine functions. This can be intuitively seen from the fact that (closed) convex and concave functions can be written as a supremum over affine functions via their conjugates.

Before we state the formal result, let us define the polyhedra holding all lower and upper affine bounds on $F$.

Definition 2.3 ([19, §2.3]). Given a function $F: 2^V \rightarrow \mathbb{R}$ and a collection $C \subseteq 2^V$, we define its generalized lower polyhedron as

$$\mathcal{L}(F | C) = \{(x,c) \in \mathbb{R}^n \times \mathbb{R} | \forall A \in C: x(A) + c \leq F(A)\},$$

and also use the shorthand $\mathcal{L}(F) = \mathcal{L}(F | 2^V)$.

Definition 2.4 ([19, §6.3]). Given a function $F: 2^V \rightarrow \mathbb{R}$ and a collection $C \subseteq 2^V$, we define its generalized upper polyhedron as

$$\mathcal{U}(F | C) = \{(x,c) \in \mathbb{R}^n \times \mathbb{R} | \forall A \in C: x(A) + c \geq F(A)\},$$

and also use the shorthand $\mathcal{U}(F) = \mathcal{U}(F | 2^V)$.

Equipped with these definitions, we can express the envelopes as follows.

Theorem 2.1 (Iyer & Bilmes [19, Lemmas 3.1 and 7.1]). We have that

$$f_\lor(x) = \inf_{(q,c) \in \mathcal{L}(F)} x^\top q + c,$$

and

$$f_\land(x) = \sup_{(q,c) \in \mathcal{U}(F)} x^\top q + c.$$  

When deriving dual problems the convex and concave conjugates of $f_\lor$ and $f_\land$ will be very useful, which we define below.

Lemma 2.1 ([20, §5.1]). The convex conjugate $f_\lor^*$ of $f_\lor$, and the concave conjugate $f_\land^*$ of $f_\land$ are equal to

$$f_\lor^*(q) = \sup_{x \in \{0,1\}^n} q^\top x - F(x),$$

and

$$f_\land^*(q) = \inf_{x \in \{0,1\}^n} q^\top x - F(x).$$
Note that, as commonly done in the literature, we could-have obtained the envelopes by taking the convex (concave) bi-conjugate of the function that agrees with $F$ on $\{0, 1\}^n$ but is equal to $+\infty (-\infty)$ elsewhere.

The importance of the convex and concave envelopes can be seen from their relationship to discrete optimization.

**Theorem 2.2** ([20, §5.1]). The convex and concave relaxations are exact for the problems of minimization and maximization respectively, i.e.,

$$
\begin{align*}
\min F(A) &= \min_{A \subseteq V} f_\vee(x), \text{ and } \\
\max F(A) &= \max_{A \subseteq V} f_\wedge(x).
\end{align*}
$$

Hence, if we can efficiently compute the convex (concave) extension we can solve the resulting minimization (maximization) problem.

Finally, there is yet another, probabilistic interpretation of these extensions, which will be useful in the interpretation of the variational bounds we will develop in Chapter 4.

**Theorem 2.3** (Dughmi [18, §3.2, §3.1.1]). For any $\mu \in [0, 1]^n$ define by $\mathcal{P}(\mu)$ the set of all distributions on $\{0, 1\}^n$ with marginals $\mu$, i.e., each $\mathbb{P} \in \mathcal{P}(\mu)$ satisfies $\mathbb{E}_{x \sim \mathbb{P}}[x_i] = \mu_i$. Then, the convex and concave envelopes can be written as

$$
\begin{align*}
f_\vee(\mu) &= \inf_{\mathbb{P} \in \mathcal{P}(\mu)} \mathbb{E}_{x \sim \mathbb{P}}[F(x)], \text{ and } \\
f_\wedge(\mu) &= \sup_{\mathbb{P} \in \mathcal{P}(\mu)} \mathbb{E}_{x \sim \mathbb{P}}[F(x)].
\end{align*}
$$

Unfortunately, as one would expect given their connection to the optimization problems, both these extensions are in general hard to compute or approximate. Another extension that is also used in the literature, but can be easily estimated by sampling, is the so-called multi-linear extension.

**Definition 2.5** (Owen [21]). The multi-linear extension of a function $F : \{0, 1\}^n \to \mathbb{R}$ is the function $f_\sim : [0, 1]^n \to \mathbb{R}$ defined as

$$
f_\sim(\mu) = \mathbb{E}_{x_i \sim \text{Bernoulli}(\mu_i)}[F(x)].
$$

Because the distribution used in the definition $f_\sim$ also belongs to $\mathcal{P}(\mu)$, we have the following relationship between these three extensions ([18])

$$
\forall x \in [0, 1]^n : f_\vee(x) \leq f_\sim(x) \leq f_\wedge(x).
$$
2.3 SUBMODULARITY

Submodular functions are a family of set functions that can both model useful phenomena, have a very rich theory, and lend themselves to the design of efficient optimization algorithms.

**Definition 2.6** (Submodularity [22, §2.3]). A function $F : 2^V \to \mathbb{R}$ is said to be **submodular** if for all $A, B \subseteq V$ we have that

$$F(A \cup B) + F(A \cap B) \leq F(A) + F(B). \quad (2.10)$$

Note that in terms of vectors condition (2.10) becomes

$$F(\min(x, x')) + F(\max(x, x')) \leq F(x) + F(x') \quad \text{for all } x, x' \in \{0, 1\}^n,$$

where the min and max operations are defined in a coordinate-wise manner. Similarly to the connection between convex and concave functions, submodular functions have their supermodular counterparts.

**Definition 2.7** (Supermodularity [22, §2.3]). A function $F : 2^V \to \mathbb{R}$ is called **supermodular** if $-F$ is submodular.

There is an equivalent definition of submodularity that is useful in understanding what kind of utilities they model. To state it, we first have to define the notion of a marginal gain, which quantifies by how much we the function value changes if we increase a set from $A \to A \cup B$.

**Definition 2.8** (Marginal gain). Given a function $2 : V \to \mathbb{R}$, the marginal gain of $A \subseteq V$ given $B \subseteq V$ is defined as

$$F(B \mid A) = F(A \cup B) - F(A).$$

With a slight abuse of notation, for any $i \in V$ and $A \subseteq V$ we also use $F(i \mid A)$ as a shorthand for $F(\{i\} \mid A)$. We can now state the alternative definition of submodularity.

**Theorem 2.4** ([15, Proposition 2.1]). A function $F : 2^V \to \mathbb{R}$ is submodular iff for all $A, B \subseteq V$ such that $A \subseteq B$ and any $i \in V - B$ it holds that

$$F(i \mid A) \geq F(i \mid B). \quad (2.11)$$

We can also state the above definition as the requirement the functions $\Delta_i : A \to F(i \mid A)$ to be non-increasing with respect to $\subseteq$ for all $i \in V$. When submodular functions are used as utilities, this definition is known...
as the diminishing returns property — the benefit of including an item can only decrease as we increase the context in which we add it. Many functions are not only submodular, but are also monotone with respect to set inclusion, so that adding an element can never result in a negative gain.

**Definition 2.9 (Monotonicity).** A function $F: 2^V \rightarrow \mathbb{R}$ is said to be monotone if $A \subseteq B$ implies $F(A) \leq F(B)$.

**Definition 2.10 (Polymatroid [22, §2.2]).** A normalized monotone submodular function is called a polymatroid (rank function).

In the polymatroid case, we have a much tighter relationship between the extensions $f_\sim$ and $f_\land$ — namely, they are always within a constant of each other, a result also known as the correlation gap inequality.

**Theorem 2.5 (Correlation gap inequality [23, Corollary 1.2, 24, Lemma 3.8]).** If $F$ is polymatroid, then it also holds that

$$\forall x \in [0,1]^n : f_\sim(x) \geq (1 - 1/e) f_\land(x).$$

All three examples that we have introduced in the beginning (Examples 2.1 to 2.3) are indeed submodular. While the set cover function is also monotone, that is not always true for modular functions or cuts. Namely, modular functions are monotone iff all their components are non-negative, while cuts not only fail to be monotone, but are symmetric, i.e., they satisfy $F(A) = F(V - A)$. We discuss below a few other commonly used families of submodular functions.

**Example 2.6 (Threshold potentials [25, 26]).** The functions of the form $F(A) = \min\{a, \sum_{i \in A} \theta_i\}$ are polymatroids if $a \geq 0$ and $\theta_i \geq 0$, and this can be easily proven using the diminishing returns definition.

**Example 2.7 (Concave-of-modular functions).** The functions of the form $F(A) = h(m(A))$ for some concave $h: [0,n] \rightarrow \mathbb{R}$ and non-negative modular function $m$ are submodular ([20, Proposition 6.1]). Of particular interest are the concave-of-cardinality functions, i.e., those of the form $F(A) = h(|A|)$, which of course correspond to using the modular function $m(A) = |A|$. These functions can be also seen as the link between submodularity and concavity ([20]), and are the building block in the construction of deep submodular functions [27].

**Example 2.8 (Facility location).** Another function widely used as a utility is the so-called facility location function, which has the form

$$F(A) = \sum_{j=1}^{m} \max_{i \in A} w_{i,j},$$
for some non-negative numbers \( w_{i,j} \geq 0 \). The name comes from the classical facility location problem from operations research, where one has to decide where to open facilities from \( n \) candidate locations. These facilities have to serve \( m \) clients, and each client will be served from exactly only one facility. If we assume that client \( j \) receives utility of \( w_{i,j} \geq 0 \) from facility \( i \), then the above function captures the sum of all client utilities.

**Example 2.9** (Entropy [28, 29]). Let \( X_1, X_2, \ldots, X_n \) be a set of discrete (continuous) variables. Then, the function mapping \( A \) to the (differential) Shannon entropy of the variable \( (X_i)_{i \in A} \) is submodular. Moreover, if each \( X_i \) is discrete, then the function is also monotone.

We have so far assumed that submodular functions are defined over all of \( 2^V \), but if we have a look at its definition (2.10), we can see that we only need the domain to be closed under unions and intersections. We call such collections lattices\(^1\).

**Definition 2.11** (Lattice [22, §2.3]). A family of subsets \( \mathcal{F} \subseteq 2^V \) of some ground set \( V \) is called a lattice if it is closed under unions and intersections, i.e., if \( A \) and \( B \) are in \( \mathcal{F} \), then so are \( A \cup B \) and \( A \cap B \).

**Definition 2.12.** The lattice consisting of all subsets \( \mathcal{F} = 2^V \) is called the simple lattice.

**Definition 2.13.** The lattice consisting of the empty set and the complete ground set \( \mathcal{F} = \{ \emptyset, V \} \) is called the trivial lattice.

Thanks to theorem of Birkhoff [30], we can compactly represent any lattice using a graph\(^2\). We show this result as presented in [22, §3.2(a)]. For an arbitrary lattice \( \mathcal{D} \) define for each \( i \in V \) the following

\[
\mathcal{D}(i) = \bigcap \{ A : i \in A \in \mathcal{D} \}. \tag{2.12}
\]

We moreover define the graph \( G(\mathcal{D}) \) with directed edges from \( i \) to each node in \( \mathcal{D}(i) \). We call a set \( A \subseteq V \) a lower ideal of \( G(\mathcal{D}) \) if it has no outgoing edges.

**Theorem 2.6** (Birkhoff [30]). Let \( \mathcal{D} \subseteq 2^V \) be a lattice containing \( \emptyset \) and \( V \). Then, the following two claims hold.

- Each lower ideal in \( G(\mathcal{D}) \) is in the lattice \( \mathcal{D} \).
- For each \( A \in \mathcal{D} \) there exists some lower ideal \( B \) of \( G(\mathcal{D}) \) equal to \( A \).

---

\(^1\) Even though lattices are typically more generally defined, in this thesis this simpler definition will suffice.

\(^2\) Technically, all the lower ideals of a poset, but this form will be more useful to us in Chapter 8.
2.3.1 Operations preserving submodularity

Simply looking at their definition (equation (2.10)) we can see that submodular functions are closed under addition and non-negative multiplication — in other words, the family of submodular functions forms a cone. They are also closed under several other operations — for example we can not create a non-submodular function by limiting its domain.

**Definition 2.14** (Restriction [22, 3.1(a)]). Given a submodular function \( F : 2^V \rightarrow \mathbb{R} \) and a set \( B \subseteq V \), we denote its restriction to \( B \) as \( F^B : 2^{V-B} \rightarrow \mathbb{R} \), defined as

\[
F^B(A) = F(A)
\]  

(2.13)

Another operation that we will define is that of contraction, which, informally stated, conditions on the inclusion of some set \( B \subseteq V \).

**Definition 2.15** (Contraction [22, 3.1(a)]). Given a submodular function \( F : 2^V \rightarrow \mathbb{R} \) and a set \( B \subseteq V \), we denote its contraction \( F_B \) of \( B \) as \( F_B : 2^{V-B} \rightarrow \mathbb{R} \), defined as

\[
F_B(A) = F(A \cup B) - F(B).
\]  

(2.14)

Note that we have subtracted \( F(B) \) to ensure that \( F_B \) is normalized so that \( F_B(\emptyset) = F(B) - F(B) = 0 \).

The last operation that we will introduce is that of reflection. Perhaps the easiest way to intuitively think about it is by inverting the domain \( \{0,1\}^n \) so that zeroes become ones and vice versa. Under such a transformation submodularity is invariant, and we define the resulting function below.

**Definition 2.16** (Reflection [20, Appendix B]). Given a submodular function \( F : 2^V \rightarrow \mathbb{R} \), we denote its reflection as \( \overline{F} : 2^V \rightarrow \mathbb{R} \), defined as

\[
\overline{F}(A) = F(V - A) - F(V).
\]  

(2.15)

2.4 Submodular Polyhedra

Submodular functions have several polyhedra that are associated with them, in addition to the lower and upper polyhedron from Definitions 2.3 and 2.4. The first polyhedron that we introduce is the set containing all modular lower bounds on \( F \), i.e., the slice of \( \mathcal{L}(F) \) obtained by taking \( c = 0 \).

**Definition 2.17** (Submodular polyhedron [22, §2.3]). Given a submodular function \( F : 2^V \rightarrow \mathbb{R} \) and a lattice \( D \) on \( V \), the submodular polyhedron \( \mathcal{P}(F \mid D) \) is defined as

\[
\mathcal{P}(F \mid D) = \{\mathbf{q} \in \mathbb{R}^n \mid \mathbf{q}(A) \leq F(A) \text{ for all } A \in \mathcal{D}\}.
\]  

(2.16)
We will also use the shorthand $\mathcal{P}(F) = \mathcal{P}(F | 2^V)$.

Of particular interest will be a specific face of the submodular polyhedron, known as the base polytope of the submodular function.

**Definition 2.18** (Base polytope [22, §2.3]). Given a submodular function $F : 2^V \to \mathbb{R}$ and a lattice $\mathcal{D}$ on $V$, the base polytope $\mathcal{P}(F | \mathcal{D})$ is defined as

$$B(F | \mathcal{D}) = \mathcal{P}(F | \mathcal{D}) \cap \{ q \in \mathbb{R}^n | q(V) = F(V) \}.$$  \hspace{1cm} (2.17)

We will also use the shorthand $B(F) = B(F | 2^V)$.

Even though the base polytope is defined with exponentially many inequalities, we can surprisingly optimize linear functions over it in $O(n \log n)$ time with only $n$ function evaluations. Namely, this can be done using the celebrated greedy algorithm of Edmonds [31], which we present in Algorithm 1.

**Algorithm 1** Edmonds’ algorithm to maximize of $\mathbf{w}^\top \mathbf{y}$ over $\mathbf{y} \in B(F)$.

1: procedure EdmondsGreedy($F$, $\mathbf{w}$)
2: Define $y^* = 0$
3: Sort $\mathbf{w}$ with some bijection $\sigma : V \to V$ such that $w_{\sigma(1)} \geq w_{\sigma(2)} \geq \ldots \geq w_{\sigma(n)}$.
4: for $i \leftarrow 1, 2, \ldots, n$ do
5: \hspace{1cm} $y^*_{\sigma(i)} \leftarrow F(\{ \sigma(i) \} | \{ \sigma(1), \ldots, \sigma(i-1) \})$
6: \hspace{1cm} end for
7: return $y^*$
8: end procedure

**Theorem 2.7** (Edmonds [31]). The solution returned from Algorithm 1 is a maximizer of $\mathbf{w}^\top \mathbf{y}$ over $\mathbf{y} \in B(F)$.

Moreover, not only can this greedy algorithm optimize linear functions over the base polytope, but it also gives a characterization of the extremal points of both $\mathcal{P}(F)$ and $B(F)$.

**Theorem 2.8.** The extreme points of $\mathcal{P}(F)$ and $B(F)$ are the same, and are exactly those points which are generated using the greedy algorithm.

Using this result we can characterize base polyhedra as convex hulls of finitely many points, namely the up to $n!$ points that can be obtained using the greedy algorithm.
Example 2.10. Consider a concave-of-cardinality function \( F(A) = h(|A|) \). Then, irrespective of the order that we choose, the marginal gains will always be equal to \( h(i) - h(i - 1) \) for \( i \in V \). Hence, if we construct the vector
\[
\Delta = (h(1), h(2) - h(1), \ldots, h(n) - h(n - 1)),
\]
the base-polytope is the convex hull of all permutations of the coordinates of \( \Delta \), which is also known as a permutahedron.

The function \( F(A) = \min\{k, |A|\} \), which can be written in the above form using \( h(z) = \min\{k, z\} \), has a corresponding vector \( \Delta = \{1\}^k \times \{0\}^{n-k} \). Hence, we can see that the base polytope is the convex hull of all \( k \)-sparse vectors, with the \( k = 1 \) reducing to the probabilistic simplex.

Another pair of polyhedra are the sub- and superdifferentials. They contain the modular functions that are global lower- and upper-bounds respectively, and are moreover tight at some specified set \( A \subseteq V \), and are defined analogously to the continuous case. However, while only convex and concave continuous functions have sub- and superdifferentials respectively, in the discrete case they are always non-empty.

Definition 2.19 (Subdifferential [22, § 6.2]). We define the subdifferential of \( F : 2^V \to \mathbb{R} \) at some \( A \subseteq V \) as
\[
\partial F(A) = \{q \in \mathbb{R}^n \mid \forall B \subseteq V : F(B) \geq q(B) + F(A) - q(A)\}. \tag{2.18}
\]

Definition 2.20 (Superdifferential [19, §6.2]). We define the superdifferential of \( F : 2^V \to \mathbb{R} \) at some \( A \subseteq V \) as
\[
\partial^F(A) = \{q \in \mathbb{R}^n \mid \forall B \subseteq V : F(B) \leq q(B) + F(A) - q(A)\}. \tag{2.19}
\]

Note that from the definitions, if \( q \in \partial F(A) (\partial_F(A)) \), then \( (q, c = F(A) - q(A)) \in \mathcal{L}(F) (\mathcal{U}(F)) \). Moreover, as we can not increase \( c \) without violating the feasibility inequalities, the subdifferentials (and superdifferentials analogously) form the faces of the generalized polyhedra.

Lemma 2.2 ([19, Lemmas 2.5 and 6.11]). The point \( (q, c) \) lies on the face of \( \mathcal{L}(F) (\mathcal{U}(F)) \) iff there exists some \( A \subseteq V \) such that
\[
q \in \partial_F(A)(\partial^F(A)) \text{ and } c = F(A) - q(A).
\]

Using the submodularity of \( F \), we can fully characterize the subdifferential at any \( A \subseteq V \), and moreover optimize over it using Edmonds’ greedy algorithm.
Table 2.1: Specific members of the superdifferential $\partial F(X)$ [32]. The names for the supergradients are from Iyer, Jegelka & Bilmes [33].

<table>
<thead>
<tr>
<th>$i \in X$</th>
<th>$\hat{q}^X(i)$</th>
<th>$\check{q}^X(i)$</th>
<th>$\overline{q}^X(i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i \notin X$</td>
<td>$\hat{q}^X(i)$</td>
<td>$\check{q}^X(i)$</td>
<td>$\overline{q}^X(i)$</td>
</tr>
</tbody>
</table>

Lemma 2.3 ([22, Lemmas 6.5 and 6.7(b)]). If $F: 2^V \to \mathbb{R}$ is submodular, then $\partial F(A) = (-\mathcal{P}(F^A)) \times \mathcal{P}(F^A)$.

On the other hand, characterizing $\partial F(A)$ is provably hard also for submodular functions. Namely, as shown by Iyer & Bilmes [19, Lemma 6.5], checking membership in $\partial F(A)$ is unfortunately NP-hard. Nevertheless, as shown by Iyer & Bilmes [32] and Iyer, Jegelka & Bilmes [33], we can leverage submodularity to define specific members of $\partial F(A)$ at any $A \subseteq V$. We provide three such supergradients in Table 2.1. These supergradients can be used to design inner approximations of $U(F)$ by taking their convex hull. In the remaining chapters we will make use of the convex hull of the bar supergradients, which we now define.

Definition 2.21 (Iyer & Bilmes [19, § 6.3.1]). Given a submodular function $F: 2^V \to \mathbb{R}$, we define the upper bar polyhedron as

$$\overline{U}(F) = \text{Conv} \{(q^A, F(A) - \bar{q}^A(A)) \mid A \subseteq V\} \subseteq U(F). \quad (2.20)$$

These polyhedra can be furthermore simplified when the function $F$ is decomposable, i.e., can be written as a sum $F(A) = \sum_{j=1}^m F^j(A)$ for some submodular functions $F^j$. This is important both for analytical reasons, but also for algorithmic, as many functions that are used in practice do admit such a factorization. Before we state the results, let us introduce the following Minkowski operations.

Definition 2.22. Given two subsets $X$ and $Y$ of some vector space over $\mathbb{R}$ and a scalar $\alpha \in \mathbb{R}$, we define the following operations

$$X + Y = \{x + y \mid x \in X, y \in Y\} \quad \text{and} \quad \alpha X = \{\alpha x \mid x \in X\}.$$ 

Theorem 2.9 ([22, §4.2]). Let $F^1, F^2, \ldots, F^m$ be submodular functions with ground set $V$, and let $\mathcal{D}$ be some lattice on $V$. Then, we have that

$$\mathcal{B}(\sum_{j=1}^m F^j \mid \mathcal{D}) = \sum_{j=1}^m \mathcal{B}(F^j \mid \mathcal{D}). \quad (2.21)$$
Theorem 2.10 ([22, Theorem 6.8]). Let $F^1, F^2, \ldots, F^m$ be submodular functions with ground set $V$. Then, for all $A \subseteq V$, we have that

$$\partial \sum_{j=1}^m F_j(A) = \sum_{j=1}^m \partial F_j(A). \quad (2.22)$$

Example 2.11 (Fujishige [22, §2.3(1)]). As a concrete example of how we can use the above results, let us describe the base polytope associated to a directed graph cut function, which is given as

$$F(A) = \sum_{i \to j \in E} w_{i \to j}[A \cap \{i, j\} = \{i\}].$$

Not that $F$ decomposes over $|E|$ functions, one per each edge. As the components $F_{i \to j}(A) = w_{i \to j}[A \cap \{i, j\} = \{i\}]$ are functions over two variables $\{i, j\}$, their base polytopes have the form

$$\mathcal{B}(F_{i \to j}) = \{(x_i, x_j) \mid x_i + x_j = 0, x_j \leq w_{i \to j}, x_j \leq 0\}.$$ 

After eliminating one of the variables using the equality constraint, it can be further simplified to

$$\{(x_i, -x_i) \mid x_i \in [0, w_{i \to j}]\}.$$ 

Finally, we can apply Theorem 2.9 and write $\mathcal{B}(F)$ as

$$\mathcal{B}(F) = \{q \in \mathbb{R}^n \mid \forall i \in V : q_i = \sum_{e \in \delta^+(i)} x_e - \sum_{e \in \delta^-(i)} x_e, \text{ and } \forall e \in E : x_e \in [0, w_e]\},$$

where $\delta^+(i)$ is the set of all outgoing edges at $i \in V$, and $\delta^-(i)$ consists of all incoming edges at $i \in V$. Hence, the base polytope of a cut function corresponds to all flow boundaries achievable in the underlying graph (see [22, §2.3(1)]).

2.5 Matroids

A class of objects very closely related to submodularity are matroids.

Definition 2.23 (Matroid [34]). A non-empty collection of sets $\mathcal{I} \subseteq 2^V$ is called a matroid if it satisfies the following conditions

\begin{itemize}
  \item [3] We are omitting the co-ordinates that are not in the domain of $F_{i \to j}$ as those must be equal to zero.
1. (downward closure) if \( A \subseteq B \subseteq V \) and \( B \in \mathcal{I} \) then \( A \in \mathcal{I} \), and

2. (exchange property) \( \forall A, B \in \mathcal{I} \) such that \( |A| < |B| \) there

\[ \exists i \in B - A: A \cup \{i\} \in \mathcal{I}. \]

The members of \( \mathcal{I} \) are called independent sets, and the maximal independent sets with respect to set inclusion are called bases.

**Example 2.12** (Linear matroids). Perhaps the most prominent example are the linear matroids, defined as follows. Given a set of \( n \) vectors \((z_1, z_2, \ldots, z_n)\) in some vector space, the matroid is defined as the set of all subsets \( A \) of \( V \) such that the vectors \( \{z_i: i \in A\} \) are linearly independent. This also motivates why the elements on \( \mathcal{I} \) are called independent sets.

**Example 2.13** (Partition matroids). Let \( V_1, V_2, \ldots, V_m \) be a partition of \( V \). Then, the set

\[ \mathcal{I} = \{ A \subseteq V \mid \forall j \in \{1, 2, \ldots, m\}: |A \cap V_j| \leq k_j \} \]

for any integers \( k_1, k_2, \ldots, k_m \) is a matroid.

**Example 2.14** (Graphic matroids). Let \( G = (V, E) \) be an undirected graph. Then, the collection of subsets of edges \( E \) that are cycle-free forms a matroid, called the graphic matroid.

The connection to submodularity can established via their rank function.

**Definition 2.24** (Matroid rank function). Given a matroid \( \mathcal{I} \subseteq 2^V \), its rank function is defined as

\[ F_{\mathcal{I}}(A) = \max\{|B| \mid B \in \mathcal{I} \text{ and } B \subseteq A\}. \]

A classical result is that the above function is submodular (see e.g. [22, §2.1]). Moreover, note that the marginal gains of the rank function are binary, which means that the base polytope will have integral vertices. Specifically, we have the following result.

**Theorem 2.11** ([22, §2.2]). If \( \mathcal{I} \subseteq 2^V \) is a matroid with bases \( B \), then

\[ \text{Conv} \mathcal{I} = \mathcal{P}(F_{\mathcal{I}}) \cap [0, \infty)^n, \text{ and} \]

\[ \text{Conv} B = \mathcal{B}(F_{\mathcal{I}}) \cap [0, \infty)^n, \]

where we define \( \text{Conv} X \) to be the convex hull of all vectors in \( X \).

Hence, Edmonds’ algorithm when applied to \( F_{\mathcal{I}} \) finds a maximally weighted basis of the matroid. For example, when we apply this to the graphic matroid, we obtain Prim’s spanning tree algorithm.
2.6 SUBMODULAR MINIMIZATION AND THE LOVÁSZ EXTENSION

The problem that we discuss next is that of unconstrained submodular minimization, i.e., solving $\min_{A \subseteq V} F(A)$. Many important problems can be cast as the minimization of a submodular objective, ranging from image segmentation \cite{35, 36} to clustering \cite{37}.

Example 2.15 (Semantic image segmentation \cite{38}). One noteworthy instance of submodular minimization is the problem of image segmentation in computer vision (for a survey of this problem see Kolmogorov & Rother \cite{38}). Namely, we are given as input an image $x \in \mathbb{R}^{n \times 3}$, and the task is to find the set of pixels $A \subseteq V$ that are occupied by an object. Moreover, we would furthermore prefer the set $A$ to be spatially consistent. One common approach to encourage such configurations is to (i) define a graph over the image $G = (V, E)$ (typically a grid-graph as the one shown in Figure 2.3) by connecting neighboring pixels, (ii) specify weights over the edges, and then (iii) solve the following min-cut problem

$$
\min_{A \subseteq V} \sum_{\{i,j\} \in E} w_{\{i,j\}} \left[ \left| \{i,j\} \right| = 1 \right] + \sum_{i \in A} v_i \cdot 
$$

The pixel scores $v_i$ are typically learned from data, while the pairwise weights can be either learned or set using some heuristic. For example, as the pairwise weights should represent a notion of similarity, if we have a feature vector $\theta_i \in \mathbb{R}^d$ for each $i \in V$, one can use weights of the form $w_{\{i,j\}} = \exp(-\|\theta_i - \theta_j\|^2)$.

A very important object in the analysis of this problem will be the support function over the base polytope, which we now formally define.

Definition 2.25. Given a submodular function $F : 2^V \rightarrow \mathbb{R}$, we define its Lovász extension $f_L : \mathbb{R}^n \rightarrow \mathbb{R}$ as $f_L(x) = \sup_{q \in B(F)} q^\top x$.

We can immediately see that $f_L$ is convex, as it is a supremum over linear functions. It is also piecewise-linear and satisfies several other properties, some of which we state below.

Lemma 2.4 (\cite[Proposition 3.1]{20}). The Lovász extension satisfies the following properties

1. If $F$ and $G$ are submodular with Lovász extensions $f_L$ and $g_L$, then the Lovász extension of $F + G$ is $f_L + g_L$.

2. For any $\lambda \geq 0$, the Lovász extension of $\lambda F$ is $\lambda f_L$. 
3. The Lovász extension is positively homogeneous, i.e., \( f_L(\alpha x) = \alpha f_L(x) \) for all \( \alpha \geq 0 \).

4. For any permutation \( \sigma : V \to V \), the Lovász extension is linear on the cone

\[
\{ x \in \mathbb{R}^n \mid x_{\sigma(1)} \leq x_{\sigma(2)} \leq \ldots \leq x_{\sigma(n)} \}.
\]

We can use the properties above, together with how Edmonds’ algorithm computes the maximizer, to write the Lovász extension of several function classes in an analytical form.

**Example 2.16** (Modular functions). The Lovász extension of a modular submodular function \( F(x) = x^\top \theta \) is given by \( f_L(x) = x^\top \theta \).

**Example 2.17** (Concave of cardinality). For a function of the form \( F(A) = h(|A|) \) for some concave \( h : [0,n] \to \mathbb{R} \), let us denote the marginal gains by

\[
\Delta = (h(1), h(2) - h(1), \ldots, h(n) - h(n-1)).
\]

Then, the Lovász extension is given by

\[
f_L(x) = \sum_{i=1}^{n} x_i^\downarrow \Delta_i,
\]

where \( x^\downarrow \) has the same set of values on its co-ordinates as \( x \), but sorted in a non-ascending order. For example, if we take \( F(A) = \min\{|A|, 1\} \), then the above formula reduces to \( f_L(x) = \max_{i=1}^{n} x_i \).

**Example 2.18** (Graph cuts). Similarly to the derivation of its base polytope, we can decompose the graph-cut function over its edges and then apply Lemma 2.4.
For an undirected graph cut on two edges \{i, j\} with weight \(w\), the Lovász extension is given as
\[
\begin{align*}
f_L(x_1, x_2) &= w \times \max\{x_1, x_2\} + (-w) \times \min\{x_1, x_2\} \\
&= w(\max\{x_1, x_2\} - \min\{x_2, x_2\}) \\
&= w|x_1 - x_2|.
\end{align*}
\]
Hence, for a general undirected graph, the Lovász extension can be computed as
\[
f_L(x) = \sum_{\{i,j\} \in E} w_{ij}|x_i - x_j|.
\]

The importance of the Lovász extension stems from the fact that, perhaps surprisingly, it agrees with convex envelope on \([0, 1]^n\). This in turn has direct implications for the minimization problem due to Theorem 2.2, as formalized in the following claims.

**Theorem 2.12** ([20, §5.1]). The Lovász extension \(f_L\) and the convex extension \(f_\vee\) agree on \([0, 1]^n\), i.e., \(\forall x \in [0, 1]^n : f_L(x) = f_\vee(x)\).

**Theorem 2.13** ([22, Lemma, 20, Proposition 3.7]). We have that
\[
\min_{A \subseteq V} F(A) = \min_{x \in [0,1]^n} f_L(x),
\]
and moreover,
\[
\argmin_{x \in [0,1]^n} f(x) = \text{Conv}\{1_A \mid A \in \argmin_{A \subseteq V} F(A)\},
\]
where \(1_A \in \{0, 1\}^n\) is the indicator vector with ones at those positions corresponding to the elements of \(A\) and zeros elsewhere.

We additionally have the following characterization of the set of minimizers of a submodular function.

**Theorem 2.14** ([22, Lemma 2.1, 39]). If \(F : 2^V \to \mathbb{R}\) is a submodular function, then the collection \(A^* = \argmin_{A \subseteq V} F(A)\) of minimizers of \(F\) is a lattice.

The first polynomial algorithm due to Grötschel, Lovász & Schrijver [40] for submodular minimization used the ellipsoid method on problem (2.24), but it is too slow to be useful in practice. Several combinatorial algorithms have been also developed (see e.g. Cunningham [41] and Orlin [42]), the fastest being that of Lee, Sidford & Wong [43] with a running time of \(O(\tau n^3 \log^2 n + n^4 \log^{O(1)} n)\), where \(\tau\) is the cost of evaluating the function. They all use the following dual problem and exploit the properties of the base polytope.
Lemma 2.5 ([22, Lemma 14.2]). The problem dual to \( \min_{x \in [0,1]^n} f_L(x) \) is

\[
\max \sum_{q \in B(F)} \min_{i=1}^n \{0, q_i\},
\]

and strong duality holds so that the above is equal to \( \min_{A \subseteq V} F(A) \).

Moreover, if there exists some \( q \in B(F) \) and \( A \subseteq V \) that satisfy

\[
\{i \mid q_i \leq 0\} \subseteq A \subseteq \{i \mid q_i < 0\} \text{ and } y(A) = F(A),
\]

then \( A \) is a minimizer of \( F \).

2.6.1 The minimum norm problem

Another approach, that also solves a continuous problem, and results in more practical algorithms, is that of finding the minimum norm problem inside \( B(F) \) [44]. This is a smooth continuous problem that has a unique solution due to the strict convexity of the objective.

Definition 2.26 (Minimum-norm point [22, §7.1(a)]). The minimum norm point of a submodular function \( F \) is defined as the element of the base polytope with the smallest norm, or equivalently as

\[
q^* = \arg\min_{q \in B(F)} \frac{1}{2} \|q\|^2.
\]

Theorem 2.15 ([20, Proposition 8.1]). The dual of the minimum norm problem is equal to

\[
- \min_{x \in \mathbb{R}^n} f_L(x) + \frac{1}{2} \|x\|^2,
\]

where remember that \( f \) is the Lovász extension of \( F \). Moreover, the optimal primal and dual points are related as \( x^* = -q^* \).

The discrete solution can be easily extracted by thresholding the optimum, as shown by the following theorem.

Theorem 2.16 ([22, Lemma 7.4]). Let \( q^* \) be the optimal solution to problem (2.26), and define the following sets

\[
A_- = \{v \mid v \in V \text{ and } q^*_v < 0\}, \text{ and } A_0 = \{v \mid v \in V \text{ and } q^*_v \leq 0\}.
\]

Then \( A_- \) and \( A_0 \) are the unique minimal and maximal minimizers of \( F \).
2.6.1.1 Frank-Wolfe

Fujishige, Hayashi & Isotani [45] suggest solving the above problem using the algorithm of Wolfe [46], shown in the generalized version by Frank & Wolfe [47] in Algorithm 2, which is a natural choice as we have a very fast linear oracle over the base polytope. This algorithm repeatedly linearizes the objective around the current iterate and moves towards the optimum of the linear approximation over the domain. While there are several strategies for choosing the step size, it turns out that even this simple variant enjoys a convergence guarantee.

Theorem 2.17 (Jaggi [48, Theorem 1]). If \( h(q) \) is \( \lambda \)-smooth, so that its gradient is \( \lambda \)-Lipschitz continuous, then for each \( k \geq 1 \) the \( k \)-th iterate \( q_k \) of Algorithm 2 satisfies

\[
h(q_k) - h(q^*) \leq \frac{2}{k+2}D^2\lambda,
\]

where \( q^* \) is the optimum, and \( D \) is the diameter of the domain.

This algorithm has been shown to be more efficient than the combinatorial algorithms, despite the fact that it has only a pseudo-polynomial running time guarantee [49], stemming from the presence of the diameter of the domain in the bound. The algorithm used by Fujishige, Hayashi & Isotani [45] uses the fully corrective variant of the Frank-Wolfe, which we do not cover here.

Algorithm 2 The Frank-Wolfe algorithm minimizing \( h \) over \( B(F) \).

```plaintext
1: procedure FRANK-WOLFE(F, q^1, h(\cdot), \epsilon)
2:     \triangleright The objective \( h \) must be convex, and \( \epsilon \) is the tolerated accuracy.
3:     for \( k \leftarrow 1, 2, \ldots, T \) do
4:         Pick \( s \in \arg\min_{q \in B(F)} q^\top \nabla h(q^k) \)
5:         if \( (q^k - s)^\top \nabla h(q^k) \leq \epsilon \) then
6:             return \( q^k \) \triangleright Upper bound on accuracy
7:         else
8:             \( q^{k+1} = (1 - \gamma_k)q^k + \gamma ks \)
9:             \( \gamma_k = \frac{2}{k+2} \)
10:        end if
11:    end for
12: end procedure
```

2.6.1.2 Solution via minimization

Sometimes, we might be interested in computing the minimum norm point even for functions for which we do have efficient minimization algorithms.
For example, it is used for the computation of proximal operators of structured norms \([50]\), and as a sub-routines for solving the minimum norm problem of decomposable functions (see e.g. \([51]\)). The following theorem makes the first connection between cardinality-penalized submodular minimization and the min-norm point.

**Theorem 2.18** ([20, Propositions 8.2 and 8.3]). Denote by \(A^\alpha\) any minimizer of \(\arg\min_{A \subseteq V} F(A) + \alpha |A|\). Then, if \(\alpha \leq \beta\) it holds that \(A^\beta \subseteq A^\alpha\), and moreover, the optimal solution \(x^*\) of problem (8.13) has coordinates

\[ x_i^* = \sup\{\alpha \in \mathbb{R} \mid i \in A^\alpha\}. \]

For example, in the case of graph-cuts, we can efficiently minimize \(F(A) + \alpha |A|\) for all \(\alpha\) using a single parametric max-flow problem (Gallo, Grigoriadis & Tarjan \([52]\)).

Another possibility for solving the min-norm problem is the divide-and-conquer algorithm of Groenevelt \([53]\), which performs at most \(n\) submodular minimizations. Namely, this algorithm minimizes

\[
\min_{q \in \mathcal{B}(F)} \sum_{i=1}^{n} h_i(q_i),
\]

where each \(h_i\) is a convex function. In Algorithm 3 we present this method as explained by Bach \([20, \S 9.1]\). Moreover, the algorithm can be generalized to minimize any fully separable convex function \(h(q) = \sum_{i=1}^{n} h_i(q_i)\) over \(q \in \mathcal{B}(F)\). Note that in the case of the min-norm problem, we have \(h_i(q_i) = \frac{1}{2} \|q_i\|^2\), and the optimal \(q\) in the second step has coordinates equal to \(F(V)/n\).

**Algorithm 3** The divide-and-conquer algorithm for minimizing \(\sum_{i=1}^{n} h_i\) over \(\mathcal{B}(F)\).

1: procedure \textbf{DIVIDEANDCONQUER}(F)
2: \(q \leftarrow \arg\min_{q \in \mathbb{R}^n, q(V)=F(V)} \sum_{i=1}^{n} h_i(q_i)\).
3: \(A^* \leftarrow \arg\min_{A \subseteq V} F(A) - \sum_{i \in A} q_i\).
4: if \(F(A^*) = q(A^*)\) then
5: \(\text{return } q\)
6: else
7: \(q_A \leftarrow \text{DIVIDE-CONQUER}(F^A)\)
8: \(q_{V-A} \leftarrow \text{DIVIDE-CONQUER}(F_A)\)
9: \(\text{return } (q_A, q_{V-A})\)
10: end if
11: end procedure
2.6.1.3 Special cases

There exist several cases that admit fast specialized algorithms for the computation of their min-norm point. We would like to point out that the algorithms typically minimize

$$\arg\min_{q \in \mathcal{B}(F)} \frac{1}{2} \|q - z\|^2,$$

i.e., they compute the projection of a given point $z$ on $\mathcal{B}(F)$, or solve the corresponding dual

$$- \arg\min_{x \in \mathbb{R}^n} f_L(x) + \frac{1}{2} \|x - z\|^2.$$

Note that these problems are exactly the min-norm point problem and its dual corresponding to the submodular function $F'(A) = F(A) - z(A)$.

Example 2.19 (Concave of cardinality). In this case, the base polyhedron is a permutahedron, for which we can leverage the projection of Lim & Wright [54]. As shown by Suehiro et al. [55, Lemma 1], for this class of functions, if $z_i \leq z_j$, then the same will be true for the projection of $z$ onto the base polyhedron. Hence, we can significantly reduce the number of inequalities and solve the problem using a single sort and isotonic regression, for a total complexity of $O(n \log n)$.

Example 2.20 (Threshold potentials). Stobbe & Krause [26] show that the corresponding base polytope is an intersection of a hyperplane and a box, and provide an $O(n \log n)$ deterministic algorithm for computing the projection.

Example 2.21 (Graph cuts). The min-norm problem for graph-cuts has received significant interest, primarily due to its use in computer vision. Namely, the dual problem becomes

$$\min_{x \in \mathbb{R}^n} \sum_{\{i,j\} \in E} w_{\{i,j\}} |x_i - x_j| + \frac{1}{2} \|x - z\|^2,$$

which, if the graph is a grid, can be seen as denoising the image $z$ by encouraging consistency across the edges $E$. This method is also known as anisotropic image denoising (see e.g. [56]). The same problem also comes up in statistics when using proximal algorithms for objectives with graph-fused lasso regularizers [57, 58]. If the graph is a chain, the problem can be solved using the taut string algorithm of Condat [59] in linear time. For the general case, one can also use splitting techniques (e.g., Jegelka, Bach & Sra [51] and Barbero & Sra [60]), which decompose the graph into manageable components for which we know how to solve the min-problem, and iteratively combine their solutions.
2.6.2 Graph-representable functions

While the combinatorial and Fujishige-Wolf algorithms can handle arbitrary submodular functions, their application is limited to only moderately sized ground sets. A class of submodular functions that admits much faster algorithms while encompassing many useful functions, are graph-representable functions. Namely, the family of functions consisting of a graph cut plus a modular term can be efficiently solved using classical min-cut/max-flow algorithms [36]. The idea is then to write other functions as graph-cuts with the help of auxiliary nodes, over which we also minimize.

**Definition 2.27** (Jegelka, Lin & Bilmes [61, §2]). A function $F: 2^V \rightarrow \mathbb{R}$ is graph-representable if there exists some graph-cut function $G: 2^{V \cup U} \rightarrow \mathbb{R}$ s.t.

$$\forall A \subseteq V : F(A) = \min_{B \subseteq U} G(A \cup B).$$

Then, to minimize $F$ we minimize $G$ over its complete domain including the auxiliary variables by computing a minimum cut, and extract a minimizer of $F$ by intersecting the minimizer of $G$ with $V$ ([61, Lemma 1]). For example, both the facility location function and threshold potentials can be represented in this way, as shown by Jegelka, Lin & Bilmes [61].

**Example 2.22.** To show how such a construction is done, consider the threshold potential $F(x) = \min\{\alpha, x^\top w\}$ for some $w \geq 0$ and $\alpha \geq 0$, which was the first class of functions for which such a representation was used by Kohli, Ladicky & Torr [25]. Namely, they write this function as

$$F(x) = \min\{\alpha, \sum_{i=1}^n w_i x_i\}$$

$$= \min_{u \in \{0,1\}} (1-u)\alpha + u \sum_{i=1}^n w_i x_i$$

$$= \alpha \alpha + \min_{u \in \{0,1\}} u(-\alpha) + \sum_{i=1}^n uw_i x_i,$$

which is can be represented as a graph as long as $w \geq 0$. Namely, each non-modular term $ux_i$ can be written as $(1-x_i)(1-u) + w_i x_i + w_i u - 1$, that can be easily represented using undirected edge $\{u, x_i\}$ and a modular term.
2.7 SUBMODULAR MAXIMIZATION

Submodular functions are used not only for modelling costs, but also to capture many natural utility functions. For example, the set cover problem (Example 2.4) is a classical instance of submodular maximization. It is also a well-studied example of perhaps the most prominent setting — that of maximizing polymatroid functions under cardinality constraints. This family of problems has found numerous applications, including experimental design [62], document summarization [63], and representation learning [64]. While the problem is NP-hard ([15]), it has a very simple algorithm that is guaranteed to produce a set with utility within a constant of the optimum. Namely, Nemhauser, Wolsey & Fisher [15] have shown that if we repeatedly greedily add the item with the largest marginal gain, we will always be within \(1 - 1/e \approx 63\%\) of the optimum. This algorithm is presented in Algorithm 4, and we now formally state its guarantee.

**Theorem 2.19** (Nemhauser, Wolsey & Fisher [15]). If \(F: 2^V \rightarrow \mathbb{R}\) is polymatroid, the solution \(A^*\) returned from Algorithm 4 is as \((1 - 1/e)\)-factor approximation, i.e., it satisfies

\[
F(A^*) \geq (1 - 1/e) \max_{A \subseteq V, |A| \leq k} F(A).
\]

**Algorithm 4** Greedy submodular maximization.

1: procedure GreedyMaximization\((F, k)\)
2: Define \(A = \emptyset\)
3: for \(j \leftarrow 1, 2, \ldots, k\) do
4:   Pick \(i^* \in \arg\max_{i \in V - A} F(i \mid A)\)
5:   if \(F(i^* \mid A) \leq 0\) then
6:     break
7:   else
8:     \(A \leftarrow A \cup \{i^*\}\)
9:   end if
10: end for
11: return \(A\)
12: end procedure

The algorithm can be also naturally adapted to optimize over the intersection of \(k\) matroids, in which case the approximation factor is \(1/(1 + k)\) [15, 65]. The only change that we have to make in the greedy algorithm is to consider at each step only those items whose addition would not violate any of the \(k\) matroid constraints.
2.7.1 The continuous greedy algorithm

An alternative approach to the problem of polymatroid maximization under a matroid constraints is the continuous greedy algorithm of Calinescu et al. [65]. They provide an algorithm that extends the $1 - 1/e$ approximation to an arbitrary matroid membership constraint, but using a radically different approach. Namely, it is based on the multi-linear extension $f_{\sim}$ of the function, which is optimized over the convex-hull of all elements of the matroid, which remember (Theorem 2.11) that can be written as $M = P(G_\mathcal{I}) \cap [0, \infty]^n$, where $G_\mathcal{I}$ is the rank function of the matroid. The algorithm proceeds as follows.

1. Find an $(1 - 1/e)$-factor approximate solution $\mu^*$ to $\max_{\mu \in M} f_{\sim}(\mu)$.

2. Round $\mu^*$ to a set $A$ satisfying $F(A) \geq f_{\sim}(\mu^*)$ using the pipage rounding method of Ageev & Sviridenko [66].

Let us explain in more detail how the first step is solved. Namely, we have that ([65, p. 6])

$$\partial_{x_i} f_{\sim}(x) = E_{A \sim x_{-i}} [F(i | A)], \tag{2.28}$$

where $x_{-i}$ is the distribution over $V - \{i\}$, such that element $j$ is sampled with probability $x_j$. Then, to maximize $f_{\sim}$ over $M$, the algorithm starts at $x_0 = 0$ and then iteratively (i) estimates the gradient $\hat{g}$ of $f_{\sim}$ around $x_0$ using (2.28), and (ii) computes the new iterate as

$$x_k = x_{k-1} + \delta \max_{\mu \in M} \mu^\top \hat{g},$$

where the optimizer is computed using Edmonds’ algorithm and $\delta$ is the step size. Note that, while this algorithm has a flavour similar to Frank-Wolfe, it differs in that it does not take a convex combinations of the current iterate and the linear optimizer.

2.7.1.1 Non-monotone maximization

The problem of maximizing submodular functions that are not necessarily monotone has also been studied in the literature. For example, Buchbinder et al. [67] provide a randomized combinatorial algorithm that achieves a $1/2$-approximation guarantee.
2.7.2 $M^\natural$-Concavity

A special sub-family of submodular functions that can be easily maximized are $M^\natural$-functions, which are defined as follows.

**Definition 2.28** ($M^\natural$-Concavity [68, §5.1]). A function $F: 2^V \to \mathbb{R}$ is said to be $M^\natural$-concave if for all $A, B \subseteq V$ and $i \in A - B$ we have either

1. $F(A) + F(B) \leq F(A - i) + F(B + i)$, or

2. there exist some $j \in B - A$ such that

$$F(A) + F(B) \leq F(A - i + j) + F(B + i - j).$$

**Lemma 2.6** ([68, §5.1]). The greedy algorithm (Algorithm 4) run with $k = n$ will return an exact maximizer if $F$ is $M^\natural$-natural concave.

While having this very nice maximization algorithm, this function class is very brittle as it is not closed even under addition, with the notable exception of adding a modular function. Its most prominent examples is the following generalization of matroid rank functions.

**Example 2.23** ([69, Theorem 1.2]). Let $\mathcal{I} \subseteq 2^V$ be a matroid and let $w: V \to [0, \infty)$ be a weighting function. The function $F: 2^V \to \mathbb{R}$ given as $F(A) = \max \{\sum_{i \in B} w(i) \mid B \in \mathcal{I} \text{ and } B \subseteq A\}$, called a weighted rank function, is $M^\natural$-concave.

Another important property of this class is that we can also characterize the superdifferential using a quadratic number of inequalities.

**Theorem 2.20** ([19]). If $F$ is $M^\natural$-concave, then we can characterize the superdifferential at any $A \subseteq V$ as

$$\partial^F(A) = \{q \in \mathbb{R}^n \mid (\forall i \in A): q_i \leq F(i \mid A - \{i\}), \quad (\forall j \notin A): q_j \geq F(j \mid A), \quad (\forall i \in A, j \notin A): q_i - q_j \leq F(A) - F(A \cup \{j\} - \{i\})\}.$$
The problem of developing efficient computational methods for approximate inference in discrete models has received significant interest in statistics, machine learning and statistical physics. In this chapter, we will formally set up the computational problems that are central to these distributions, state some known hardness results, and introduce several established frameworks used to develop tractable approximation methods.

We will consider distributions over $n$ categorical random variables $X_1, X_2, \ldots, X_n$, each of them taking on values in the label set $\mathcal{L} = \{0, 1, \ldots, k-1\}$. The index set will be denoted by $V = \{1, 2, \ldots, n\}$. Formally, we will analyze models of the form

$$P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \frac{1}{Z} \exp(-F(x))) \nu(x), \quad (3.1)$$

where $F: \mathcal{L}^n \to \mathbb{R}$ is the energy function, and $\nu: \mathcal{L}^n \to \{0, 1\}$ is the base measure. Note that the configurations that have high probability under $P$ necessarily have lower energies. For example, the most probable configurations of $P$, also called maximum a posteriori (MAP) states, are the minimizers of $F$ over $\{x \in \mathcal{L}^n \mid \nu(x) = 1\}$. The quantity $Z$, called the partition function, ensures that the distribution is normalized and sums up to 1, so it is defined as

$$Z = \sum_{x \in \mathcal{L}^n} \exp(-F(x))) \nu(x). \quad (3.2)$$

Note that the distribution is invariant under shifting $F$ by a scalar, as it would be absorbed into the partition function $Z$, which is constant with respect to $x$. The incorporation of a base measure is necessary to allow for distributions that are defined only over a subset of $\mathcal{L}^n$. For example, if we would like to condition on some event $\mathcal{A} \subseteq \mathcal{L}^n$ we can use $\nu(x) = [x \in \mathcal{A}]$. We define the support of $P$ as

$$\mathcal{X} = \{x \in \mathcal{L}^n \mid \nu(x) = 1\}. \quad (3.3)$$

Furthermore, we will throughout the thesis use the following convention

$$P(x) = P(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n),$$

1 The same could have been achieved by letting $F$ take on infinite values, but we will stick to this notation.
as standard in the approximate inference literature — namely, we use uppercase letters for the random variables, and lowercase for the values they take. We also define for any distribution $P$ its Shannon entropy as follows.

**Definition 3.1** (Shannon [70]). For any distribution $P$ we define its entropy as

$$H(P) = - \sum_{x \in \mathcal{X}} P(x) \log P(x). \tag{3.4}$$

Let us consider several models that result from different energy functions.

**Example 3.1** (Ising model [71]). A standard model in statistical physics is the Ising model, used for modelling systems of particles (e.g., gases, magnets). Namely, assuming that we have $n$ binary random variables $X_1, \ldots, X_n$ each one with two possible states 0 and 1, the Ising model is defined as

$$P(x) = \frac{1}{Z} \exp(- \sum_{i=1}^{n} \gamma_i x_i - \sum_{i=1}^{n-1} \sum_{j=1}^{i} \beta_{i,j} x_i x_j), \tag{3.5}$$

for some numbers $\gamma_i$ and $\beta_{i,j}$. The case when $\forall i, j: \beta_{i,j} \leq 0$ is called ferromagnetic (attractive, log-supermodular), while the case when $\forall i, j: \beta_{i,j} \geq 0$ is called anti-ferromagnetic (repulsive, log-submodular). In certain special cases many of its properties (including the computation of $Z$) can be solved analytically (see e.g. Baxter [72]), or approximated within any accuracy (Jerrum & Sinclair [73]).

**Example 3.2** (Potts model [74]). A generalization of the Ising model is the Potts model, where each variable can take on $k$ values, and has the following form

$$P(x) = \frac{1}{Z} \exp \left(- \sum_{i=1}^{n} \theta_{i,x_i} - \sum_{i=1}^{n} \sum_{j=1}^{n} \beta_{i,j} \mathbb{1}[x_i \neq x_j] \right) \tag{3.6}$$

for some numbers $\beta_{i,j}$ and $\theta_{i,a}$. Note that if $\beta_{i,j} > 0$, then the model prefers the random variables $X_i$ and $X_j$ to take on the same value, while if $\beta_{i,j} < 0$, then the model prefers that they take on different values.

### 3.1 The Inference Problems and Their Hardness

The are two fundamental questions associated with any discrete distribution. The first one is that of computing the partition function $Z$. From the formula (3.2), we see that a naïve computation would need to add up $k^n$
numbers, which is prohibitory expensive even for toy models. Its importance stems from the fact that it is an indivisible quantity from the log-likelihood, which is used for both model selection and learning. Namely, if we are given \( m \) samples \( x_1, x_2, \ldots, x_m \) and want to choose a model that best explains them, the log-likelihood of the data under the distribution \( P \) is

\[
- \sum_{i=1}^{m} F(x_i) - n \log Z.
\]

Because we can typically easily evaluate the energy \( F \), the main problem is the computation of \( \log Z \). In the search for approximate methods for estimating \( Z \), one is often interested in lower- and upper approximations. Formally, we would like to have a method that computes some approximation \( \hat{Z} \) that comes with a guarantee that \( Z \leq \hat{Z} \) or \( Z \geq \hat{Z} \). Then, we can create provable intervals for the log-likelihoods and design better informed model selection strategies.

The second problem is that of computing marginal probabilities under \( P \). Specifically, we want to compute the quantities

\[
P(X_{i_1} = x_{i_1}, X_{i_2} = x_{i_2}, \ldots, X_{i_m} = x_{i_m})
\]

for some indices \( i_1, i_2, \ldots, i_m \) and corresponding values \( x_{i_j} \in \mathcal{L} \). We will predominantly focus our attention to the unary and pairwise marginals, defined as

\[
P(X_i = a) = \sum_{x \in \mathcal{X}: x_i = a} P(x) = \frac{\sum_{x \in \mathcal{X}: x_i = a} \exp(-F(x))}{\sum_{x \in \mathcal{X}} \exp(-F(x))}, \quad \text{and}
\]

\[
P(X_i = a, X_j = b) = \sum_{x \in \mathcal{X}: x_i = a, x_j = b} P(x) = \frac{\sum_{x \in \mathcal{X}: x_i = a, x_j = b} \exp(-F(x))}{\sum_{x \in \mathcal{X}} \exp(-F(x))}.
\]

Note that in the binary case (\( k = 2 \)), the unary and pairwise marginals correspond to the first and second-moments of the distribution, i.e., \( \mathbb{E}_P[x] \) and \( \mathbb{E}_P[xx^\top] \) respectively. These two problems are closely related — namely, if we can efficiently compute \( Z \), we can also compute the marginals by dividing the partition function obtained after multiplying in \( v(x) = [x_i = a] \) with \( Z \). Moreover, many approximation algorithms for the partition function also yield as a side-product estimates of the marginals. Hence, we will see that even if we design algorithms with the primary motivation of approximating \( Z \), we typically also obtain estimates of the marginals.
3.1.1 Computational hardness

Unfortunately, the above problems are notoriously hard to compute or approximate. Consider the Ising model from Example 3.1 — in the ferromagnetic case ($\forall i, j: \beta_{i,j} \geq 0$) Jerrum & Sinclair [73, Theorem 5] have shown that computing $\mathcal{Z}$ is $\#P$-hard, but can be approximated within any accuracy if a certain condition on the weights $\gamma_i$ is satisfied. Goldberg & Jerrum [75, Theorem 1] have proven that the general ferromagnetic case is hard to approximate. The anti-ferromagnetic ($\beta > 0$) case has also been shown to be hard to approximate by Jerrum & Sinclair [73, Theorem 14].

3.2 Exponential family models

A class of distributions that comes with very powerful theory, and is also being widely applied in practice, is the exponential family. In the remaining of this section we will formally define it and show some of its properties. The presentation follows Wainwright & Jordan [76, §3]. We furthermore state the results only for the specialized case of discrete variables.

**Definition 3.2** ([76, §3.2]). Let $X_1, X_2, \ldots, X_n$ be discrete variables taking on values in $\mathcal{L}$, let $\nu: \mathcal{L}^n \rightarrow \{0, 1\}$ be a base measure, and let $\phi: \mathbb{R}^n \rightarrow \mathbb{R}^d$, which is called the sufficient statistic. The associated exponential family consists of all distributions of the form

$$P_\theta(x_1, x_2, \ldots, x_n) = \exp \left( \theta^\top \phi(x) - A(\theta) \right) \nu(x),$$

where $\theta \in \mathbb{R}^d$ is a vector holding the parameters, and $A: \mathbb{R}^d \rightarrow \mathbb{R}$ is the log-partition function defined as

$$A(\theta) = \log \sum_{x \in \mathcal{X}} \exp(\theta^\top \phi(x)).$$

Given a sufficient statistic $\phi$, we will associate to any distribution $\mathbb{P}$ the expectation of $\phi$, i.e., $\mu = \mathbb{E}_{x \sim \mathbb{P}}[\phi(x)] \in \mathbb{R}^d$, whose entries are called the mean parameters (Wainwright & Jordan [76, §3.4.1]). The set of all realizable mean parameters forms an object that will be of crucial importance in the analysis of exponential families.

**Definition 3.3** (Marginal polytope [76, §3.4.1]). The marginal polytope $\mathbb{M}$ associated with an exponential family is defined as

$$\mathbb{M} = \{\mu \in \mathbb{R}^d: \exists \text{ distribution } \mathbb{P} \text{ on } \mathcal{X} \text{ s.t. } \mathbb{E}_{x \sim \mathbb{P}}[\phi(x)] = \mu\}.$$
Lemma 3.1 ([76, §3.4.1]). The marginal polytope can be written as the convex hull of the image of $\mathcal{X}$ under $\phi$, i.e.,

$$M = \text{Conv}\{\phi(x) \in \mathbb{R}^d \mid x \in \mathcal{X}\} = \text{Conv}\{\phi(x) \in \mathbb{R}^d \mid \nu(x) \neq 0\}.$$ 

The log-partition function has a close connection to the mean parameters and their covariance.

Theorem 3.1 ([76, Proposition 3.1]). The log-partition function $A(\theta)$ is convex, has derivatives of all orders, and its first two derivatives are given by

$$\nabla A(\theta) = \mathbb{E}_{x \sim \mathcal{P}_\theta}[\phi(x)], \quad \text{and}$$

$$\nabla^2 A(\theta) = \text{Cov}_{x \sim \mathcal{P}_\theta}[\phi(x)]. \quad (3.9)$$

Not only does the function $\nabla A: \mathbb{R}^n \rightarrow \mathbb{M}$ map the parameters of the model to the corresponding mean parameters, but it also covers almost all of them. Hence, the exponential family is sufficiently rich to model most realizable mean parameters.

Theorem 3.2 ([76, Theorem 3.3]). The mapping $\nabla A: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is onto relint $M$. In other words, for each $\mu \in \text{relint} \mathbb{M}$ there exists some $\theta(\mu) \in \mathbb{R}^d$ such that

$$\mathbb{E}_{x \sim \mathcal{P}_{\theta(\mu)}}[\phi(x)] = \mu.$$ 

Moreover, the elements of this family have the particular property that among all distributions with mean parameters $\mu \in \mathbb{M}$, the exponential family with parameter $\mathcal{P}_{\theta(\mu)}$ is the one with largest Shannon entropy. This is also a standard way to motivate their definition (see e.g. [76, §3.1]).

Theorem 3.3 ([76, §3.5.2, §3.1]). Among all distributions $\mathcal{P}$ satisfying $\mathbb{E}_\mathcal{P}[\phi(x)] = \mu \in \mathbb{M}$, the exponential family parameter $\mathcal{P}_{\theta(\mu)}$ is the one with maximal Shannon entropy.

The partition function can be written as an optimization problem involving its convex conjugate, which has a clear probabilistic explanation. Moreover, this variational characterization is typically the starting point when developing approximate inference algorithms.

Theorem 3.4 ([76, Theorem 3.4]). The conjugate function of $A$ is equal to

$$A^*(\mu) = \begin{cases} -\mathcal{H}[\mathcal{P}_{\theta(\mu)}] & \text{if } \mu \in \text{relint} \mathbb{M}, \\ +\infty & \text{if } \mu \notin \mathbb{M} \end{cases}, \quad (3.11)$$

where $\mathcal{H}$ is the Shannon entropy of the distribution.
Theorem 3.5 ([76, Theorem 3.4]). The log-partition function can be written in terms of its conjugate as

\[ A(\theta) = \sup_{\mu \in M} \mu^\top \theta - A^*(\mu), \] (3.12)

with the supremum in (3.12) attained uniquely at \( \mu^* \in M \) satisfying the moment-matching condition

\[ \mu^* = \mathbb{E}_{x \sim P_\theta}[\phi(x)]. \]

While we have so far stated the results for arbitrary sufficient statistics, there is a particular choice of \( \phi \) that is widely used in the literature and results in easily interpretable mean parameters.

Definition 3.4 (Markov random field (MRF) [76, §3.4.1]). Given a graph \( G = (V, E) \), we define the Markov random field as the exponential family distribution resulting from the following sufficient statistic

\[ \phi(x) = \left[ \left[ x_i = a \right]\right]_{i \in V} \times \left[ \left[ x_i = a \right]\left[ x_j = b \right]\right]_{\{i,j\} \in E}, \]

where \( \times \) is the direct product concatenating its arguments.

Hence, the over-complete representation consists of all indicator functions over all variables and edges of the graph. To keep the notation succinct, we follow the common convention to index the co-ordinates of \( \phi(x) \) as

\[ \phi(x)_{i;a} = \left[ x_i = a \right] \text{ and } \] (3.13)
\[ \phi(x)_{i,j;a,b} = \left[ x_i = a \right]\left[ x_j = b \right]. \] (3.14)

We will also use the same indexing for the parameter vector \( \theta \) and the mean parameters \( \mu \). Moreover, for any \( i \in V \) we will denote by \( \mu_i \) the subvector \( [\mu_{i,a}]_{a \in \mathcal{L}} \), and we will similarly collect all elements \( [\mu_{i,j,a,b}]_{a,b \in \mathcal{L}} \) in \( \mu_{i,j} \). Note that in this representation, the mean parameters, given by

\[ \mu_{i;a} = \mathbb{E}_{P_\theta}[\left[ X_i = a \right]] = \mathbb{P}_\theta(X_i = a), \text{ and } \] (3.15)
\[ \mu_{i,j;a,b} = \mathbb{E}_{P_\theta}[\left[ X_i = a \right]\left[ X_j = b \right]] = \mathbb{P}_\theta(X_i = a, X_j = b), \] (3.16)

are exactly the unary and pairwise marginals of the distribution.
3.2.1 Examples

To clarify the concepts introduced in this section, we present several models that belong to the exponential family.

Example 3.3. Let us consider the Markov random field over graph with no edges. The sufficient statistic $\phi(x)$ consists of the $m = nk$ entries $\phi_{i,a}$. To describe the marginal polytope, consider $\Phi = \{\phi(x) : x \in X\}$. Note that for each $i \in V$, the set $\Phi_i = \{\phi(x)_i : x \in X\}$ consists of the $k$ vectors that have all coordinates zero except for a single one. Hence, we have that $\mathcal{M} = \text{Conv } \Phi$ is equal to $n$ copies of $\text{Conv } \Phi_i$, which is itself the probabilistic simplex. The partition function can be easily seen to be a sum of soft-maxes

$$A(\theta) = \sum_{i \in V} \log \sum_{a \in L} \exp(\theta_{i,a}).$$

Similarly, the negative entropy $A^*(\mu)$ is equal to

$$A^*(\mu) = \sum_{i \in V} \sum_{a \in L} \mu_{i,a} \log \mu_{i,a}.$$  

Example 3.4. The Potts model from Example 3.2 correspond to MRFs with parameters $\theta_{i,a} = -\gamma_{i,a}$ and $\theta_{i,j,a,b} = -\beta_{i,j}[a \neq b]$.

Example 3.5. There is a particular MRF instance, that has an analytic description of both its marginal polytope and its entropy. Namely, if the graph $G = (V, E)$ contains no cycles (i.e., is a tree), then we can describe its marginal polytope using the so-called local agreement conditions (for a proof see [76, Proposition 4.1])

$$\mathcal{L}(G) = \{\mu \in [0, \infty)^m \mid (\forall i \in V) : \sum_{a \in L} \mu_{i,a} = 1, \text{ and}$$

$$(\forall j \in \delta(i), a \in L) : \mu_{i,a} = \sum_{b \in L} \mu_{i,j,a,b}\}. \quad (3.17)$$

In other words, if we constrain the marginals to be consistent across the edges, then global consistency will also be satisfied. The entropy can also be written in a closed form as

$$-A^*(\mu) = \sum_{i \in V} (1 - |\delta(i)|)H_i(\mu_i) + \sum_{\{i,j\} \in E} H_{i,j}(\mu_{i,j}), \quad (3.18)$$

where $\delta : V \rightarrow E$ maps a node to its incident edges,

$$H_i(\mu_i) = -\sum_{a \in L} \mu_{i,a} \log \mu_{i,a}, \text{ and}$$

$$H_{i,j}(\mu_{i,j}) = -\sum_{a,b \in L} \mu_{i,j,a,b} \log \mu_{i,j,a,b}. \quad (3.19)$$
The partition function can be then computed using its conjugate representation (3.12), which can be efficiently solved using a message-passing dynamic programming algorithm also known as belief propagation.

Unfortunately, it is in general very challenging to give a succinct representation of the marginal polytope for complicated models. As a concrete example, where we do understand the polytope very well, we consider a class of objects covered in the previous chapter.

**Example 3.6.** Consider the binary case so that each variable can takes either zero or one, and let $\mathcal{I} \subseteq 2^V$ be a matroid (Definition 2.23). We will use as a base measure and sufficient statistic $v(x) = [x \text{ is a base of } \mathcal{I}]$ and $\phi(x) = x$ respectively. Then, from Lemma 3.1, we know that the marginal-polytope consists of the convex hull of all indicator bases in the matroid, which from Theorem 2.11 we know to be equal to $\mathcal{B}(F_{\mathcal{I}})$, where $F_{\mathcal{I}}$ is the rank function of the matroid.

### 3.3 Approximate Inference Techniques

#### 3.3.1 Variational inference for exponential families

As we have already hinted, the starting point for developing inference algorithms for exponential families is the variational form of the partition function

$$A(\theta) = \sup_{\mu \in \mathcal{M}} \mu^\top \theta - A^*(\mu).$$

The problem is that we have neither a concise representation of $\mathcal{M}$, nor do we know how to compute the entropy $-A^*(\mu)$. By choosing various approximations $\overline{\mathcal{M}}$ of $\mathcal{M}$ and $\overline{H}$ of $-A^*(\mu)$, we will arrive at several algorithms with different guarantees. Namely, we solve the approximate problem

$$\overline{A}(\theta) = \sup_{\tau \in \overline{\mathcal{M}}} \tau^\top \theta + \overline{H}(\tau),$$  \hspace{1cm} (3.20)

where we have also changed the notation for the marginals from $\mu$ to $\tau$ to indicate that fact that they might not be valid mean parameters as we only use an approximation to the marginal polytope.

In the remaining of this section we will assume that the model is represented as a MRF over some graph $G = (V, E)$. Then, one common approximation for $\mathcal{M}$ is the local polytope $\overline{\mathcal{M}} = \mathcal{L}(G)$, which would be indeed correct if $G$ is a tree. Otherwise, we only have the inclusion $\mathcal{L}(G) \supseteq \mathcal{M}$, as all marginals must be locally consistent. We could also use the tree entropy
approximate inference techniques and try to solve the resulting (in general non-convex) problem. Yedidia, Freeman & Weiss [77] have shown that iterating the fixed point equations resulting from the KKT conditions gives the same updates as classical belief propagation ([78]). However, except in special conditions (e.g. log-supermodularity Ruozzi [79]), the resulting approximation is not guaranteed to bound the partition function.

To ensure that we obtain an upper bound, several methods use an outer bound of the marginal polytope function $\mathbb{M} \supseteq \mathbb{M}$ together with a concave entropy approximation $\underline{H}$ that dominates the true entropy everywhere over $\mathbb{M}$. Moreover, due to the concavity of the approximation, the resulting variational problem can be efficiently solved to within arbitrary precision, so that at an optimum we will obtain an upper bound. For example, we can again use the local-polytope approximation together with entropy approximations of the form

$$\underline{H}(\tau, \rho) = \sum_{i \in V} \rho_i H_i(\tau_i) + \sum_{\{i,j\} \in E} \rho_{i,j} H_{i,j}(\tau_{i,j}), \tag{3.21}$$

where the weights $\rho_i$ and $\rho_{i,j}$ are so-called counting numbers, and the functions $H_i$ and $H_{i,j}$ are defined in (3.19). The counting numbers have to be carefully chosen to ensure the concavity of the entropy approximation over $\mathbb{M}$. Heskes [80] and Meshi et al. [81] give conditions on how to choose them in a principled manner. A particular choice is to choose the edge weights from the base polytope corresponding to the graphic matroid $\mathcal{I}$ of the graph $G$, i.e., the convex hull of all spanning trees of $G$, and fix the vertex weights to $\rho_i = 1 - \sum_{e \in \delta(i)} \rho_e$. This method is known as tree-reweighted belief propagation and is due to Wainwright, Jaakkola & Willsky [82]. Moreover, because the weights appear only linearly in (3.21), it is convex if treated as a function of $\rho$. Hence, Wainwright & Jordan [83] furthermore suggest to also optimize over the weights in a double-loop algorithm to tightening the bound.

An approximation for the case of binary MRFs, that also yields an upper bound, is the relaxation of Wainwright & Jordan [83] which solves a
semi-definite program (SDP). For any $\mu \in \mathbb{R}^m$ they define the following symmetric matrix

$$M(\mu) = \begin{bmatrix}
1 & \mu_{1;1} & \mu_{2;1} & \cdots & \mu_{n;1} \\
\mu_{1;1} & \mu_{1;1} & \mu_{1,2;1,1} & \cdots & \mu_{1,n;1,1} \\
\mu_{2;1} & \mu_{2,1;1,1} & \mu_{2;1} & \cdots & \mu_{2,n;1,1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mu_{n;1} & \mu_{n,1;1,1} & \mu_{n,2;1,1} & \cdots & \mu_{n;1}
\end{bmatrix}$$

The main observation is that if $\mu$ are the true marginals, then the above matrix is exactly equal to $E_{P_\theta}[[1, x][1, x]^T]$, which has to be positive definite. Moreover, if we have $\mu_{i;1}$ and $\mu_{i,j;1,1}$ for all $i, j \in V$, we can figure out the remaining entries of $\mu$ from the normalization and marginalization constraints. We thus arrive at the following outer relaxation on $M$

$$M = \{ \tau \in \mathbb{R}^m \mid M(\tau) \succeq 0, \forall i \in V: \tau_{i;1} = 1 - \tau_{i,0}, \forall \{i, j\} \in E: \tau_{i,j;0,0} = 1 + \tau_{i,j;1,1} - \tau_{i,1} - \tau_{j,0}, \forall \{i, j\} \in E: \tau_{i,j;1,0} = \tau_{i,j;1,1} - \tau_{j,1}, \text{ and } \forall \{i, j\} \in E: \tau_{i,j;0,1} = \tau_{i,j;1,1} - \tau_{i,1}\}. \quad (3.23)$$

They also suggest further tightening of the above relaxation, which we do not cover here. To obtain an upper bound from the variational problem, we also need an upper bound on the entropy, and Wainwright & Jordan [76, Lemma 2] suggest the following one

$$- A^*(\mu) \leq \overline{H}(\mu) = \frac{1}{2} \log \det(M(\mu)) + \frac{1}{12} \text{blkdiag}[0, I] + \frac{n}{2} \log(2\pi e), \quad (3.24)$$

where blkdiag creates a block diagonal matrix with the given arguments concatenated along the diagonal. Note that the resulting problem is that of maximizing a log-determinant over the positive semi-definite cone, which can be solved using interior point methods.

Finally, we can also obtain lower-bounds on the partition function by using an inner approximation $\overline{M} \subseteq M$ chosen in a careful way so that we can always evaluate $A^*(\mu)$ for $\mu \in \overline{M}$. This method is known as mean-field, and to illustrate its most common form let us consider the case where $\overline{M}$
corresponds to the family of fully factorized distributions (i.e., those with zero pairwise weights). Namely, the set is given by

\[ \mathbb{M}(V) = \{ \mu \in [0, \infty)^m \mid (\forall i \in V) : \sum_{l=1}^{k} \mu_i = 1, \text{ and} \] \[ (\forall j \in \delta(i), (a, b) \in \mathcal{L}^2) : \mu_{i,a \mid j,b} = \mu_{i,a} \mu_{j,b} \}, \tag{3.25} \]

so that for any \( \mu \in \mathbb{M} \) we can calculate the entropy as

\[ -A^*(\mu) = - \sum_{i \in V} \sum_{a \in \mathcal{L}} \mu_{i,a} \log \mu_{i,a}. \]

Because \( \mathbb{M} \) is non-convex (see e.g. [76, Example 5.4]) the resulting problem

\[ \max_{\mu_{i,a} \in \Delta^{k-1}} \sum_{i \in V} \sum_{a \in \mathcal{L}} \mu_{i,a} \theta_{i,a} + \sum_{\{i,j\} \in \mathcal{E}} \sum_{a,b \in \mathcal{L}} \mu_{i,a} \mu_{j,b} \theta_{i,j,a,b} - \sum_{i \in V} \sum_{a \in \mathcal{L}} \mu_{i,a} \log \mu_{i,a}, \]

where \( \Delta^{k-1} \) is the \((k-1)\)-dimensional simplex, becomes very challenging. Nevertheless, we can easily optimize with respect to one block \( \mu_i \) at a time. Hence, the most common approach is that of iteratively updating the above bound in a coordinate ascent manner, with faster algorithms existing for special choices of pairwise potentials (e.g. Krähenbühl & Koltun [84]).

### 3.3.2 Mean-field

The mean-field method can be also easily generalized to arbitrary models not necessarily written in an exponential family form. Moreover, this is also the approach used when deriving the variational mean-field bound for continuous Bayesian models. Given two distributions \( \mathbb{P} \) and \( \mathbb{Q} \), such that \( \mathbb{P}(x) = 0 \) implies \( \mathbb{Q}(x) = 0 \), the Kullback-Leibler (KL) divergence between them (see e.g. [85] for an information theoretic interpretation) is given as

\[ \text{KL}(\mathbb{Q} \parallel \mathbb{P}) = \mathbb{E}_\mathbb{Q} \left[ \log \frac{\mathbb{Q}(x)}{\mathbb{P}(x)} \right]. \tag{3.26} \]

If we assume that \( \mathbb{P} \) is of the form (3.1) and noting that the divergence is always non-negative, we arrive at the following inequality

\[ 0 \leq \text{KL}(\mathbb{Q} \parallel \mathbb{P}) = \mathbb{E}_\mathbb{Q} \left[ \log \frac{\mathbb{Q}(x)}{\mathbb{P}(x)} \right] \tag{3.27} \]

\[ = \log Z + \mathbb{E}_\mathbb{Q} \left[ \log \frac{\mathbb{Q}(x)}{\exp(-F(x))} \right] \tag{3.28} \]

\[ = \log Z - \mathbb{H}[\mathbb{Q}] + \mathbb{E}_\mathbb{Q}[F], \tag{3.29} \]
which yields the bound \( \log Z \geq H[Q] - E_Q[F] \). The main idea is then to optimize the above bound with respect to the distribution \( Q \) over some family of distributions \( Q \ni Q \). The considered distributions \( Q \) have to be chosen in a way that would let us efficiently estimate and optimize the lower bound. For example, if all variables are binary, \( \nu \equiv 1 \) and \( Q \) consists of all factorized distributions of the form

\[
Q_q(x) = \prod_{i=1}^n q_i^{x_i}(1 - q_i)^{1-x_i},
\]

for some \( q \in [0,1]^n \), the bound becomes

\[
\log Z \geq \max_{q \in [0,1]^n} \sum_{i=1}^n (-q_i \log q_i - (1 - q_i) \log(1 - q_i)) - f_\sim(q),
\]

where remember that \( f_\sim \) is the multi-linear extension of \( F \). Then, performing coordinate ascent with respect to \( q_i \) by setting its partial derivative to yields the condition

\[
q_i \leftarrow \sigma(-g_i),
\]

where \( \sigma(z) = 1/(1 + e^{-z}) \) is the sigmoid function and \( g_i = \partial_{q_i} f_\sim(q) \), which was given in equation (2.28).

### 3.3.3 Perturb-and-MAP

This recently popularized technique, originally suggested by Papandreou & Yuille [86], relies on a different representation of the partition function. We follow the presentation of Balog et al. [87], based on the continuous case analysis by Maddison, Tarlow & Minka [88]. We first introduce for each \( x \in \mathcal{X} \) a Poisson clock \( T_x \) that rings after a time drawn from an \( \text{Exponential}(\exp(-F(x))) \) distribution. The time until the first clock rings, a minimum of \(|\mathcal{X}|\) exponential distributions, is again exponential, i.e.,

\[
\min_{x \in \mathcal{X}} T_x \sim \text{Exponential}(\sum_{x \in \mathcal{X}} \exp(-F(x))) = \text{Exponential}(Z).
\]

Taking the negative log of both sides and using the fact that the negative logarithm of \( \text{Exponential}(\lambda) \) is a \( \text{Gumbel}(\log \lambda) \), we obtain

\[
-\log \min_{x \in \mathcal{X}} T_x \sim -\log \text{Exponential}(Z)
\]

\[
\max_{x \in \mathcal{X}} -\log T_x \sim \text{Gumbel}(\log Z)
\]

\[
\max_{x \in \mathcal{X}} \text{Gumbel}(-F(x)) \sim \text{Gumbel}(\log Z).
\]
Because the mean of a Gumbel(\(\eta\)) is \(\eta + c\), where \(c\) is the Euler–Mascheroni constant, we can write \(\log Z\) as the following expectation

\[
\log Z = \mathbb{E}_{\gamma(x) \sim \text{Gumbel}(-c)} \max_{x \in \mathcal{X}} \gamma(x) - F(x). \tag{3.30}
\]

Hence, we have to repeatedly perturb the energy by a random perturbation \(\gamma(x)\) and compute the MAP configuration of this new distribution. However, this representation does not render the estimation of the partition function any easier, because we have to sample a total of \(|\mathcal{X}| = k^n\) perturbations. Fortunately, similarly to how we have approximated the partition function by approximating the variational form, we can also simplify the perturbative distribution to obtain estimates of the partition function. Hazan & Jaakkola [89] suggest adding low-order perturbations to estimate the partition function, whose result for the first-order perturbations we present below.

**Theorem 3.6** ([89, Theorem 2]). Let \(\{\gamma_{i,a}\}_{i \in V, a \in \mathcal{L}}\) be a set of independent Gumbel random variables. Then, we have the following upper bound on the partition function

\[
\log Z = \mathbb{E}_{\gamma_{i,a} \sim \text{Gumbel}(-c)} \max_{x \in \mathcal{X}} \sum_{i \in V} \gamma_{i,x_i} - F(x). \tag{3.31}
\]

### 3.3.4 Sampling

While we will solely focus on variational techniques in this thesis, we would like to point to an orthogonal approach, that is also widely used in practice. Namely, if we can sample independent points \(x_1, x_2, \ldots, x_m\) from \(P\), we can easily estimate any expectation \(\mathbb{E}_{x \sim P}[g(x)]\) as \(\frac{1}{m} \sum_{j=1}^{m} g(x_j)\), which will converge to the correct value as \(m \to \infty\) thanks to the law of large numbers. However, sampling from arbitrary distributions can be a very challenging task and no easier than the inference problems we have presented in this chapter. A common approach, called Markov Chain Monte Carlo (MCMC, see e.g. Brooks *et al.* [90]), is to instead generate a sequence of correlated points \(x_1, x_2, \ldots, x_m\) that, informally, in the limit correspond to samples from \(P\). More specifically, one creates an ergodic Markov chain ([91]) with stationary distribution \(P\), in which case the empirical estimate converges almost surely to the true expectation as we take the number of points to infinity. Hence, due to the Markovian property the \(j + 1\)-th point is generated using information only from the previous \(j\)-th point. There are several strategies how to generate such distributions, including Metropolis-Hastings (Metropolis *et al.* [92] and Hastings [93]) and
Gibbs sampling (Geman & Geman [94]). The latter is particularly interesting to discrete models, as it can be easily described and implemented. Namely, at each iteration $j$ we pick a coordinate $i' \in N$ and we pick the next point $x_{j+1}$ as following

$$\begin{align*}
x_{j+1,i} &= \begin{cases} 
x_{j,i} & \text{if } i \neq i', \text{ or} \\
\sim \mathbb{P}(X_{i'} | X_{i''} = x_{j,i''} \text{ for } i'' \neq i') & \text{otherwise.}
\end{cases}
\end{align*}$$

Hence, we either copy the coordinate of the previous point, or we sample it from a simple categorical distribution taking on $k$ values. While these methods provably give correct answers in the limit, it is in general very challenging to establish convergence, and we might have to run them for a very long time to get correct answers.
We begin this chapter by covering the necessary background on divergence measures in Sections 4.1 and 4.2. The material in Section 4.3 is a simple reformulation of the well-known infinity divergence. To the best of our knowledge, the connection to the relaxations and polyhedra presented in Section 4.4, and the connection to TRWBP in Section 4.5 are novel. In Section 4.3.1 we show a connection between the infinite divergences and global lower- and upper-estimators of discrete functions, an idea that was behind all of the papers in Section 1.7, starting with

- Djolonga, J. & Krause, A. *From MAP to Marginals: Variational Inference in Bayesian Submodular Models* in NIPS (2014),

Moreover, we have already used the upper and lower polyhedra to perform inference in


while the connection between the inclusive infinity divergence and the lower polyhedra has been shown in


In Section 4.6, we present the work of Shpakova & Bach [95].

### 4.1 Introduction

A popular approach to designing approximate inference algorithms is that of minimizing divergence measures. The idea is to find a distribution $Q$ that is as close as possible to the target distribution $P$, but remains computationally tractable. We typically constrain $Q$ to be a member of some parametric family of distributions $Q$ that can capture the characteristics of $P$ we are interested in, but are also simpler than $P$. Having computed $Q$, we

we can use it in lieu of \( P \) in the problems we are concerned with — for example, we could compute optimal actions by minimizing the Bayes risk over \( Q \), or estimate some of its statistical properties. In this way we cast the problem of inference to that of minimizing some metric of disagreement over a family of distributions.

There are many possible ways of quantifying the distance between two distributions. Depending on the discrepancy measure that we select, we will end up with approximate distributions that have different qualitative characteristics. For example, some divergences might put more weight on how the moments of the distribution are captured, while others might focus on the modes. It is thus very important to understand what kind of biases the chosen divergence entails, as that will have an impact on the downstream tasks where the approximation is used. The choice of a distance measure also has practical consequences — for example, while there are many algorithms and techniques for optimization, they can not be always applied directly to the problems that arise from the above strategy. Sometimes it can be very hard to even estimate the objective function up to additive and multiplicative constants. Hence, to ensure that we can practically solve the resulting problem, we typically have to further constrain the set of distances we choose from.

The most prominent example of this inference strategy is the mean-field approach from Section 3.3.2. Namely, we have seen that if we minimize the Kullback-Leibler divergence, we obtain, in addition to an approximation of \( P \), a lower bound on the partition function. Moreover, the optimization problem can be also approached in a principled manner, as we can estimate the objective and its derivatives by sampling. Minka [96] has analyzed in more depth the general problem, where one uses an arbitrary \( \alpha \)-divergence [97, 98]. He also discusses what kind of distributions are preferred by these distances, and proposes a scheme for their minimization using a generalized expectation propagation framework.

In this chapter we devote our attention to a specific measure of discrepancy — the infinite Rényi divergence. We first discuss the qualitative properties of the approximations that result from minimizing this distance. Then, we will more closely examine the resulting computational problems, and show how they relate to some of the concepts we have seen in Chapter 2. Specifically, we will make a close connection between the inference problems under this divergence, and the polyhedra and continuous extensions we have introduced. The results in this chapter will serve as a cornerstone for the remaining of the thesis, where we develop inference
algorithms for several families of distributions by utilizing their properties when applied to these divergences.

Then, we will apply this divergence to classical MRFs. Perhaps surprisingly, we will see that finding the closest tree-structured distribution recovers tree-reweighted belief propagation, a classical inference algorithm motivated from a different perspective. Finally, we present a result due to Shpakova & Bach [124] that relates the infinity divergence with the Perturb-and-MAP scheme.

4.2 BACKGROUND — RÉNYI DIVERGENCES

One commonly used family of divergences is due to Rényi [99], who has also shown that they are the unique family of functions that satisfy a set of information theoretic postulates. In this chapter, we consider distributions defined over some finite set $\mathcal{X}$.

**Definition 4.1 (Rényi [99, §3]).** Define the Rényi divergence of order $\alpha \in (0, \infty) - \{1\}$ between $P$ and $Q$ as

$$D_\alpha(P \parallel Q) \equiv \frac{1}{\alpha - 1} \log \sum_{x \in \mathcal{X}} P(x)^\alpha Q^{1-\alpha}(x)$$

$$= \frac{1}{\alpha - 1} \log E_{x \sim Q}[(P(x)/Q(x))^{\alpha}]. \quad (4.1)$$

This family includes many commonly used divergences, for example we can obtain the KL divergence as a special limiting case [99]

$$D_1(P \parallel Q) \equiv \lim_{\alpha \to 1} D_\alpha(P \parallel Q) = D_{KL}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \frac{P(x)}{Q(x)}. \quad (4.2)$$

We would like to point out that these divergences are in general not metrics as they are neither symmetric nor satisfy the triangle inequality. They are however non-negative and are zero if and only if $P = Q$ ([99, §3]). Moreover, while in the literature these divergences are also discussed when $\alpha$ is negative, we have decided to keep the divergences always non-negative and swap their arguments whenever necessary.

Another set of divergences that are commonly used are the $\alpha$-divergences [97, 98], which take on a similar form

$$\frac{1}{\alpha(\alpha - 1)} \sum_{x \in \mathcal{X}} P(x)^\alpha Q^{1-\alpha}(x),$$
and are closely related to the Rényi divergences, as shown by Cichocki & Amari [100, §2.2]. Most notably, the set of minimizers of both metrics for the same \( \alpha \) coincides.

It turns out that, similarly as in the mean-field case, by minimizing these divergences we are optimizing bounds on the partition function.

**Theorem 4.1** ([96, Theorem 2]). Assume that \( P(x) = \overline{P}(x) / Z \) for some non-negative and possibly un-normalized function \( \overline{P} : \mathcal{X} \to [0, \infty) \). Then we have the following inequalities

\[
\log Z \leq \frac{1}{\alpha} \log \sum_{x \in \mathcal{X}} P^\alpha(x)Q^{1-\alpha}(x) \quad \text{for all } \alpha > 1, \text{ and}
\]

\[
\log Z \geq -\frac{1}{\alpha - 1} \log \sum_{x \in \mathcal{X}} \overline{Q}^\alpha(x)P^{1-\alpha}(x) \quad \text{for all } \alpha > 0.
\]

Note that because the right hand sides of the above inequalities are monotone functions of \( D_\infty(\overline{P} \parallel Q) \) and \( D_\infty(P \parallel Q) \) respectively, by maximizing the divergences we tighten the corresponding bounds. Note that while these bounds always hold, they can be very hard to exactly evaluate, and their estimation using importance sampling can result in an unwieldy variance.

As we have already argued, it is important to understand the characteristics of the solutions that one obtains from minimizing these divergences. Minka [96], building on the argument of Frey et al. [101], classifies these divergences into two broad families — inclusive or covering, and exclusive or zero-forcing. Informally, a divergence is inclusive if it prefers to cover as much mass as possible from \( P \), and is conservative when it comes to making false negative mistakes, so that it would prefer approximations with higher entropy. The second class of distributions are those with contrasting behaviour — they would rather err on the side of over-confidence by possibly erroneously giving some events probabilities closer to zero and one. Specifically, we observe inclusive behaviour when we minimize \( D_\alpha(\overline{P} \parallel Q) \) for \( \alpha \geq 1 \), and exclusive when we minimize \( D_\alpha(Q \parallel \overline{P}) \) for any \( \alpha \geq 0 \). For example, the canonical examples are the exclusive mean-field \( D_{KL}(Q \parallel P) \) and the inclusive expectation propagation \( D_{KL}(P \parallel Q) \) objectives. This can be also seen if we allow \( P \) to assign zero probabilities. Then, the exclusive divergences will force \( Q \) to be zero wherever \( P \) is, which is also the reason behind their alternative name. Moreover, from Theorem 4.1, note that by minimizing the inclusive divergences we are optimizing upper bounds on the partition function, while by optimizing exclusive divergences we are maximizing lower bounds.
To illustrate this behaviour, we consider the problem of approximating a model with two dependent binary variables $X_1$ and $X_2$. As the approximation family $Q$ we use the set of fully independent distributions, i.e., those of the form $Q(x) = q_1^{x_1}(1 - q_1)^{1-x_1}q_2^{x_2}(1 - q_2)^{1-x_2}$. For a fixed, randomly chosen, distribution $P$ we plot in Figure 4.1 the paths of the optimal distributions $Q_\alpha = \text{argmin}_{Q \in Q} D_\alpha(P \parallel Q)$ and $Q_\alpha = \text{argmin}_{Q \in Q} D_\alpha(Q \parallel P)$. We can see that in the inclusive case the entropy steadily increases, while, in contrast, in the exclusive case it decreases. In other words, the optimal path gravitates towards or away from the center of the unit square, which contains the higher entropy models.

The distances that we will be of primary interested to us are the inclusive and exclusive infinite divergences, which are obtained by taking the limit $\alpha \to \infty$. They have been already discussed by Minka [96], and recently used by Li & Turner [102] for inference in continuous models.

**Definition 4.2** (Van Erven & Harremoës [103, Theorem 6]). Define the Rényi divergence of infinite order between $P$ and $Q$ as

$$D_\infty(P \parallel Q) = \lim_{\alpha \to \infty} D_\alpha(P \parallel Q) = \log \max_{x \in \mathcal{X}} \frac{P(x)}{Q(x)}.$$

Motivated from the discussion above, we will refer to $D_\infty(P \parallel Q)$ and $D_\infty(Q \parallel P)$ as the inclusive and exclusive infinite Rényi divergences re-
spectively. Extrapolating from Figure 4.1, we would expect them to have the most pronounced behaviour of coverage and exclusivity. To obtain a better intuition, we plot in Figure 4.2 their values, in addition to those of the KL distances, on the problem of performing inference on a discrete problem with a single Bernoulli variable. As one would expect, they all achieve their minimum of zero only at the true distribution, as the true distribution is in $Q$. However, they differ significantly once we move away from the optimum. If we analyze their derivatives, we can see that $D_\infty(Q \parallel P)$ and $D_\infty(P \parallel Q)$ have drastically different preferences. Namely, while $D_\infty(P \parallel Q)$ has a smaller gradient towards 0.5, where the more uncertain distributions are, the opposite behaviour is true for $D_\infty(Q \parallel P)$. The same observation can be made for the pair of KL divergences, but the contrast is not as striking. More precisely, in the one dimensional example $D_\infty(P \parallel Q)$ is equal to

$$\log \max\{\frac{p}{q}, \frac{1-p}{1-q}\} = \left\lfloor q < p \right\rfloor \log \frac{p}{q} + \left\lfloor q \geq p \right\rfloor \log \frac{1-p}{1-q}.$$ 

Hence, the divergence becomes infinitely large as we get closer to the extremities — in other words, the divergence acts as a logarithmic barrier
\(- \log \min_x Q(x)\) close to the extremal points. The same argument can be generalized when we also have distributions over \(n\) discrete random variables. Then, the divergence will blow up as we approach distributions that assign zero probability to some atom. On the other hand, \(D_\infty(Q\parallel P)\) can be written as

\[
\log \max\{q, \frac{1-q}{1-p}\} = [q < p] \log \frac{1-q}{1-p} + \frac{1}{p} [q \geq p] \log \frac{q}{p}.
\]

Thus, instead of acting as a barrier, the divergence penalizes by \(\log \max_x Q(x)\) as \(\max_x Q(x) \to 1\), which always remains finite.

To illustrate the divergences on a more complicated example, where \(Q\) does not contain the true distribution, we consider the setting from Figure 4.1. Namely, we want to approximate a coupled distribution \(P\) over two binary variables using a distribution \(Q\) that belongs to the family of fully independent distributions \(Q\). We show in Figure 4.3 the contours of the losses, the corresponding optima, and the true marginals. For the inclusive infinite divergence, we note that the loss becomes infinite at the boundary of the unit square, but that is not the case for the exclusive one. Moreover, from these plots the inclusive divergence seems more computationally tractable due to its simpler loss surface — a point, which we will formally prove in the next section.

### 4.3 The Infinite Divergence for Discrete Models

Having discussed the qualitative behaviour of these divergences, we now turn to their computational aspects. Namely, we will analyze the resulting optimization problems from several angles, that would allow us to make connections to some of the results and approximations from Chapter 2.

Throughout the section we make the same assumption about the form of \(P\) as in Chapter 3, i.e., that it is equal to

\[
P(x) = \exp(-F(x) - \log Z)v(x),
\]

for some energy function \(F: \mathcal{L}^n \to \mathbb{R}\), and base measure \(v: \mathcal{L}^n \to \{0, 1\}\).

The family of approximations we consider is of the form

\[
Q = \{Q = \exp(-Q(x)v(x)/Z_Q) \mid Q \in Q^o\},
\]

i.e., those that have the same form as \(P\), but whose energy function comes from some pre-determined family \(Q^o\). The support of these distribution is the set \(\mathcal{X} = \{x \in \mathcal{L}^n \mid v(x) = 1\}\). We assume that \(Q^o\) has been chosen
**Figure 4.3:** The divergence minimization problems for four different divergences (one per column) on three separate random instances (one per row). The targets $P$ have correlated variables, and we want to approximate them using factorized distributions $Q$. Each point in the square uniquely represents a distribution $Q$ with the corresponding marginals and the contours correspond to the values of the divergences. The red dots are positioned at the true marginals, while the green diamonds are placed at the distribution that minimizes the corresponding divergence.
in a way that we can efficiently perform inference for each $Q \in \mathcal{Q}$, including the computation of $\log Z_Q$. We begin the analysis with the following straightforward reformulation of the infinite divergences.

**Lemma 4.1.** Assume that

$$
\mathbb{P}(x) = \frac{1}{Z} \exp(-F(x)) \nu(x), \quad \text{and} \\
Q(x) = \frac{1}{Z_Q} \exp(-Q(x)) \nu(x) \quad \text{for} \quad Q \in \mathcal{Q}.
$$

Then, the distance from $\mathbb{P}$ to the family $\mathcal{Q}$ can be written as

$$
\log Z + \inf_{Q \in \mathcal{Q}} D_\infty(\mathbb{P} \parallel Q) = \inf_{Q \in \mathcal{Q}^o} \log Z_Q + \sup_{x \in \mathcal{X}} [Q(x) - F(x)], \quad \text{and} \\
(4.5)
$$

$$
\log Z - \inf_{Q \in \mathcal{Q}} D_\infty(Q \parallel \mathbb{P}) = \sup_{Q \in \mathcal{Q}^o} \log Z_Q + \inf_{x \in \mathcal{X}} [Q(x) - F(x)], \quad (4.6)
$$

where note that the right hand sides lower- and upper-bound $\log Z$ respectively as the divergences are non-negative.

**Proof.** Equation (4.6) is equivalent to

$$
\inf_{Q \in \mathcal{Q}^o} \log \sup_{x \in \mathcal{X}} \frac{\mathbb{P}(x)}{Q(x)} = \inf_{Q \in \mathcal{Q}^o} \log \sup_{x \in \mathcal{X}} \exp \left( Q(x) + \log Z_Q - F(x) - \log Z \right) \\
= \inf_{Q \in \mathcal{Q}^o} \log Z_Q + \sup_{x \in \mathcal{X}} [Q(x) - F(x)] - \log Z.
$$

Similarly, equation (4.5) can be expanded as

$$
\inf_{Q \in \mathcal{Q}^o} \log \sup_{x \in \mathcal{X}} \frac{Q(x)}{\mathbb{P}(x)} = \inf_{Q \in \mathcal{Q}^o} \log \sup_{x \in \mathcal{X}} \exp \left( -Q(x) - \log Z_Q + F(x) + \log Z \right) \\
= \inf_{Q \in \mathcal{Q}^o} \log Z_Q - \inf_{x \in \mathcal{X}} [Q(x) - F(x)] + \log Z
$$

which completes the proof after re-arranging the terms. \qed

The fact that we obtain upper and lower bounds also follows from Theorem 4.1, which implies the relationship illustrated in Figure 4.4. The main reason why we have reformulated the bounds in this form, is that they elucidate the computational problems that are involved in the minimization
of the divergences. Specifically, in addition to the requirement that we have to be able to normalize $Q$, we should also know how to solve the following discrete optimization problems

$$\max_{x \in X} [Q(x) - F(x)] \text{ for } D_\infty(P \parallel Q), \text{ and}$$
(4.7)

$$\max_{x \in X} [F(x) - Q(x)] \text{ for } D_\infty(Q \parallel P).$$
(4.8)

The first problem can be seen as computing the MAP after perturbing the energy by $-Q(x)$. Hence, if we can compute the MAP in $P$, we should be able to do so after adding the perturbation if we want to exactly evaluate the bound. In the second case, we arrive at an analogous problem — we are computing the MAP in $Q$ after perturbing its energy by $-F(x)$. We can also understand these problems from another perspective if we think of $Q \in \mathcal{Q}_0$ as a function that approximates the energy $F$ over $\mathcal{X}$. Namely, they are precisely computing the points in the domain $x$ where the gap between $F$ and $Q$ is the largest.

Moreover, assume that the approximate energy $Q_q$ is parametrized by some vector of parameters $q$. Then, if we assume that the maxima in equations (4.7) and (4.8) are unique and do not change under small perturbations of $q$, the two optimization problems locally coincide with

$$\arg\max_q \log Q_q(x^*_+) = \arg\max_q - \log Z_{Q_q} - Q_q(x^*_+) \quad \text{and}$$
(4.9)

$$\arg\min_q \log Q_q(x^*-) = \arg\min_q + \log Z_{Q_q} + Q_q(x^*-) \text{ respectively,}$$
(4.10)

where $x^*_+$ and $x^*_-$ are the optima of (4.7) and (4.8). Hence, the problem of minimizing the inclusive divergence resembles that of maximum likelihood learning over $Q_q$. Specifically, the point at which the likelihood is computed is not sampled, but is computed using an auxiliary optimization problem. For example, if we optimize the loss using a first order method, we will perform updates similarly to those that result from stochastic optimization of the maximum likelihood learning function, but with a particular parameter-dependent choice of the point where the objective is evaluated. The analogous argument can be also made for exclusive divergence,
in that we want to minimize the likelihood of the point that is most over-
estimated under $Q_q$.

### 4.3.1 Connection to global estimators

We can derive the bounds in Lemma 4.1 from a much simpler principle
without referring to any divergence measures. If for some $Q \in \mathcal{Q}_o$
and $c \in \mathbb{R}$ we know that the inequality

$$Q(x) + c \leq F(x)$$  \hskip 1cm (4.11)

holds everywhere on $\mathcal{X}$, then from the monotonicity of the exponential we have that

$$\log Z = \log \sum_{x \in \mathcal{X}} \exp(-F(x)) \leq \log \sum_{x \in \mathcal{X}} \exp(-Q(x) - c) = -c + \log Z_Q.$$  

Moreover, for any fixed $Q$ and $F$ we can also find the $c$ that yields the
best bound. Given that the bound gets tighter as we increase $c$, we want
the largest $c$ that would not violate inequality (4.11). Specifically, it is easy
to see that it is given by $c = \max_{x \in \mathcal{X}} [F(x) - Q(x)]$, which would in turn transform the bound to

$$\log Z_Q - \max_{x \in \mathcal{X}} [F(x) - Q(x)] = \log Z_Q + \min_{x \in \mathcal{X}} [Q(x) - F(x)],$$  

that is exactly equation (4.6).

The other bound can be analogously derived. If $Q(x) + c \geq F(x)$ every-
where, then

$$\log Z = \log \sum_{x \in \mathcal{X}} \exp(-F(x)) \geq \log \sum_{x \in \mathcal{X}} \exp(-Q(x) - c) = -c + \log Z_Q.$$  

Now, the optimal $c$ is $\max_{x \in \mathcal{X}} [F(x) - Q(x)]$, so that the bound becomes

$$\log Z_Q - \max_{x \in \mathcal{X}} [F(x) - Q(x)],$$

which we have seen before as equation (4.5).

The idea to derive bounds by global estimators has been first suggested
by Lafferty & Ravikumar [104] for inference in MRFs. Namely, they pro-
spose to find among all energy functions $Q$ that lower bound $F$ the one that
minimizes $\max_{x \in \mathcal{X}} F(x)/Q(x)$. Additionally, many of the inference strate-
gies that we will show in the upcoming chapters have been first de-
veloped from this perspective. While this viewpoint provides us with a very
straightforward approach to deriving the bounds on the partition function
in contrast to the infinite divergences, it is not clear how to give a qualitative analysis of the resulting approximations.

4.4 THE FULLY FACTORIZED BINARY CASE

In this section we will focus on the binary case, i.e., where we have \( n \) variables \( X_1, X_2, \ldots, X_n \) each one taking on values in \( \{0, 1\} \), so that the support of the distribution \( X \) is a subset of \( \{0, 1\}^n \). As the approximation family we will consider the fully factorized distributions.

Definition 4.3. Define the set \( Q \) of completely factorized distributions as

\[
Q = \{ Q | Q(x) = \exp(-q^\top x - A(-q))v(x) \text{ for some } q \in \mathbb{R}^n \}. \tag{4.12}
\]

Remember that in this notation \( \log Z_Q \) is equal to the log-partition function \( A(-q) \), which is convex and has been analyzed in Section 3.2. We are using the negated parameter \(-q\) for the exponential family, to be consistent with the definition of \( P \). Coincidently, this choice will also result in duals which are easier to interpret. Also note that the random variables are not necessarily fully independent under \( Q \) due to the presence of a base measure.

We have decided to discuss the simplest approximation family, as that is what we will primarily use in the upcoming chapters, and because it furthermore lets us utilize some of the concepts from Chapter 2 without any modifications. The results that we discuss here can be easily translated to arbitrary approximation families.

Note that for this family of approximations, to evaluate the divergences we have to solve the problems \( \min_{x \in X} q^\top x - F(\mathbf{x}) \) and \( \max_{x \in X} q^\top x - F(\mathbf{x}) \). In other words, we should be able to maximize and minimize \( F \) under a linear perturbation. In the second case, we can think of \( q \) as a vector holding negative prices, so that perturbed problem is that of finding set with the lowest energy under a price penalty. In economics, if \(-F\) models a utility, the mapping from \( q \) to the optima of the perturbed problem is also known as the demand correspondence [105, §3.3].

4.4.1 As optimization over general polyhedra

The first connection that we make is to the upper and lower polyhedra we have defined in Definitions 2.3 and 2.4. Namely, we will prove that
the problem of inference using the infinite divergence can be equivalently cast as that of optimizing smooth functions over them. For example, the minimization of the exclusive divergence is equivalent to maximization over the upper polyhedron.

**Theorem 4.2** (Lower bound). If \( Q \) is the family of factorized distributions, we can write the lower bound (r.h.s. of (4.5)) as

\[
\sup_{q \in \mathbb{R}^n} A(-q) + \inf_{x \in \mathcal{X}} [q^\top x - F(x)] = \sup_{(q,c) \in \mathcal{U}(F|\mathcal{X})} A(-q) - c. \tag{4.13}
\]

**Proof.** The first part follows by expanding equation (4.5), which reads

\[
\sup_{q \in \mathbb{R}^n} A(-q) + \inf_{x \in \mathcal{X}} [q^\top x - F(x)].
\]

Using a standard strategy used for solving minimax problems using linear programming, we will introduce an extra variable \( t \) to capture the infimum. This is done by adding the set of constraints

\[
\mathcal{C} = \{(q, t) \mid q^\top x - F(x) \geq t \text{ for all } x \in \mathcal{X}\},
\]

and writing the objective as \( \inf_{(q,t) \in \mathcal{C}} A(-q) + t \). The fact that these optimization problem are equivalent can be easily seen as any optimum \((q^*, t^*)\) must satisfy \( t^* = \min_{x \in \mathcal{X}} [q^*^\top x - F(x)] \). Then the result follows by comparing the definition of \( \mathcal{C} \) with that of \( \mathcal{U}(F | \mathcal{X}) \).

Using an analogous argument, we can also show that minimizing the inclusive divergence reduces to optimization over the lower polyhedron.

**Theorem 4.3** (Upper bound). If \( Q \) is the family of factorized distributions, we can write the upper bound (4.6) as

\[
\inf_{q \in \mathbb{R}^n} A(-q) + \sup_{x \in \mathcal{X}} [q^\top x - F(x)] = \inf_{(q,c) \in \mathcal{L}(F|\mathcal{X})} A(-q) - c \tag{4.14}
\]

**Proof.** The proof is essentially the same as that of Theorem 4.2. We expand the bound (4.6) and obtain

\[
\inf_{q \in \mathbb{R}^n} A(-q) + \sup_{x \in \mathcal{X}} [q^\top x - F(x)],
\]

which using the same trick as before can be written as

\[
\sup_{(q,-t) \in \mathcal{L}(F|\mathcal{X})} A(-q) + t,
\]

from which the result follows. \( \square \)
While these two problems have the same objective function, they are very different. Not only they are defined over different polyhedra, but even more importantly, the inclusive divergence minimizes, while the exclusive one maximizes the objective. Because \( A(-q) - c \) is easily seen to be convex due to the convexity of the log-partition function (Theorem 3.1), we should expect that optimizing the inclusive divergence will be much easier. Note that this agrees with what we have observed for the two-dimensional case in Figure 4.3.

4.4.2 The entropic forms

As the last part of the analysis, we will consider the unconstrained binary case, i.e., when \( \nu \equiv 1 \) and \( \mathcal{X} = \{0, 1\}^n \). We will show that in this case the bounds can be also seen as entropy-regularized MAP optimization. Note that in this case the log-partition function is \( A(-q) = \sum_{i=1}^{n} \log(1 + e^{-q_i}) \), the marginal polytope coincides with \([0, 1]^n\), and the entropy is given by the sum of binary entropies

\[-A^*(\mu) = \mathcal{H}(\mu) = -\sum_{i=1}^{n} (\mu_i \log \mu_i + (1 - \mu_i) \log (1 - \mu_i)).\]

where, with a abuse of notation, we have used \( \mathcal{H}(\mu) = -A^*(\mu) \). This special form is important as it will let us establish a direct connection to the convex and concave envelopes on \([0, 1]^n\), even though the results can be easily extended, by either always keeping track of \( \mathcal{X} \), or by allowing the function \( F \) to take on infinite values. We will first show the result for the lower bound, stemming from the exclusive divergence.

**Theorem 4.4.** If \( Q \) is the family of factorized distributions and \( \mathcal{X} = \{0, 1\}^n \), we can write the lower bound (4.5) as

\[
\sup_{\mu \in \mathcal{M}} \mathcal{H}(\mu) - f_\wedge(\mu).
\] (4.15)
Proof. Starting from the left hand side of Theorem 4.3, we have
\[
\sup_{q \in \mathbb{R}^n} f^*(q) + A(-q) = \sup_{q, \mu \in M} f^*(q) - \mu^T q - A^*(\mu) = -A^*(\mu) + \sup_{q \in \mathbb{R}^n} [-\mu^T q + f^*(q)]
\]
\[
= \sup_{\mu \in M} -A^*(\mu) - \inf_{q \in \mathbb{R}^n} [\mu^T q - f^*(q)]
\]
\[
= \sup_{\mu \in M} \mathcal{H}(\mu) - f^*(\mu).
\]

The representation for the upper bound is derived from Fenchel duality as follows.

**Theorem 4.5.** If \( Q \) is the family of factorized distributions and \( X = \{0, 1\}^n \), we can write the upper bound (4.6) as
\[
\sup_{\mu \in M} \mathcal{H}(\mu) - f^*(\mu). \tag{4.16}
\]

Moreover, if \( q^* \) is the optimum to the primal, then \( \mu^* = -\nabla A(-q^*) \) is the dual optimum.

Proof. Follows directly from Fenchel duality, because \( A \) is continuous everywhere, so that Theorem A.1 translates to
\[
\inf_{q \in \mathbb{R}^n} A(-q) + f^*(q) = \sup_{\mu \in M} -A^*(\mu) - f^*(\mu).
\]

As a corollary, we can state the three equivalent forms that we have derived for the bounds.

**Corollary 1.** If \( Q \) is the family of factorized distributions and \( \nu \equiv 1 \), we have the following variational bounds on the partition function
\[
\log Z \geq \sup_{\mu \in M} \mathcal{H}(\mu) - f^*(\mu) = \sup_{q} A(-q) + f^*(q) = \sup_{(q, c) \in \mathcal{L}(F)} A(-q) - c, \quad \text{and}
\]
\[
\log Z \leq \sup_{\mu \in M} \mathcal{H}(\mu) - f^*(\mu) = \inf_{q} A(-q) + f^*(q) = \inf_{(q, c) \in \mathcal{U}(F)} A(-q) - c.
\]
<table>
<thead>
<tr>
<th>Divergence</th>
<th>Objective</th>
<th>Property</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean-field</td>
<td>(H[\mu] - f_\sim(\mu))</td>
<td>Neither convex nor concave</td>
</tr>
<tr>
<td>(D_\infty(P \parallel Q))</td>
<td>(H[\mu] - f_\lor(\mu))</td>
<td>Concave</td>
</tr>
<tr>
<td>(D_\infty(Q \parallel P))</td>
<td>(H[\mu] - f_\land(\mu))</td>
<td>Difference of concave</td>
</tr>
</tbody>
</table>

**Table 4.1:** The maximization problems over \(\mu \in [0,1]^n\) and their properties for the three divergences.

Moreover, we would like to point out that the entropic problems, including the mean-field one from Section 3.3.2, all have a similar form. Namely, as shown in Table 4.1, they are each minimizing one of the three continuous extensions we have discussed, with the addition of an entropy regularizer. Namely, the regularizer will pull the optimal \(\mu^*\) towards the center of \([0,1]^n\). Moreover, while \(D_\infty(P \parallel Q)\) can be efficiently minimized if we can compute the convex extension of the energy \(F\), that does not seem to be the case for the other two divergences as they are not in general convex optimization problems.

### 4.5 The Markov Random Field Case

As a concrete example, let us consider the problem of inference in MRFs. Namely, we assume that the target distribution \(P\) is an MRF defined over some graph \(G = (V, E)\). As the family of approximate distributions \(Q\) we will use all MRFs defined on a subgraph \(G_T = (V, T)\) of \(G\), i.e., containing only some \(T \subseteq E\) of the edges of \(G\). The edges \(T\) have to be carefully chosen so that we can easily do inference in every \(Q \in Q\), the most common choice being that of a tree graph. Formally, the distributions will be assumed to take on the following forms

\[
P(x) = \exp(-\theta^T \phi(x) - \log Z), \quad \text{and} \quad (4.17)
\]
\[
Q(x) = \exp(-q^T \phi_T(x) - A_T(-q)), \quad (4.18)
\]

where \(\phi: \mathcal{L}^n \rightarrow \mathbb{R}^m\) and \(\phi_T: \mathcal{L}^n \rightarrow \mathbb{R}^{m'}\) are the over-complete sufficient statistics (Definition 3.4) over \(G\) and \(G_T\) respectively. Then, minimizing
the inclusive infinite divergence we obtain from (4.6) the following upper bound

\[ \log Z \leq \inf_{q \in \mathbb{R}^{m'}} A_T(-q) + \sup_{x \in \mathcal{X}} q^T \phi_T(x) - \theta^T \phi(x). \]

We will furthermore define the matrix \( p_T \in \{0, 1\}^{m \times m'} \) mapping parameter vectors in \( \mathbb{R}^{m'} \) to \( \mathbb{R}^m \) by copying the unary weights and the pairwise parameters corresponding to the edges \( T \), while setting the remaining coordinates to zero. Then, we can re-write the above bound as

\[ \inf_{q \in \mathbb{R}^n} A_T(-q) + \sup_{\mu \in \mathcal{M}} (p_T q)^T \mu - \theta^T \mu, \]

where \( \mathcal{M} \) is the marginal polytope of exponential family corresponding to \( \phi(x) \). As we in general cannot represent \( \mathcal{M} \) succinctly, we relax it to an outer bound \( \overline{\mathcal{M}} \supseteq \mathcal{M} \), which results in the following upper bound

\[ \log Z \leq \inf_{q \in \mathbb{R}^{m'}} A_T(-q) + \sup_{\mu \in \overline{\mathcal{M}}} \mu^T (p_T q - \theta), \quad (4.19) \]

whose dual is given in the following claim.

**Theorem 4.6.** The Fenchel dual of problem (4.19) is

\[ \sup_{\mu \in \overline{\mathcal{M}}} -\mu^T \theta - A_T^*(p_T^T \mu). \quad (4.20) \]

**Proof.** Define the function \( g(z) = \sup_{\mu \in \overline{\mathcal{M}}} \mu^T (z - \theta) \), whose conjugate is equal to

\[ g^*(\mu) = \begin{cases} \mu^T \theta & \text{if } \mu \in \overline{\mathcal{M}}, \text{ and} \\ +\infty & \text{otherwise}. \end{cases} \]

Then, we can write the primal as

\[ \inf_{q \in \mathbb{R}^n} A_T(-q) + g(p_T^T q), \]

Now, as both \( A \) and \( g \) are defined everywhere, we can apply Theorem A.2 to obtain the claimed dual. \( \square \)

Remember from Theorem 3.4 that \( A_T^* \) is infinite outside the marginal polytope \( \overline{\mathcal{M}}_T \) corresponding to the sufficient statistic \( \phi_T \). Hence, we implicitly require that the projection of \( \overline{\mathcal{M}} \) onto the coordinates correspond
to the exact marginal polytope of the variational family. For example, this is satisfied when we use the local polytope $\mathcal{M} = \mathcal{L}(G)$ from (3.17) and sub-graph $G_T$ is a tree, in which case we obtain the starting point from [106, §7.1] for deriving tree-reweighted belief propagation (TRWBP [82]). Specifically, if all the weights in TRWBP are fixed on a single tree, then the above bound is the same as the one obtained by TRWBP.

### 4.6 Relationship to Perturb-and-Map

Shpakova & Bach [95] have proven a relationship between the inclusive infinite divergence and the Perturb-and-MAP bound with first-order perturbations (Theorem 3.6) for a specific model family — namely, they have shown that the Perturb-and-MAP bound is always tighter if $F$ is submodular. We present their argument, but extended to any arbitrary binary model. First, they note that the bound from the first order-perturbation is equal to

$$
\mathbb{E}_{\gamma_i \sim \text{Gumbel}(-c)} \left[ \max_{x \in \{0,1\}} \sum_{i=1}^{n} x_i \gamma_i - F(x) \right] =
$$

$$
\mathbb{E}_{\gamma_i \sim \text{Gumbel}(-c)} \left[ \max_{x \in \{0,1\}} \sum_{i=1}^{n} x_i (\gamma_i,1 - \gamma_i,0) - F(x) + \sum_{i=1}^{n} \gamma_i,0 \right] =
$$

$$
\mathbb{E}_{\gamma_i \sim \text{Logistic}(0)} \left[ \max_{x \in \{0,1\}} \sum_{i=1}^{n} x_i \gamma_i - F(x) \right] + \sum_{i=1}^{n} \mathbb{E}_{\gamma_i \sim \text{Gumbel}(-c)} [\gamma_i,0],
$$

where the last equality follows from the fact that the difference of two Gumbels is a logistic distribution. This observation has been also made by Papandreou & Yuille [86]. Then, we can state the formal result.

**Theorem 4.7** (Shpakova & Bach [95]). Define a random vector $\gamma \in \mathbb{R}^n$ whose entries $\gamma_i$ are sampled as $\gamma_i \sim \text{Logistic}(0)$. Then, the following inequality holds

$$
\log Z \leq \mathbb{E}_\gamma [f_\wedge(x) + \gamma^T x] \leq \max_{x \in [0,1]^n} f_\wedge(x) + H(x).
$$
4.6 Relationship to Perturb-and-MAP

Proof.

\[
\mathbb{E}_\gamma \left[ \max_{\mathbf{x} \in \{0,1\}^n} \mathbf{\gamma}^\top \mathbf{x} - F(\mathbf{x}) \right] =
\]

\[
\mathbb{E}_\gamma \left[ \max_{\mathbf{x} \in \{0,1\}^n} \mathbf{\gamma}^\top \mathbf{x} - f_\forall (\mathbf{x}) \right] = \quad \text{(Theorem 2.2)}
\]

\[
\mathbb{E}_\gamma \left[ \max_{\mathbf{x} \in \{0,1\}^n} \mathbf{\gamma}^\top \mathbf{x} - \sup_{(\mathbf{q},c) \in \mathcal{L}(F)} (\mathbf{q}^\top \mathbf{x} + c) \right] = \quad \text{(Theorem 2.1)}
\]

\[
\mathbb{E}_\gamma \left[ \max_{\mathbf{x} \in \{0,1\}^n} \min_{(\mathbf{q},c) \in \mathcal{L}(F)} (\mathbf{\gamma} - \mathbf{q})^\top \mathbf{x} - c \right] = \quad \text{(Theorem A.3)}
\]

\[
\mathbb{E}_\gamma \left[ \min_{(\mathbf{q},c) \in \mathcal{L}(F)} \max_{\mathbf{x} \in \{0,1\}^n} (\mathbf{\gamma} - \mathbf{q})^\top \mathbf{x} - c \right] \leq \min \mathbb{E}_\gamma \left[ \max_{\mathbf{x} \in \{0,1\}^n} (\mathbf{\gamma} - \mathbf{q})^\top \mathbf{x} - c \right] = \quad \text{(Theorem 4.3)}
\]

Now, note that each term \( \mathbb{E}_{\gamma_i} \max_{x_i \in \{0,1\}} (q_i + \gamma_i) x_i \) can be seen as the full Perturb-and-MAP approximation to the one variable model

\[
P(X_i = x_i) = \frac{1}{1 + e^{-q_i}} e^{-q_i x_i},
\]

so due to (3.30) it must evaluate to the log-partition function of the above distribution, namely \( \log(1 + e^{-q_i}) \). Hence, the last term is equal to

\[
\sum_{i=1}^{n} \log(1 + e^{-q_i}) + c = A(-\mathbf{q}) + c,
\]

and this completes the proof due to Theorem 4.3. \( \square \)
5

Inference in Log-Submodular and Log-Supermodular Models

Part of the material in this chapter has been already published in the following conference proceedings.

- Djolonga, J. & Krause, A. From MAP to Marginals: Variational Inference in Bayesian Submodular Models in NIPS (2014)

5.1 Introduction

Having introduced the necessary background on discrete optimization in Chapter 2, and on infinite divergences in the previous chapter, we now consider the first model families — log-submodular and log-supermodular distributions. They contain distributions over subsets $A \subseteq V$ of some finite ground set $V$, and are defined as

\begin{align}
\mathbb{P}(A = A) &= \frac{1}{Z} \exp(+F(A)), \text{ and} \\
\mathbb{P}(A = A) &= \frac{1}{Z} \exp(-F(A))
\end{align}

respectively. We would also like to point out that distributions over sets are also known as point processes. As in Section 2.1, we will also treat them as being defined over $n$ binary random variables $X_1, X_2, \ldots, X_n$, where $X_i$ indicates the presence of item $i$ in the random set $A$. We will make no further assumptions beyond the submodularity of $F$ throughout the chapter.

Log-supermodular and log-submodular distributions are also said to satisfy the positive and negative lattice condition respectively ([108]). Log-supermodular distributions have been also studied in the continuous case,
in which case they are known as multivariate totally positive distributions of order 2 \cite{109}. In this chapter, we will focus only on the binary case.

These families are closed under several important operations thanks to the closure properties of submodular functions. First, because submodularity is retained under addition, they are closed under multiplication. Second, they are closed under conditioning on the presence or absence of specific elements of $V$ in $A$, or equivalently, setting some of the variables $X_i$ to zero or one. Specifically, if we want to condition on the event that the sampled set $A$ is in the interval $[B,C] = \{A \mid B \subseteq A \subseteq C\}$ for some $B \subseteq C \subseteq V$, the distribution is given as

$$
P(A \mid A \in [B,C]) = \begin{cases} 0 & \text{if } A \notin [B,C], \\ \frac{\exp(\pm F(A))}{Z'} & \text{if } A \in [B,C], \end{cases}
$$

where $Z' = \sum_{A \in [B,C]} \exp(\pm F(A))$ ensures that it is normalized. Note that any set $A$ that has non-zero probability can be uniquely written as the union $A = B \cup A'$ for some $A' \in C - B$. Hence, we can also see the distribution \eqref{eq:5.3} as being defined over $A' \in C - B$. Formally, if we define a random variable $A'$ taking on values in $[C - B]$ defined by

$$
P(A' = A') = \exp(\pm F(A' \cup B)) / Z',
$$

it can be easily seen that $P(A' = A') = P(A = A' \cup B \mid A \in [B,C])$. To conclude, note that $A'$ is easily seen to be a member of the considered model class with energy function $F_{C_B}^C$, i.e., the function obtained from $F$ by contracting $B$ and restricting it to $C$, which we know to be submodular.

While both classes are defined through submodular functions, they have vastly different properties. Hence, we begin this chapter by presenting separately the properties of these classes, and showcasing several models that have been extensively studied in the literature.

Then, we will discuss how to perform approximate inference — specifically, we will tackle the problem of computing $Z$ and the unary marginals. Writing these models in an exponential family form might require combinatorially many parameters, so the application of inference methods tailored for exponential families can be very challenging. Hence, we approach this problem from the divergence minimization viewpoint. Concretely, we will focus on the optimization of the inclusive and exclusive infinite divergences over the family of fully factorized distributions. In other words, we consider the family $Q$ of distributions of the form

$$Q(A) = \exp(-q(A) - A(-q)),$$
where \( A(-\mathbf{q}) = \sum_{i=1}^{n} \log(1 + e^{-q_i}) \). We will then study the challenges that emerge from the resulting optimization problems, and suggest strategies to overcome them. We will also prove guarantees of the quality of the resulting approximations depending on the curvature of the energy function. Furthermore, we will show an even stronger approximation result for the family log-polymatroidal distributions.

Finally, we will numerically analyze the efficacy of the resulting algorithms, on both synthetic and real data.

**Prior Work**  
The MAP problems have received significant attention, as they reduce to the submodular minimization and maximization problems from Chapter 2. On the other hand, not all inference techniques from Chapter 3 are applicable, as we do not make any factorization assumption about the energy \( F \). For example, computing the exact updates for mean-field and belief-propagation can be in general prohibitively expensive. There have been also several other results that pertain to log-supermodular models. Weller & Jebara [110] has shown that both the belief-propagation and mean-field objectives can be optimized for pairwise models using a graph-cut if we discretize the marginals. Specifically, if we assume that each marginal takes on some set of \( k \) values, then we need to solve a min-cut in a graph with \( nk \) vertices. If the function is not pairwise, but we can evaluate its multi-linear extension, then we can apply the approach of Bach [111]. Moreover, Ruozzi [79] has shown that belief propagation is guaranteed to provide a lower bound for log-supermodular models. Gotovos, Hassani & Krause [112] and Rebeschini & Karbasi [113] study the convergence speed of Gibbs samplers for both model classes, and provide bounds on the mixing times that depend on uniform bounds of the marginal gains. Shpakova & Bach [95], whose result we have presented in Section 4.6, have originally shown the connection between Perturb-and-MAP and the inclusive divergence for log-supermodular models. Iyer & Bilmes [114] suggest to use distributions of the form \( P(A) \propto \pm F(A) \), which they call submodular point processes. These models are closed under mixtures and conditioning, and many classes can be exactly normalized as the partition function separates across the components of \( F \). They are however not closed under multiplication, positivity can be challenging to ensure for the submodular case, and a theory of approximate inference for them is yet to be developed.
5.2 THE MODEL CLASSES

5.2.1 Log-supermodular models

Before we discuss their properties, let us introduce some examples that often appear in the literature.

Example 5.1 (The ferromagnetic Ising model). One of the best understood log-supermodular models, due to its importance in statistical physics, is the Ising model from Example 3.1, defined as

$$P(x) = \exp(- \sum_{i=1}^{n} \gamma_{i}x_{i} - \sum_{i=1}^{n} \sum_{j=1}^{n-1} \beta_{i,j}x_{i}x_{j}). \quad (5.4)$$

Namely, note each term in the energy $\beta_{i,j}x_{i}x_{j}$ is submodular if $\beta \leq 0$, which is exactly the definition of the ferromagnetic case.

Example 5.2 (Regular MRFs). An extension of the Ising model are the binary Markov random fields with submodular potentials. Formally, the MRF has a submodular energy (also called regular) if its pairwise terms satisfy

$$\theta_{i,j,a,b} + \theta_{i,j;b,a} \geq \theta_{i,j,a,a} + \theta_{i,j;b,b}. \quad \text{It can be easily shown (see e.g. Kolmogorov & Rother [38]) that all models satisfying the above condition can be represented as the sum of a graph cut and a modular function. In other words, these models have the same representational power as the graph cut image segmentation model we have suggested in Example 2.15.}$$

Example 5.3 ($\mathcal{P}^{n}$ potentials). Another log-supermodular model that has been successfully used for image segmentation are the higher-order $\mathcal{P}^{n}$ potentials by Kohli, Ladicky & Torr [115]. They go beyond pairwise interactions and define potentials that are defined over larger spatial regions. Namely, they start by first generating several layers of superpixels of the image, as illustrated in Figure 5.2. Then, they encourage each such super-pixel to be consistently labelled using a carefully crafted energy function. For example, if we have obtained a set of $m$ superpixels $\{P_{j} \subseteq V\}_{j=1}^{m}$, then one could use the model

$$P(A) = \exp(- \sum_{j=1}^{m} \phi(|P_{j} \cap A|/|P_{j}|) - m(A)),$$

for some modular function $m$ holding the unary potentials, and a concave function $\phi: [0, 1] \rightarrow \mathbb{R}$. As we want to have consistently labelled superpixels, we want the
energy to be lowest when $A \cap P_j = P_j$ or $A \cap P_j = \emptyset$. Formally, we want to choose the concave function to achieve its minimum at 0 and 1, and its maximum at 1/2, as the one shown in Figure 5.1. One could for example use $\varphi(u) = u^\alpha (1-u)^\alpha$ for some $\alpha \in (0, 1]$. Because the energy function is a sum of concave-of-cardinality potentials (Example 2.7), the distribution is log-supermodular.

In the above examples, one can note that the variables tend to exhibit a cooperative, or attractive, behaviour. Namely, in regular MRFs the model prefers neighbouring pixels to have the same label whenever that would not cause the unary term to significantly increase. Similarly, for the higher-order potentials, if most of the pixels in the superpixel have the tendency towards one of the labels, the remaining ones should be encouraged to take on the majority label too. This intuition can be formalized using the classical inequality named after Fortuin, Kasteleyn & Ginibre [117].

**Theorem 5.1 (FKG Inequality [117, Proposition 1]).** Let $\mathcal{P}$ be a log-supermodular distribution, and let $G: 2^V \to \mathbb{R}$ and $H: 2^V \to \mathbb{R}$ be any monotone set functions. Then the following inequality holds

$$E_{\mathcal{P}}[G(A)H(A)] \geq E_{\mathcal{P}}[G(A)]E_{\mathcal{P}}[H(A)].$$

(5.5)

In other words, any two monotone functions can be only positive correlated under $\mathcal{P}$ — this property is also known as positive association ([108, 118]). As a simple instantiation, if we take $G(A) = \mathbb{1}[i \in A]$ and $H(A) = \mathbb{1}[j \in A]$ for some $i, j \in V$, both easily seen to be monotone, the above inequality translates to

$$E_{\mathcal{P}}[\mathbb{1}[i \in A]]E_{\mathcal{P}}[\mathbb{1}[j \in A]] \geq 0 \Leftrightarrow \text{Cov}(X_i, X_j) \geq 0.$$  

Hence, if we think of the distribution as defined over boolean random variables, they can exhibit only positive correlations. The following generalization of the above inequality, also known as the four functions theorem, implies the important property that log-supermodular distributions are also closed under marginalization.

**Theorem 5.2 (Ahlswede & Daykin [119, Theorem 1]).** Let $F_1, F_2, F_3, F_4: 2^V \to \mathbb{R}$ be set function that satisfy

$$F_1(A)F_2(B) \leq F_3(A \cup B)F_4(A \cap B)$$

(5.6)

for all $A, B \subseteq V$. Then the following inequality holds

$$\left[ \sum_{A \in \mathcal{A}} F_1(A) \right] \left[ \sum_{B \in \mathcal{B}} F_2(B) \right] \leq \left[ \sum_{C \in \mathcal{A} \cup \mathcal{B}} F_3(C) \right] \left[ \sum_{C \in \mathcal{A} \cap \mathcal{B}} F_4(C) \right]$$

(5.7)

for any $\mathcal{A}, \mathcal{B} \subseteq 2^V$. 
Specifically, assume that we want to marginalize out some \( C \subseteq V \), in which case we will obtain a distribution \( \mathbb{P}' \) over \( A \subseteq V - C \) defined as

\[
\mathbb{P}'(A) = \sum_{C' \subseteq C} \mathbb{P}(A \cup C').
\]

If we plug in \( F_1(A) = F_2(A) = F_3(A) = F_4(A) = \exp(-F(A)) \), condition (5.6) follows from the log-supermodularity of the distribution, and applying the theorem to \( A = \{ A \cup C' \mid C' \subseteq C \} \) and \( B = \{ B \cup C' \mid C' \subseteq C \} \), we obtain

\[
\mathbb{P}'(A)\mathbb{P}'(B) \leq \mathbb{P}'(A \cup B)\mathbb{P}'(A \cap B),
\]

which is exactly the log-supermodularity of the marginalized distribution.

To summarize, log-supermodular distributions are closed under conditioning, multiplication, and marginalization. Moreover, they model positive correlations, which makes them a natural choice for many problems where one wants to capture cooperative, or attractive, behaviour. Finally, due to the submodularity of its energy function, the problem of computing its MAP configuration can be solved in polynomial time using the methods for submodular minimization we have covered in Section 2.6.

### 5.2.2 Log-submodular models

We begin our presentation of log-submodular models by showcasing several instances of this model class.
Example 5.4 (The antiferromagnetic Ising model). The antiferromagnetic model, in contrast to its ferromagnetic counterpart Example 5.1 is log-submodular using an analogous argument.

Example 5.5 (FLID [120]). Another model is the facility Location Diversity (FLID) model of Tschiatschek, Djolonga & Krause [120]. It has been used for item recommendation, and its likelihood is given by

$$
\mathbb{P}(A) = \frac{1}{\mathcal{Z}} \exp \left( \sum_{i \in A} u_i + \sum_{j=1}^{d} \left[ \max_{i \in A} w_{ij} - \sum_{i \in A} w_{ij} \right] \right),
$$  \hspace{1cm} (5.8)

for some $w_{ij} \in [0, \infty)$ and $u_i \in \mathbb{R}$. We can easily see that it is log-submodular, because it can be written as a sum of facility location (Example 2.8) and a modular function. The motivation behind the model is the assumption that there exist $d$ latent properties that the items have, such that each item $i$ has an "expression level" $w_{ij}$ of property $j \in \{1, 2, \ldots, d\}$. Moreover, once we select an item that strongly exhibits property $j$, adding any additional items should result in a lower probability, unless that new item covers some of the other dimensions. In other words, this distribution should be able to capture the items that are substitutes of one another — purchasing a car should probably decrease the likelihood of buying other vehicles. Finally, each item $i \in V$ is assumed to have some intrinsic value which is captured using the modular term $u_i$.  

Figure 5.2: The original image (from [35]) and two layers of superpixels to be used for defining attractive higher order potentials. The super-pixels were created using the algorithm of Comaniciu & Meer [116].
Example 5.6 (Determinantal point process (DPP) [121, 122]). Let \( L \in \mathbb{R}^{n \times n} \) be a positive semi-definite matrix. Then, the determinant point process associated with \( L \) is given by
\[
P(A) = \frac{\det(L_A)}{\det(L + I)},
\]
where \( L_A \) is the submatrix formed from \( L \) by the rows and columns indexed by \( A \).

Continuous DPPs have a long history in physics, where they have been used to model processes of repulsive particles — for further references please see [122, §1]. According to Lyons [122], the discrete case that we study has first appeared as an exercise in the book of Daley & Vere-Jones [123]. They have been popularized in machine learning by Kulesza & Taskar [121]. Note that the distribution is not given up to an unknown normalization constant, as the partition function has a closed form equal to \( \det(L + I) \). If \( L_A \) has strictly positive eigenvalues, so then it never assigns zero probability, we can re-write the distribution as
\[
P(A) = \frac{\exp(\log \det L_A)}{\det(L + I)}.
\]
The distribution can be seen to be log-submodular due to the submodularity of the entropy (Example 2.9), as \( \log \det L_A \) is proportional to the entropy of a Gaussian random vector with a covariance matrix \( L_A \). Moreover, the distribution can be also easily marginalized (see e.g. [121, Theorem 2.1]), and we can obtain any marginal in cubic time. It is known (see e.g. [121, §2.1]) that they can model only repulsive behaviour, i.e., the variables can only be negative correlated under the model. Moreover, as proven by Lyons [122, Corollary 6.6], they satisfy the following, also known as negative association (NA [108, 124]) condition
\[
\mathbb{E}[F(A)G(A)] \leq \mathbb{E}[F(A)]\mathbb{E}[G(A)], \tag{5.9}
\]
where \( F, G : 2^V \rightarrow \mathbb{R} \) are monotone and depend on disjoint domains.

Analogously to the previous section, one might expect not only DPPs, but all of the above models always exhibit repulsive behaviour. However, this property is not always satisfied. Borcea, Brändén & Liggett [125] provide a concrete example of a log-submodular distribution that has positive pairwise correlations, which we present here in a modified form due to Alkis Gotovos.

Example 5.7 (Borcea, Brändén & Liggett [125, Example 2.1]). Consider the undirected graph cut function on a chain graph with 3 variables, i.e., the distribution has the form
\[
P(x) = \frac{1}{Z} \exp(a[x_1 \neq x_2] + a[x_2 \neq x_3]).
\]
We show in Table 5.1 the energy of all states, as well as the partition function. Due

<table>
<thead>
<tr>
<th>State x</th>
<th>F(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, 0, 0</td>
<td>exp(0)</td>
</tr>
<tr>
<td>0, 0, 1</td>
<td>exp((\alpha))</td>
</tr>
<tr>
<td>0, 1, 0</td>
<td>exp(2(\alpha))</td>
</tr>
<tr>
<td>0, 1, 1</td>
<td>exp((\alpha))</td>
</tr>
<tr>
<td>1, 0, 0</td>
<td>exp((\alpha))</td>
</tr>
<tr>
<td>1, 0, 1</td>
<td>exp(2(\alpha))</td>
</tr>
<tr>
<td>1, 1, 0</td>
<td>exp((\alpha))</td>
</tr>
<tr>
<td>1, 1, 1</td>
<td>exp(0)</td>
</tr>
</tbody>
</table>

Table 5.1: The example function resulting in a log-submodular model with a positive correlation. Note that \(Z = 2 + 4 \exp(\alpha) + 2 \exp(2\alpha)\).

to the symmetry it can be easily seen that
\[
P(X_1 = 1) = P(X_3 = 1) = 1/2.
\]

However, we have that the joint probability is equal to
\[
P(X_1 = 1, X_3 = 1) = \frac{1 + \exp(2\alpha)}{2 + 4 \exp(\alpha) + 2 \exp(2\alpha)} = \frac{1}{2} \times \frac{1}{\text{sech}(\alpha) + 1},
\]
which is always greater than \(P(X_1 = 1)P(X_3 = 1) = 1/4\) because the image of \(\text{sech}\) is the interval \((0, 1)\). Hence variables \(X_1\) and \(X_3\) are positively correlated with one another for any \(\alpha\).

The main problem with log-submodularity is that these distributions, unlike their log-supermodular counterparts, are not closed under marginalization. Pemantle [108] has motivated the development of a theory of negative correlation, and has suggested several definitions that should capture this behaviour. One such condition, known as CNA+ ([108]) is the requirement for (5.9) to hold if we add a modular term to the energy, or marginalize some of the variables. This is indeed satisfied by DPPs ([122, Corollary 6.6]), and a subset of submodular models known as strongly Rayleigh distributions [125]. Unfortunately, this property does not seem easily verifiable as is depends on the zeros of the generating polynomial of \(P\).

5.3 Inference in Log-Submodular Models

Because the optimization properties of submodular minimization and maximization significantly differ, we will treat the two model classes separately.
Let us first focus on the log-submodular case, i.e., those whose energy function is equal to \(-F\) for some submodular function \(F: 2^V \rightarrow \mathbb{R}\). Note that there is a minus in front of \(F\) in the energy function — we have decided for this convention so that \(F\) always refers to a submodular function throughout this chapter, which would also make straightforward the references to the results from Chapter 2.

Minimizing \(D_\infty(Q \parallel P)\) Let us start with the analysis of the problem resulting from the minimization of the exclusive divergence. Because it can be easily seen that \(\mathcal{U}(-F) = -\mathcal{L}(F)\), the lower bound resulting from this divergence can be written using the polyhedral primal (Theorem 4.2) as

\[
\sup_{(q,c) \in \mathcal{L}(F)} \sum_{i=1}^{n} \log(1 + e^{q_i}) + c \leq \log Z. \tag{5.10}
\]

Remember that this is a convex maximization problem. As convex functions always achieve their maximum at an extremal point, we will obtain the same optimum if we only optimize over the vertices of the domain.

**Theorem 5.3.** The supremum in equation (5.10) is attained for some \((q^*, 0) \in \mathcal{L}(F)\), where \(q^*\) is an extreme point of \(B(F)\).

**Proof.** First, we will prove that the supremum is attained at some point \((q^*, c^*)\) with \(c^* = 0\), i.e., \(q^* \in \mathcal{P}(F)\). Namely, because the objective is concave and the optimization domain is polyhedral, we can directly apply [126, Corollary 32.3.4] as \(\mathcal{L}(F)\) contains no lines (follows because it does have extreme points [19, Lemma 2.6]).

Hence, we know that there is an optimal \((q^*, 0)\) such that \(q^*\) is an extreme point of \(\mathcal{P}(F)\). Finally, from Theorem 2.8 we know that the extreme points of \(\mathcal{P}(F)\) and \(B(F)\) are the same, and this completes the proof. \(\Box\)

Because the base polytope has finitely many vertices, we have reduced the continuous optimization problem to discrete optimization. However, even though there are only finitely many extreme points of \(B(F)\), they can be up to \(n!\) of them as we obtain one for every permutation (Theorem 2.8) using Edmonds' greedy algorithm (Algorithm 1). We have thus reduced the optimization problem to

\[
\max_{\pi \in \text{Sym}(V)} \sum_{i=1}^{n} \log(1 + \exp(F(\pi(i) \mid \{\pi(1), \pi(2), \ldots, \pi(n-1)\}))),
\]
where \( \text{Sym}(V) \) is the group of all permutations on \( V \). The function \( u \to \log(1 + e^u) \), also called the softplus function, can be seen as a smooth approximation to \( \max(0, u) \). Hence, among all permutations, we have to find one that will result in the largest sum of marginal gains, where negative gains are being smoothly rounded to zero.

Perhaps the most straightforward algorithm would be the greedy strategy — iteratively construct the permutation by selecting as the next element the one that has the largest marginal gain. In other words, we run the greedy algorithm (Algorithm 4) of Nemhauser, Wolsey & Fisher [15] for \( k = n \) and without stopping when negative marginal gains are encountered. Then, if \( \pi \) is the permutation obtained using this strategy, we suggest as an approximate distribution \( Q \) the one that is fully factorized with marginal probabilities

\[
Q(X_{\pi(i)} = 1) = \sigma(F(\pi(i) | \{\pi(1), \pi(2), \ldots, \pi(n-1)\}))
\]

where \( \sigma \) is the sigmoid function. Thanks to the results of the greedy maximization algorithms for submodular functions (Theorem 2.19 and Lemma 2.6), the fact that submodular functions have diminishing gains, and because the sigmoid is greater than 1/2 iff its argument is non-negative, we have the following two results.

**Lemma 5.1.** Let \( Q^* \) be the distribution computed using Algorithm 5. If \( F \) is polymatroid, the set \( A_k \) containing the \( k \) elements with highest marginals \( k \) marginals is a \((1 - 1/e)\)-factor approximation to \( \max_{|A|\leq k} F(A) \).

**Lemma 5.2.** Let \( Q^* \) be the distribution computed using Algorithm 5. If \( F \) is \( M^\natural \)-concave, the MAP of \( Q^* \), computed by thresholding the marginals

\[
A^* = \{i \in V | Q(X_i = 1) \geq 1/2\}
\]

is a maximizer of \( F \), and hence also a MAP of \( P \).

**Minimizing \( D_\infty(P \parallel Q) \)** Let us now consider the problem of optimizing the inclusive infinite divergence. Because \( L(-F) = -U(F) \), using Theorem 4.3 we can write the upper bound as

\[
\inf_{(q,c) \in U(F)} \sum_{i=1}^{n} \log(1 + e^{qi}) + c \geq \log Z. \tag{5.11}
\]

While this is a convex optimization problem, it is unfortunately hard to solve due to the presence of the upper polyhedron, which does not have a
Algorithm 5 Greedy maximization of the lower bound.

1: procedure GreedyLowerBound($F$)
2: Define $A = \emptyset$
3: $q \leftarrow 0 \in \mathbb{R}^n$
4: for $i \leftarrow 1, 2, \ldots, n$ do
5:     Pick $j^* \in \arg\max_{j \in V - A} F(j \mid A)$
6:     $q_{j^*} = F(j \mid A)$
7:     $A \leftarrow A \cup \{j^*\}$
8: end for
9: return $Q$ with marginals $Q(X_i = 1) = \sigma(q_i)$.
end procedure

Concise description. As a first approach, we suggest to minimize the above problem only over supergradients. Recall the definition (Definition 2.20) of the superdifferential at any $A \subseteq V$

$$\partial^F(A) = \{ q \in \mathbb{R}^n \mid \forall B \subseteq V : F(B) \leq q(B) + F(A) - q(A) \}.$$ 

Note that if $q \in \partial^F(A)$, then $(q, F(A) - q(A)) \in U(F)$. Hence, if we minimize (5.11) only over supergradients, we have the bound

$$\inf_{(A \subseteq V, q \in \partial^A(F))} \sum_{i=1}^n \log(1 + e^{q_i}) - q(A) + F(A). \quad (5.12)$$

Unfortunately, the above problem is a mixed discrete-continuous problem that would be very hard to optimize directly. We propose to minimize it only over the bar supergradients from Table 2.1, i.e., restrict the domain of problem (5.12) to pairs of the form $(A, \overline{q}^A)$. In this way we have effectively reduced it to the discrete optimization problem of selecting the optimal $A$. Fortunately, as shown by the following lemma, we can solve the problem exactly using a single submodular minimization problem.

**Lemma 5.3.** Optimizing problem (5.12) over all pairs of the form $(A, \overline{q}^A)$ is equivalent to the following submodular minimization problem

$$\min_{A \subseteq V} F(A) + m_1(A),$$

where the modular function $m_1$ has entries

$$m_1(\{i\}) = \log(1 + e^{-F(i|V-i)}) - \log(1 + e^{+F(i)}).$$

**Proof.** Note that the bar supergradient is defined as

$$\overline{q}^A = a \odot 1_A + b(1 - 1_A),$$
for the vectors $a$ and $b$ in $\mathbb{R}^n$ with entries $a_i = F(i \mid V - \{i\})$ and $b_i = F(\{i\})$ respectively, so that the optimization problem is equal to

$$\min_{A \subseteq V} \sum_{i \in A} \log(1 + e^{a_i}) + \sum_{i \notin A} (1 + e^{b_i}) - \sum_{i \in A} a_i + F(A).$$

Now, because $\log(1 + e^u) - u = \log(1 + e^u) - \log e^u = \log(1 + e^{-u})$, the problem reduces to

$$\min_{A} \sum_{i \in A} \log(1 + e^{-a_i}) + \sum_{i \notin A} \log(1 + e^{b_i}) + F(A),$$

which we can easily transform the claimed problem by subtracting the constant $\sum_{i \in V} \log(1 + e^{b_i})$.

Even though we cannot optimize over $grow$ and $shrink$ supergradients, we can evaluate all three at the optimum for the problems above and pick the one that gives the best bound.

As a second approach we suggest to attack problem (5.11) directly, by replacing the upper polyhedron with the inner bar bound $\overline{U}(F) \subseteq U(F)$ from Definition 2.21. As we would be optimizing only over a smaller subset, the bound would not be violated, and the problem becomes

$$\min_{(q, c) \in \overline{U}(F)} \sum_{i=1}^{n} \log(1 + e^{q_i}) + c. \quad (5.13)$$

Note that the gradient of the objective is always of the form $\mathbf{g}, 1 \in \mathbb{R}^{n+1}$. We can leverage the fact that the last component is non-negative to our benefit. Namely, that would let us compute the linear oracle over $\overline{U}(F)$, which we now formalize in the following lemma.

**Lemma 5.4.** A minimizer of $(\mathbf{g}, 1)^{\top}$ over $\overline{U}(F)$ is given by $(\mathbf{q}^*_A, F(A^*) - \mathbf{q}^*_A(A^*))$, where $A^*$ is the optimizer of the following submodular minimization problem

$$\min_{A} F(A) + m_2(A),$$

where the modular function $m_2$ has entries

$$m_2(\{i\}) = g_i(F(i \mid V - \{i\}) - F(\{i\})) - F(i \mid V - \{i\}).$$

**Proof.** The optimization problem is equal to

$$\min_{A \subseteq V} \mathbf{g}^{\top} \mathbf{q}^A + 1 \times [F(A) - \mathbf{q}^A(A)].$$
As in the proof of Lemma 5.3, we use the fact that \( q^A = a \odot 1_A + b(1 - 1_A) \), so that the optimization problem becomes (ignoring constants terms)

\[
\min_{A \subseteq V} \sum_{i \in A} g_i a_i + \sum_{i \notin A} g_i b_i + [F(A) - \sum_{i \in A} a_i],
\]

from which we subtract \( \sum_{i \in V} g_i b_i \) to obtain

\[
\min_{A \subseteq V} \sum_{i \in A} (g_i (a_i - b_i) - a_i) + F(A),
\]

and this is exactly what we had to show after expanding \( a_i \) and \( b_i \). \( \square \)

Hence, we can apply Frank-Wolfe to problem (5.13) as we have a linear oracle, a convex smooth objective, and a compact domain.

**THE \( M^\natural \)-CONCAVE CASE**  If the function happens to, in addition of submodularity, also satisfy \( M^\natural \)-concavity, we can optimally solve the minimization of the inclusive divergence. Namely, remember that (see e.g. Theorem 4.3) we can also state the optimization problem as

\[
\sum_{i=1}^n \log(1 + e^{-q_i}) + \max_{A \subseteq V} q(A) + F(A),
\]

which is a convex optimization problem. The biggest difficulty lies in the inner discrete maximization problem, which we can solve for \( M^\natural \)-concave functions in linear time using the greedy method according to Lemma 2.6. Hence, we can cheaply compute subgradients and use any first order optimization method to achieve any arbitrary precision in polynomial time.

Alternatively, if we have a good candidate superdifferential \( \partial^A(F) \), we can use the fact that we know (Theorem 2.20) how to exactly characterize the superdifferential of \( M^\natural \)-concave functions. Specifically, if we fix the superdifferential \( \partial^A(F) \), the optimization problem (5.12) becomes

\[
\min_{q \in \partial^F(A)} \sum_{i \in A} \log(1 + e^{-q_i}) + \sum_{j \notin A} \log(1 + e^{q_j}) + F(A), \tag{5.14}
\]

with the constraint set given as

\[
\partial^F(A) = \{ q \in \mathbb{R}^n \mid (\forall i \in A) : q_i \leq F(i \mid A - \{i\}), \\
(\forall j \notin A) : q_j \geq F(j \mid A), \\
(\forall i \in A, j \notin A) : q_i - q_j \leq F(A) - F(A \cup \{j\} - \{i\}) \}.
\]
As the objective is strictly convex in $q$ due to the presence of the softplus, we propose to minimize the objective in a block coordinate descent manner, where the blocks are $\{q_i \mid i \in A\}$ and $\{q_j \mid j \notin A\}$. We prove the updates below, and give the complete algorithm in Algorithm 6.

**Lemma 5.5.** The updates in Algorithm 6 are minimize the objective value in a coordinate-wise manner.

**Proof.** If we fix all $q_i$ for $i \in A$, then $q_j$ for $j \notin A$ have to satisfy

$$q_j \geq F(j \mid A), \text{ and}$$

$$q_j \geq q_i - F(A) + F(A \cup \{j\} - \{i\}) \text{ for all } i \in A.$$ 

As the objective monotonically increases with $q_j$, we want to keep it as small as possible, which means that we have to set it as the maximum of the right hand sides of these $1+|A|$ inequalities.

Similarly, if we fix $q_j$ for $j \notin A$, then $q_i$ for $i \in A$ has to satisfy

$$q_i \leq F(i \mid A - \{i\}), \text{ and}$$

$$q_i \leq q_j + F(A) - F(A \cup \{j\} - \{i\}) \text{ for all } j \notin A.$$ 

As the objective is monotonously decreasing in $q_i$, we want to keep it as small large as possible, which means setting it to the minimum of the right hand sides of the above inequalities.

**Algorithm 6** Alternate optimization for (5.14) over $q \in \partial^A(F)$

```latex
\textbf{Require:} M\textsuperscript{2} function $F$, tight set $A$ s.t. $m(A) = F(A)$ for the returned $m$

$q \leftarrow 0$

for $t = 1, 2, \ldots, T$

\begin{align*}
\forall i \in A: & \quad q_i \leftarrow \min\{F(i \mid A - \{i\}), \min_{j \in A - A} q_j + F(A) - F(A \cup \{j\} - \{i\})\}
\end{align*}

\begin{align*}
\forall j \notin A: & \quad q_j \leftarrow \max\{F(j \mid A), \max_{i \in A} q_i + F(A \cup \{j\} - \{i\}) - F(A)\}
\end{align*}

end for

return $q$
```

5.4 INFECTION IN LOG-SUPERMODULAR MODELS

Let us now turn our attention to log-supermodular models. The first problem we consider is that of minimizing the exclusive divergence, which will lend itself to approximate approaches similarly to those we have just seen for the minimization of the inclusive divergence of log-submodular models.
minimizing $D_\infty(Q \| P)$ Using Theorem 4.2 we can write the lower bound as
\[
\sup_{(q,c) \in U(F)} \sum_{i=1}^{n} \log(1 + e^{-q_i}) - c \leq \log Z.
\]
Unfortunately, we are confronted with two problems — not only that the problem is that of convex maximization, but we can not describe the constraint set. Nevertheless, because we know that there will be an optimum at a face of the domain, it makes sense to optimize over the superdifferentials, as they make up the faces of $U(F)$ (Lemma 2.2). Analogously to what we have seen in the previous section, if we plug in the bar supergradients $\overline{q}^A$ we have the following lower bound that we can optimize
\[
\max_{A \subseteq V} \sum_{i=1}^{n} \log(1 + e^{-\overline{q}_i}) - (F(A) - \overline{q}^A(A)), \tag{5.15}
\]
which, again turns out again to be an instance of submodular minimization.

**Lemma 5.6.** We can obtain an optimal $A^*$ to problem (5.15) by solving the following problem
\[
\min_{A \subseteq V} F(A) + m_3(A),
\]
where the modular function $m_3$ has entries
\[
m_3\left(\{i\}\right) = \log(1 + e^{-F(i)}) - \log(1 + e^{F(i)\mid V\setminus\{i\}}).
\]
which is an instance of submodular minimization.

**Proof.** The argument is almost exactly the same as that in the proof of Lemma 5.3. Namely, given that $\overline{q}^A = 1_A \odot a + (1 - 1_A) \odot b$, the optimization problem becomes
\[
\max_{A \subseteq V} \sum_{i \in A} \log(1 + e^{-a_i}) + \sum_{i \notin A} \log(1 + e^{-b_i}) - F(A) + \sum_{i \in A} a_i.
\]
Now, using the fact that
\[
\log(1 + e^{-u}) + u = \log(1 + e^{-u}) + \log e^u = \log(1 + e^u),
\]
if we instead minimize the negation of the function, the problem becomes
\[
\min_{A \subseteq V} - \sum_{i \in A} \log(1 + e^{a_i}) - \sum_{i \notin A} \log(1 + e^{-b_i}) + F(A),
\]
and the result follows by adding the constant $\sum_{i \in V} \log(1 + e^{-b_i})$. \qed
We now arrive at the last case, the inclusive divergence for log-supermodular distributions. Because we know how to compute the convex extension of submodular functions in polynomial time (Theorem 2.12), we can solve this problem using the entropic form of the bound (Theorem 4.5), equal to

$$\log Z \leq \sup_{\mu \in [0,1]^n} \mathcal{H}(\mu) - f_L(\mu),$$

where $f_L$ is the Lovász extension of $F$, and remember that

$$\mathcal{H}(\mu) = \sum_{i=1}^n -\mu_i \log \mu_i - (1 - \mu_i) \log(1 - \mu_i).$$

Nevertheless, when developing algorithms it will be more useful to approach the above problem from its dual. Instead of deriving the dual using Fenchel's theorem, we will instead start from the primal form (Theorem 4.3) of the upper bound, namely

$$\log Z \leq \inf_{q \in \mathbb{R}^n} A(-q) + f^*_\vee(q).$$

We take this route, as in the upcoming chapters we will run into similar problems, which can be all significantly simplified using the following lemma.

**Lemma 5.7.** Assume that we want to minimize $f^*_\vee(x) + g(-x)$ for some submodular function $F : 2^V \to \mathbb{R}$ and proper convex $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$. Then, if $g^*$ is $\infty$ outside $[0,1]^n$, we have the following relationship

$$\inf_{x \in \mathbb{R}^n} f^*_\vee(x) + g(-x) = \inf_{x \in \mathcal{B}(F)} g(-x).$$

*In other words, there is always an optimal point $x^*$ satisfying $f^*_\vee(x^*) = 0$.***

**Proof.** From Theorem A.1, the dual problem will have no gap as $f^*_\vee$ is continuous everywhere, and has the form

$$\sup_{x \in \mathbb{R}^n} -f_\vee(x) - g^*(x),$$

where $f_\vee(x) = (f^*_\vee)^*(x)$ is the convex envelope of $F$, equal to the Lovász extension on $[0,1]^n$ and $+\infty$ outside. Hence, using the assumption about
g^*, the dual problem is the same if we replace f with the Lovász extension as the \([0,1]^n\) constraint will be implicitly enforced by \(g^*\), i.e.,
\[
\sup_{x \in \mathbb{R}^n} -f_L(x) - g^*(x),
\]
The Fenchel dual of the above, namely
\[
\inf_{x \in \mathbb{R}^n} f_L^*(x) + g(-x)
\]
will again have no gap as \(f_L\) is defined everywhere, and this completes the proof as the conjugate of the Lovász extension \(f_L^*\) is the indicator function on the base polytope.

**Corollary 2.** For log-supermodular models minimizing \(D_\infty(P \parallel Q)\) over all factorized distributions is equivalent to
\[
\min_{q \in B(F)} \sum_{i=1}^n \log(1 + e^{-q_i}), \quad (5.18)
\]
which is polynomial-time tractable via \(O(|V|)\) submodular minimization problems using the divide-and-conquer algorithm (Algorithm 3).

As a side remark, we would like to point out that the above reduction does not necessarily hold for non log-supermodular distributions.

**Lemma 5.8.** There is a log-submodular distribution such that the resulting problem (5.16) has no optimum \(x^*\) with \(f^*_\vee(x^*) = 0\).

**Proof.** From Theorem 4.3 we know that (5.16) is equivalent to
\[
\inf_{(q,c) \in \mathcal{L}(F)} \sum_{i=1}^n \log(1 + e^{-q_i}) - c.
\]
Fixing \(c = 0\) reduces the problem to
\[
\inf_{q \in \mathbb{R}^n} \sum_{i=1}^n \log(1 + e^{-q_i}) \text{ s.t. } \forall A \subseteq V: q(A) \leq F(A). \quad (5.19)
\]
If we introduce for each \(A \subseteq V\) a dual variable \(\lambda_A \geq 0\) to enforce the constraint \(q(A) + c \leq F(A)\), we obtain the following Lagrange dual of problem (5.19) is
\[
\begin{align*}
\max & - \sum_{A \subseteq V} F(A)\lambda_A + \sum_{i \in V} h(\sum_{A \ni i} \lambda_A) \\
\text{subject to } & 0 \leq \sum_{A \ni i} \lambda_A \leq 1 \quad \text{for all } i \in V,
\end{align*}
\]

\[ (5.20) \]
where \( h(p) = -p \log p - (1 - p) \log (1 - p) \) is the binary entropy function (defined so that \( h(0) = h(1) = 0 \)). The Lagrange dual is easily derived if we note that \(-h(-u)\) is the convex conjugate of the primal objective and use [127, § 5.1.6]. Consider the function

\[
F(\emptyset) = 0, F(\{1\}) = -20, F(\{2\}) = -8, F(\{1, 2\}) = -16,
\]

which is supermodular as

\[
F(1 \mid \{2\}) = -16 - (-8) = -8 > -20 = F(\{1\}), \text{ and }
\]

\[
F(2 \mid \{1\}) = -16 - (-20) = 4 > 8 = F(\{2\}).
\]

If we take the dual variable \( \lambda_\emptyset = \lambda_V = 0, \lambda_1 = 1 = \lambda_2 = 1 \), which is easily seen to be feasible, the dual objective is \(-F(1)\lambda_1 - F(0)\lambda_0 + 0 = 28\) and thus the best bound achievable with \( c = 0 \) is at most 27.1.

For \( c = -1 \) consider the point \( q = (-19, -7) \), which is feasible for the primal as

\[
x_1 + c = -19 - 1 = -20 \leq F(\{1\}) = 20,
\]

\[
x_2 + c = -7 - 1 = -8 \leq F(\{1, 2\}) = 8, \text{ and }
\]

\[
x_1 + x_2 + c = -19 - 7 - 1 = -27 \leq F(\{1, 2\}) = -16.
\]

The primal objective value for \( q \) is \(< 27.1\), which is strictly better than the best achievable value for \( c = 0 \), and this completes the proof.

Let us analyze strategies for solving problem (5.18). We have already noted in Corollary 2 that we can apply the divide-and-conquer algorithm, which would yield an exact solution, if we can efficiently minimize \( F \) when perturbed with a modular function. As an alternative, we can solve the problem iteratively using the Frank-Wolfe algorithm (Algorithm 2), as explained in Section 2.6.1.1.

Perhaps surprisingly, it turns out that the approximate distribution has a close connection to the min-norm problem that we have seen and analyzed for the problem of submodular minimization in Section 2.6.1. Specifically, the parameter \( q^* \) minimizing the inclusive divergence (5.18), is exactly the min-norm point of the submodular function \( F \).

**Theorem 5.4.** Problems (5.18) and (2.26) have the same solution.

We will show a stronger result from which Theorem 5.4 follows by taking \( w_i = 1 \) and \( y_i = 0 \).
Lemma 5.9. For positive weights \( w_i > 0 \) the objectives \( \sum_{i \in V} w_i(x_i - y_i)^2 \) and \( \sum_{i \in V} \frac{1}{w_i} \log(\exp(-w_i x_i) + \exp(-w_i y_i)) \) have the same optimum under the constraint \( x \in B(F) \).

Proof. For the second objective we have that
\[
\sum_{i \in V} \frac{1}{w_i} \log(\exp(-w_i x_i) + \exp(-w_i y_i)) = (\text{Using } z \equiv -x)
\]
\[
\sum_{i \in V} \frac{1}{w_i} \log(\exp(w_i z_i) + \exp(-w_i y_i)).
\]
Hence, the optimum of the problem is the negative of the problem on \(-B(F)\), which is the base polytope of the reflection \( F(A) = F(V - A) - F(V) \). The gradient of this objective with respect to any \( z_i \) is equal to \( \sigma(w_i(z_i + y_i)) \), where \( \sigma(u) = 1/(1 + e^{-u}) \) is the sigmoid function. As the weights \( w_i \) are positive, we have that
\[
\sigma(w_i(z_i + y_i)) \leq \sigma(w_j(z_j + y_j)) \iff w_i(z_i + y_i) \leq w_j(z_j + y_j).
\]
From [22, Theorem 8.1], it follows that the above problem and \( \sum_i w_i(z_i + y_i)^2 \) have the same solution \( z^* \) on \(-B(F)\). However, if \( z^* \) is the projection of \(-y\) onto \(-B(F)\), then \( x = -z^* \) is the projection of \( y \) onto \( B(F) \).

We would like to point out that similar results have been shown (for other objectives) by Nagano & Aihara [128]. The first consequence of the above theorem is that we can construct the approximate distribution using any of the specialized algorithms we have discussed in Section 2.6.1, so that we do not have to resort to the divide-and-conquer and Frank-Wolfe algorithms. Moreover, thanks to the connection between the min-norm point (Theorem 2.16) and the minimizer of \( F \), we can easily extract the MAP from the optimal distribution \( Q \).

Corollary 3. We can extract the unique minimal and maximal MAP solutions by thresholding the optimal marginal vector at \( 1/2 \).

Hence, we can characterize the approximate distribution as not only being inclusive, but as also sharing the MAP configuration with \( P \).

5.5 APPROXIMATION GUARANTEES

Because the ferromagnetic and antiferromagnetic models are log-supermodular (Example 5.1) and log-submodular respectively (Example 5.4), we can not
hope to find any strong approximation guarantees for arbitrary submodular functions due to the hardness results from Section 3.1. Moreover, remember that we are optimizing the infinite divergences — it is not necessarily true that they will result in the tightest bounds nor most precise unary marginals, because that is not their primary objective. Nevertheless, it turns out that there are several model classes for which one can provide meaningful approximation guarantees regarding the estimate of the partition function.

### 5.5.1 Curvature-based bounds

Because we are essentially approximating the energy function $F$ with a modular function, we can obtain bounds that decrease as $F$ gets closer to being linear. To prove the result, we start with the following straightforward lemmas, which state that if we can linearize the non-modular part of $F$ within a multiplicative constant, this can be translated this into a multiplicative approximation bound on $\log Z$.

**Lemma 5.10.** Assume that $F(A) = G(A) + m(A)$ for some $G: 2^V \to \mathbb{R}$ and modular function $m$. If some $q' \in \mathbb{R}^n$ and $\eta \in [0, 1]$ it holds that

$$\forall A \subseteq V: (1 - \eta)G(A) \leq q'(A),$$

then for $q = q' + m$ we have

$$\sum_{i=1}^{n} \log(1 + e^{-q_i}) - \log \sum_{A \subseteq V} \exp(-F(A)) \leq \eta \max_{A \subseteq V} G(A). \quad (5.21)$$

Similarly, if for some $q' \in \mathbb{R}^n$ and $\gamma \in [0, 1]$ it holds that

$$\forall A \subseteq V: (1 + \gamma)G(A) \geq q'(A),$$

then for $q = q' + m$ we have

$$\log \sum_{A \subseteq V} \exp(-F(A)) - \sum_{i=1}^{n} \log(1 + e^{-q_i}) \leq \gamma \max_{A \subseteq V} G(A). \quad (5.22)$$
Proof. Let us set $G_{\text{max}} = \max_{A \subseteq V} G(A)$. Then, the first case can be proven as follows.

$$\sum_{i=1}^{n} \log(1 + e^{-q_i}) - \log \sum_{A \subseteq V} e^{-F(A)} = \log \frac{\sum_{A \subseteq V} e^{-F(A)}}{\sum_{A \subseteq V} e^{-F(A)}}$$

$$\leq \log \frac{\sum_{A \subseteq V} e^{-(m(A)+q'(A))}}{\sum_{A \subseteq V} e^{-F(A)}}$$

$$= \log \frac{\sum_{A \subseteq V} e^{-F(A)+\eta G(A)}}{\sum_{A \subseteq V} e^{-F(A)}}$$

$$\leq \log \frac{\sum_{A \subseteq V} e^{-F(A)+\eta G_{\text{max}}}}{\sum_{A \subseteq V} e^{-F(A)}} \quad \eta \geq 0.$$ 

The other case can be also easily shown.

$$\log \sum_{A \subseteq V} e^{-F(A)} - \sum_{i=1}^{n} \log(1 + e^{-q_i}) = \log \frac{\sum_{A \subseteq V} e^{-F(A)}}{\sum_{A \subseteq V} e^{-(m(A)+q'(A))}}$$

$$\leq \log \frac{\sum_{A \subseteq V} e^{-F(A)}}{\sum_{A \subseteq V} e^{-(m(A)+(1+\eta)G(A))}}$$

$$= \log \frac{\sum_{A \subseteq V} e^{-F(A)-\eta G(A)}}{\sum_{A \subseteq V} e^{-F(A)-\eta G_{\text{max}}}} \quad \eta \geq 0.$$ 

$$= \eta G_{\text{max}}$$

Lemma 5.11. Assume that $F(A) = G(A) + m(A)$ for some $G : 2^V \to \mathbb{R}$ and modular function $m$. If some $q' \in \mathbb{R}^n$ and $\eta \in [0,1]$ it holds that

$$\forall A \subseteq V : (1 - \eta)G(A) \leq q'(A),$$

then for $q = q' + m$ we have

$$\log \sum_{A \subseteq V} \exp(F(A)) - \sum_{i=1}^{n} \log(1 + e^{q_i}) \leq \eta \max_{A \subseteq V} G(A). \quad (5.23)$$
Similarly, if for some \( \mathbf{q} \in \mathbb{R}^n \) and \( \gamma \in [0, 1] \) it holds that
\[
\forall A \subseteq V: (1 + \eta)G(A) \geq q'(A),
\]
then for \( \mathbf{q} = \mathbf{q}' + \mathbf{m} \) we have
\[
\sum_{i=1}^{n} \log(1 + e^{q_i}) - \log \sum_{A \subseteq V} e^{-F(A)} \leq \gamma \max_{A \subseteq V} G(A). \tag{5.24}
\]

Proof. The proof is analogous to that of the previous theorem, and we provide it only for completeness. We will again set \( G_{\text{max}} = \max_{A \subseteq V} G(A) \).

\[
\sum_{i=1}^{n} \log(1 + e^{-q_i}) - \log \sum_{A \subseteq V} e^{-F(A)} = \log \frac{\sum_{A \subseteq V} e^{-(m(A)+q'(A))}}{\sum_{A \subseteq V} e^{-F(A)}} \\
\leq \log \frac{\sum_{A \subseteq V} e^{-(m(A)+(1-\eta)G(A))}}{\sum_{A \subseteq V} e^{-F(A)}} \\
= \log \frac{\sum_{A \subseteq V} e^{-F(A)+\eta G(A)}}{\sum_{A \subseteq V} e^{-F(A)}} \\
\leq \log \frac{\sum_{A \subseteq V} e^{-F(A)+\eta G_{\text{max}}}}{\sum_{A \subseteq V} e^{-F(A)}} \quad \eta \geq 0. \\
= \eta G_{\text{max}}
\]

\[
\log \sum_{A \subseteq V} e^{-F(A)} - \sum_{i=1}^{n} \log(1 + e^{-q_i}) = \log \frac{\sum_{A \subseteq V} e^{-F(A)}}{\sum_{A \subseteq V} e^{-(m(A)+q'(A))}} \\
\leq \log \frac{\sum_{A \subseteq V} e^{-F(A)}}{\sum_{A \subseteq V} e^{-(m(A)+(1+\eta)G(A))}} \\
= \log \frac{\sum_{A \subseteq V} e^{-F(A)-\eta G(A)}}{\sum_{A \subseteq V} e^{-F(A)-\eta G_{\text{max}}}} \quad \eta \geq 0. \\
= \eta G_{\text{max}}
\]

The notion of curvature is defined for polymatroid functions.
**Definition 5.1** (Conforti & Cornuejols [129, §1]). Let $G : 2^V \to \mathbb{R}$ be a polymatroid function. The curvature $\kappa$ of $G$ is defined as

$$\kappa = 1 - \min_{i \in V : G(\{i\}) > 0} \frac{G(i|V - \{i\})}{G(\{i\})}. \quad (5.25)$$

The curvature is always between 0 and 1 and is equal to 0 if and only if the function is modular. Although the curvature is property of polymatroid functions, we can still show results for the general case as any submodular function $F$ can be decomposed [130] as the sum of a modular term $m(\cdot)$ defined as $m(\{i\}) = F(i|V - \{i\})$ and $G = F - m$, which is a polymatroid function, because for each $A \subseteq V - \{i\}$ and $i \in A$

$$G(i \mid A) = F(i \mid A) - F(i \mid V - \{i\}) \geq 0,$$

where the inequality follows from the diminishing gains property and the fact that $A \subseteq V - \{i\}$. Moreover, as pointed out by Iyer, Jegelka & Bilmes [131], we can also remove any $i$ such that $G(\{i\}) = 0$ from the ground set of $G$ because then $G(A \cup \{i\}) = G(A)$ for any $A \subseteq V - \{i\}$. Hence, if $G(\{i\}) = 0$, then $F$ is linear with respect to that component, which in turn implies that $X_i$ is statistically independent from the other random variables.

The main insight behind the bounds that we will show is that given a function with a curvature strictly less than one, we can obtain both lower and upper bounds of $G$ that approximate it within a constant everywhere on $2^V$.

**Lemma 5.12** (Iyer, Jegelka & Bilmes [131, Lemma 3.1]). Given a polymatroid $G : 2^V \to \mathbb{R}$ with curvature $\kappa$, for all $A \subseteq V$ it holds that

$$G(A) \leq \sum_{i \in A} G(\{i\}) \leq \frac{|A|}{1 + (|A| - 1)(1 - \kappa)} G(A). \quad (5.26)$$

**Lemma 5.13.** Let $G : 2^V \to \mathbb{R}$ be a polymatroid with curvature $\kappa$, and take any extreme point $q$ of $B(G)$. Then, it holds that

$$\forall A \subseteq V : q(A) \leq F(A) \leq \frac{1}{1 - \kappa} q(A). \quad (5.27)$$
5.5 Approximation Guarantees

Proof.

\[ q'(A) = \sum_{i \in A} [G(i | \{\sigma^{-1}(1), \ldots, \sigma^{-1}(i-1)\})] \]

\[ \geq \sum_{i \in A} G(i | V - \{i\}) \]

\[ \geq (1 - \kappa) \sum_{i \in A} G(\{i\}) \]

\[ \geq (1 - \kappa)G(A). \]

From (5.25).

\[ z(A) = \sum_{i \in A} G(\{i\}) \text{ is in } \partial^G(\emptyset). \]

If we now combine Lemmas 5.10 and 5.11 and Lemmas 5.12 and 5.13, we arrive at the following result.

Corollary 4. Assume that \( F(A) = G(A) + m(A) \), for some polymatroid \( G \) and a modular \( m \). Then, if \( G \) has curvature \( \kappa \) and \( G_{\text{max}} = \max_{A \subseteq V} F(A) \), the following bounds hold

\[ \sum_{i=1}^{n} \log(1 + e^{+q_i}) \leq \log \sum_{A \subseteq V} \exp(+F(A)) \leq \kappa G_{\text{max}} \sum_{i=1}^{n} \log(1 + e^{q_i}) \]

\[ \sum_{i=1}^{n} \log(1 + e^{-q_i}) \leq \log \sum_{A \subseteq V} \exp(-F(A)) \leq \frac{\kappa}{\kappa - 1} G_{\text{max}} \sum_{i=1}^{n} \log(1 + e^{q_i}) \]

\[ \frac{\kappa}{\kappa - 1} G_{\text{max}} \sum_{i=1}^{n} \log(1 + e^{+q_i}) \leq \log \sum_{A \subseteq V} \exp(+F(A)) \leq \sum_{i=1}^{n} \log(1 + e^{q_i}) \]

\[ \kappa G_{\text{max}} \sum_{i=1}^{n} \log(1 + e^{-q_i}) \leq \log \sum_{A \subseteq V} \exp(-F(A)) \leq \sum_{i=1}^{n} \log(1 + e^{q_i}) \]

where \( q \) is any vertex of \( \mathcal{B}(G) \), and \( w \in \partial^F(A) \) has elements \( w_i = G(\{i\}) \).

Note that we establish bounds for specific sub-/super-gradients. Since our variational scheme considers these as well, the same quality guarantees hold for the optimized bounds. Further, note that we get a dependence on the range of the function via \( G_{\text{max}} \). However, if we consider \( \alpha F \) for large \( \alpha > 1 \), most of the mass will be concentrated at the MAP (assuming it is unique). In this case, our method also performs well, as it can always choose gradients that are tight at the MAP. When we optimize over super-gradients, all possible tight sets are considered. Similarly, the subgradients are optimized over \( \mathcal{B}(F) \), and for any \( A \subseteq V \) there exists some \( q_A \in \mathcal{B}(F) \) tight at \( A \).
5.5.2 Constant factor approximation for log-polymatroids

In the special case of log-polymatroidal distributions one can obtain constant-factor approximations of the partition function. We will moreover require that $F$ be not only be a polymatroid, but that we can also write $F$ as

$$F(A) = \sum_{j=1}^{m} F^j(A),$$

where each $F^j$ is itself a polymatroid with a computable concave extension $f^j_\wedge$. Formally, the considered models have the form

$$P(A) = \frac{1}{\mathcal{Z}} \exp(\sum_{j=1}^{m} F^j(A)).$$

For example, this assumption is satisfied if each $F^j$ is $M^-\neg$-concave. Shioura [69] analyzed exactly this model class for the problem of submodular maximization, building on the work of Calinescu et al. [65], who considered the special case when each $F^j$ is a weighted matroid rank function (Example 2.23). The primary reason for this additional condition is that we can then compute a concave extension\(^1\) of $F$ that is provably within a multiplicative constant of the multi-linear extension. Namely, remember from Theorem 2.5 that

$$\forall \mu \in [0, 1]^n: \left(1 - 1/e\right) f_\wedge(\mu) \leq f_\sim(\mu) \leq f_\wedge(\mu).$$

Hence, for each $F^j$ we have

$$\left(1 - 1/e\right) f^j_\wedge(\mu) \leq f_\sim(\mu) \leq f^j_\wedge(\mu),$$

and by summing these inequalities we obtain the bound

$$\forall \mu \in [0, 1]^n: \left(1 - 1/e\right) \sum_{j=1}^{m} f^j_\wedge(\mu) \leq f_\sim(\mu) \leq \sum_{j=1}^{m} f^j_\wedge(\mu) \equiv f_\sim(\mu). \quad (5.28)$$

Thus, the function $f_\sim = \sum_{j=1}^{m} f^j_\wedge$ is both a concave extension of $F$ and within $1 - 1/e$ of the multi-linear extension $f_\sim$.

---

\(^1\) Remember that the energy function is defined as $G = -F$, and its concave extension is $g_\wedge = -f_\vee$. 
Before we state the result and its proof, remember that in this case our energy function is \(-F\), so that the mean-field bound (Section 3.3.2) and the upper bound from the inclusive infinite divergence (Theorem 4.5) sandwich the log-partition function as follows

\[
\max_{\mu \in [0,1]^n} \mathbb{H}(\mu) + f_\sim(\mu) \leq \log Z \leq \max_{\mu \in [0,1]^n} \mathbb{H}(\mu) + f_\wedge(\mu) \tag{5.29}
\]

If we plug in \(f_\cap\) in lieu of \(f_\wedge\), we obtain a looser bound, namely

\[
\log Z \leq \max_{\mu \in [0,1]^n} \mathbb{H}(\mu) + f_\wedge(\mu) \leq \max_{\mu \in [0,1]^n} \mathbb{H}(\mu) + f_\cap(\mu), \tag{5.30}
\]

which we can efficiently optimize, as we have assumed that we can easily evaluate the concave extension \(f_\cap\). The first result that we show, is that this upper bound is a \(1 - 1/e\) constant approximation to \(\log Z\), which is an immediate consequence of inequality (5.28).

**Theorem 5.5.** Let \(\mu^*\) be the optimizer of \(\max_{\mu \in [0,1]^n} \mathbb{H}(\mu) + f_\cap(\mu)\). Then, it holds that

\[
(1 - 1/e)[\mathbb{H}(\mu^*) + f_\cap(\mu^*)] \leq \log Z \leq \mathbb{H}(\mu^*) + f_\cup(\mu^*). \tag{5.31}
\]

**Proof.** We can easily show this result as follows

\[
\mathbb{H}(\mu) + f_\cap(\mu) \leq \mathbb{H}(\mu^*) + (1 - 1/e)^{-1} f_\sim(\mu^*) \\
\leq (1 - 1/e)^{-1} [\mathbb{H}(\mu^*) + f_\sim(\mu^*)] \\
\leq (1 - 1/e)^{-1} \log Z,
\]

where the first inequality follows from equation (5.28), and the third one from the fact that the mean-field objective always lower-bounds the partition function. \(\square\)

We can moreover obtain a guaranteed lower bound that is within a constant factor of \(\log Z\). Specifically, solving the mean-field objective exactly will result in a constant factor approximation of \(\log Z\).

**Theorem 5.6.** Let \(\mu^o \in [0,1]^n\) be the vector holding the true marginals \(\mathbb{P}\), i.e., \(\mu_i = \mathbb{E}_\mathbb{P}[X_i = 1]\). Then

\[
\mathbb{H}(\mu^o) + f_\sim(\mu^o) \leq \log Z \leq \frac{e}{e - 1} [\mathbb{H}(\mu^o) - f_\sim(\mu^o)].
\]

\(\approx 1.582\)
Proof. The first inequality is exactly the bound from the exclusive KL divergence. To show the second inequality, note that

\[
0 = \mathbb{E}_P[\log \frac{P(A)}{P(A)}] \\
= \mathbb{E}_P[F(A) - \log Z] - \mathbb{E}_P[\log P(A)] \\
= -\log Z + \mathbb{E}_P[F(A)] + \mathbb{H}[P]
\]

which can be also obtained by expanding \(0 = D_{\text{KL}}(P \parallel P)\). If we re-arrange the terms, we arrive at

\[
\log Z = \mathbb{E}_P[F(A)] + \mathbb{H}[P].
\]

Because \(P\) and the fully factorized distribution \(\mu^o\) have the same marginals, i.e., \(P \in \mathcal{P}(\mu^o)\) in the notation of Theorem 2.3, and the fact that the exponential family model has the largest entropy among all distributions with marginals \(\mu^o\) (Theorem 3.3), it holds that

\[
\mathbb{H}[P] \leq \mathbb{H}(\mu^o).
\]

Similarly, from the probabilistic definition of \(f_\land\) (Theorem 2.3) we have that

\[
\mathbb{E}_P[F(A)] \leq f_\land(\mu^o).
\]

We can then combine these two observations to prove the theorem

\[
\log Z \leq \mathbb{H}(\mu^o) + f_\land(\mu^o) \\
\leq \mathbb{H}(\mu^o) + \frac{e}{e-1} f_\sim(\mu^o) \\
\leq \frac{e}{e-1}[\mathbb{H}(\mu^o) + f_\sim(\mu^o)].
\]

\[\square\]

Note that in the above theorem the bound holds for a specific \(\mu^o \in [0, 1]^n\) — the optimal one might give an even better bound, as it is not necessarily true that the true marginals form an optimum of the optimization problem. Moreover, solving the mean-field problem exactly will be in general a very hard problem. However, we will show that if we take the vector \(\mu^*\) maximizing the bound from the looser bound (5.30), and plug its solution in the mean-field bound we will obtain a \(e^2 / (e - 1)^2\)-factor approximation guarantee to \(\log Z\). This is of course not surprising, given that \(f_\cap\) and \(f_\sim\) are guaranteed to be within a multiplicative constant of each other.
Theorem 5.7. Let $\mathbf{\mu}^*$ be the optimizer of $\max_{\mathbf{\mu} \in [0,1]^n} \mathbb{H}(\mathbf{\mu}) + f_\cap(\mathbf{\mu})$. Then, it holds that

$$\mathbb{H}(\mathbf{\mu}^*) + f_\sim(\mathbf{\mu}^*) \leq \log Z \leq \left(\frac{e}{e-1}\right)^2[\mathbb{H}(\mathbf{\mu}^*) + f_\sim(\mathbf{\mu}^*)].$$ \hspace{1cm} (5.32)

Proof. Let $\mathbf{\mu}^o$ be the vector of true marginals of $P$. Then, we have that

$$f_\sim(\mathbf{\mu}^*) + \mathbb{H}(\mathbf{\mu}^*) \geq (1 - 1/e)[f_\wedge(\mathbf{\mu}^*) + \mathbb{H}(\mathbf{\mu}^*)]$$

$$\geq (1 - 1/e)[f_\wedge(\mathbf{\mu}^o) + \mathbb{H}(\mathbf{\mu}^o)]$$

$$\geq (1 - 1/e)[f_\sim(\mathbf{\mu}^o) + \mathbb{H}(\mathbf{\mu}^o)]$$

$$\geq (1 - 1/e)^2 \log Z,$$

from which the claim follows as $(1 - 1/e)^{-1} = e/(e - 1)$.

5.6 experiments

5.6.1 Synthetic models

The first set of experiments are on synthetic instances, and their purpose is to address the following questions.

(i) How large is the gap between the upper- and lower-bounds for the log-partition function and the marginals?

(ii) How accurate are the factorized approximations?

(iii) How does the accuracy depend on the amount of evidence (i.e., concentration of the posterior), the curvature of the function, and the type of model considered?

(iv) How does our method compare to mean-field, where it can be efficiently applied?

Note that our models provide both lower and upper bounds on the partition function. Hence, because the marginals are defined as a fraction of two partition functions (Section 3.1), we can obtain intervals that probably hold the true marginals. We consider approximate marginals obtained from the following four methods.

• Lower: Obtained from the exclusive infinity divergence, which yields a lower bound on the partition function. For log-submodular models
this is the greedy algorithm Algorithm 5, while for the log-supermodular models we picked the best bar supergradient using Lemma 5.6.

- **Upper**: Obtained from the inclusive infinity divergence, which yields an upper bound on the partition function. For log-supermodular models we used the divide-and-conquer algorithm (Algorithm 3) to compute the min-norm point, while for log-submodular we found the best bar supergradient using Lemma 5.3.

- **Lower bound**: The lower bound of the interval holding the true marginal.
- **Upper bound**: The upper bound of the interval holding the true marginal.

The functions that we use are all graph-representable (Section 2.6.2), so that we can easily minimize the energy function, and apply the methods we have proposed. Since the update equations are easily computable for graph-cuts, we have also implemented mean-field for the first experiment. The results are shown in Figure 5.3, and the experiments are explained below. We plot the averages of several repetitions of the experiments. Because computing the intervals requires two inference problem for each marginal, we have limited ourselves to small problems with at most 100 variables.

**Log-supermodular: cuts / pairwise MRFs**  
Our first experiment evaluates our methods on a sequence of distributions that are increasingly more concentrated. Motivated by applications in semi-supervised learning, we sampled data from a 2-dimensional Gaussian mixture model with 2 clusters. The centers were sampled from \( \mathcal{N}((3,3), I) \) and \( \mathcal{N}((-3, -3), I) \) respectively. For each cluster, we sampled \( n = 50 \) points from a bivariate normal. These \( 2n \) points were then used as nodes to create a graph with weight between points \( x \) and \( x' \) equal to \( e^{-||x-x'||} \). As prior we chose \( \mathbb{P}(A) \propto \exp(-F(A)) \), where \( F \) is the cut function in this graph, hence \( \mathbb{P}(A) \) is a regular MRF. Then, for \( k = 1, \ldots, n \) we consider the conditional distribution on the event that \( k \) points from the first cluster are on one side of the cut and \( k \) points from the other cluster are on the other side. As we provide more evidence, the posterior concentrates, and the intervals for both the log-partition function and marginals shrink. Compared to the ground truth, the estimates of the marginal probabilities improve as well. Due to non-convex objective, mean-field occasionally gets stuck in local optima, resulting in very poor marginals. To overcome this, we show the best bound out of 20 random restarts.
log-supermodular: decomposable functions  Our second experiment evaluates the performance as a function of the curvature of \(F\). It is motivated by a problem in outbreak detection on networks. Assume that we have a graph \(G = (V, E)\) and some of its nodes \(E \subseteq V\) have been infected by some contagious process. Instead of \(E\), we observe a noisy set \(N \subseteq V\), corrupted with a false positive rate of 0.1 and a false negative rate of 0.2. We used a log-supermodular prior

\[
P(A) \propto \exp\left(- \sum_{v \in V} \left( \frac{|N_v \cap A|}{|N_v|} \right)^\alpha \right),
\]

where \(\alpha \in [0, 1]\) and \(N_v\) is the union of \(v\) and its neighbors. This prior prefers smaller sets and sets that are more clustered on the graph. Note that \(\alpha\) controls the preference of clustered nodes and affects the curvature. We sampled random graphs with 100 nodes from a Watts-Strogatz model and obtained \(E\) by running an independent cascade starting from 2 random nodes. Then, for varying \(\alpha\), we consider the posterior, which is log-supermodular, as the noise model results in a modular likelihood. As the curvature increases, the intervals for both the log-partition function and marginals decrease as expected. Surprisingly, the marginals are very accurate (< 0.1 average error) even for very large curvature. This suggests that our curvature dependent bounds are very conservative, and much better performance can be expected in practice.

log-submodular: facility location  Our last synthetic experiment evaluates the accuracy of our approach when quantifying uncertainty in submodular maximization tasks. Concretely, we consider the problem of sensor placement in water distribution networks, which can be modeled as submodular maximization [132]. More specifically, we have a water distribution network and there are some junctions \(V\) where we can put sensors that can detect contaminated water. We also have a set \(\mathcal{J}\) of contamination scenarios. For each \(j \in \mathcal{J}\) and \(i \in V\) we have a utility \(w_{i,j} \in [0, 1]\), that comes from real data [132]. Moreover, as the sensors are expensive, we would like to use as few as possible. We use the facility-location model, more precisely

\[
P(A = A) \propto \exp(\sum_{j \in \mathcal{J}} \max_{i \in A} w_{i,j} - 2|A|).
\]

Instead of optimizing for a fixed placement, here we consider the problem of sampling from \(P\) in order to quantify the uncertainty in the optimization
task. We used the following sampling strategy. We consider the nodes from $V$ in a fixed order $(\sigma(1), \sigma(2), \ldots, \sigma(n))$. Then, at each round $i$ we sample a Bernoulli $Z$ with probability $\mathbb{P}(Z = 1) = Q(X_{\sigma(i)} = 1)$, where $Q$ is the approximate distribution from the inclusive divergence. Then, in the following rounds we condition on $\sigma(i) \in A$ if $Z = 1$, or $\sigma(i) \notin A$ if $Z = 0$. We repeated the experiment several times using randomly sampled 500 contamination scenarios and 100 locations from a larger dataset. Note that our approximations get better as we condition on more information (i.e., proceed through the iterations of the sampling procedure above). Also note that even from the very beginning, the marginals are very accurate ($< 0.1$ average error).

5.6.2 Image segmentation

We now report experimental results on applying the inclusive infinite divergence to a challenging image segmentation problem as motivated in Example 5.3. The goal of our experiments is to test the scalability of our approach to large problems using the connection to the min-norm point, and to evaluate the quality of the marginals both qualitatively and quantitatively. We used the data from Jegelka & Bilmes [35], which contains a total of 36 images, each with a highly detailed (pixel-level precision) ground truth segmentation. Due to intractability, we cannot compute the exact marginals against which we would ideally wish to compare. As a proxy for measuring the quality of the approximations, we use the area under the receiver operating characteristic (ROC) curve (AUC) as compared to the ground truth segmentation. We classify each pixel independently as fore- or background by comparing its approximate marginal against a threshold, which we vary to obtain the ROC curve. We have used the following energy

$$F(A) = \alpha m(A) + \beta F_{\text{cut}}(A) + \gamma \sum_{P_i \in \mathcal{P}} |P_i| \phi \left( \frac{|A \cap P_i|}{|P_i|} \right),$$

which contains both pairwise and higher-order interactions. Its components have been specified as follows.

- The unary potentials $m(\cdot)$ were learned from labeled data using a 5 component GMM.
(a) The performance of the proposed methods on graph cut instances as a function of the number of data points we condition on.

(b) The performance of the proposed methods on network inference instances as a function of the curvature.

(c) The performance of the proposed methods on the facility location problem as a function of the number of data points we condition on.

**Figure 5.3:** We plot the bounds on the log-partition function, the average gap between the upper and lower bounds on the marginals, and the absolute mean error of the marginals. Note that both the bounds get tighter and the marginals get more accurate as we either condition on more data, or reduce the curvature of the model.

- The pairwise potentials $F_{\text{cut}}$ connect neighboring pixels $\mathbf{x}$ and $\mathbf{x}'$ with weights $w(\mathbf{x}, \mathbf{x}') = \exp(-\theta \| \mathbf{x} - \mathbf{x}' \|^2)$, where $\mathbf{x}$ and $\mathbf{x}'$ are the RGB values of the pixels.
Figure 5.4: Example marginals from the different approximation procedures for the original image Figure 5.4a with ground truth segmentation Figure 5.4e. For the results comparing BP and DR we have used the same pairwise weights and weights. The numbers in the parenthesis correspond to the constants by which the unaries and the prior are multiplied (i.e. to \((\alpha, \beta)\) for the pairwise models and \((\alpha, \gamma)\) for the higher-order model). Note how BP is overconfident, whereas our methods offer marginals with much higher dynamic range.
• The higher order potentials were generated using the mean-shift algorithm of Comaniciu & Meer \cite{116}. We have used two overlapping layers of superpixels, each layer with different granularity. The concave function was defined as \( \phi(z) = z^{0.6}(1 - z)^{0.6} \).

We compared the following inference techniques. The reported typical running times are for an image of size 427 \( \times \) 640 pixels on a quad core machine and we report the wall clock time of the inference code (without setting up the factor graph or generating the superpixels).

• Unary potentials only with independent predictions, i.e., \( \beta = \gamma = 0 \).

• Belief propagation (BP) and mean-field (MF) for the pairwise model (i.e. \( \gamma = 0 \)). We have used the implementation from libDAI \cite{133}. The maximum number of iterations was set to 70. We note that this code is not parallelized. When we observe fast convergence, for example BP can converge in 3 iterations, it takes about 45 seconds. Even though we have set a relatively low number of iterations, the running times can be extremely slow if the methods do not converge. For example, running mean-field for 70 iterations can take more than 9 minutes.

• Our approach using only pairwise potentials (\( \gamma = 0 \)), solved using the total variation Douglas-Rachford (DR) code from \cite{134, 60, 51}. We ran for at most 100 iterations. The inference takes typically less than a second.

• The model having only higher order potentials (HOP) only (\( \beta = 0 \)). We have used a generalization of the algorithm from \cite{51}, which we present in Section 6.3.1. The inference takes less than 13 seconds.

For every method we tested several variants using different combinations for \( \alpha, \beta, \gamma \) and \( \theta \), as given in Table 5.2. Then, we performed a leave-one-out cross-validation for estimating the average AUC. We have also generated a sequence of 10 trimaps by growing the boundary around the true foreground to estimate accuracy over the hardest pixels, namely those at the boundary.

accuracy We first wish to quantitatively compare the accuracy of the approximate marginals. We report the aggregate results in Figure 5.3, and the ROC curves in Figure 5.5. We can clearly see that our approach outperforms the traditional inference methods for both objectives — the AUC
Parameter Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$</td>
<td>0.1, 0.001, 0.0001</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>1, 0.1, 0.01, 0.001</td>
</tr>
<tr>
<td>$\beta$</td>
<td>10, 1, 0.1, 0.01, 0.001</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>10, 1, 0.1, 0.01, 0.001</td>
</tr>
</tbody>
</table>

Table 5.2: The parameters used in the experiments.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HOP</td>
<td>0.9670</td>
<td>0.0549</td>
<td>0.9600</td>
<td>0.0608</td>
</tr>
<tr>
<td>DR</td>
<td>0.9568</td>
<td>0.0663</td>
<td>0.9465</td>
<td>0.0707</td>
</tr>
<tr>
<td>BP</td>
<td>0.9500</td>
<td>0.0636</td>
<td>0.9414</td>
<td>0.0791</td>
</tr>
<tr>
<td>MF</td>
<td>0.9500</td>
<td>0.0635</td>
<td>0.9441</td>
<td>0.0731</td>
</tr>
<tr>
<td>UNARY</td>
<td>0.9484</td>
<td>0.0658</td>
<td>0.9436</td>
<td>0.0736</td>
</tr>
</tbody>
</table>

Table 5.3: Average scores of the methods estimated using leave-one-out cross validation. The Avg. AUC column is the average area under the ROC curve. The Avg. AUCT column reports the average of the mean AUC over the 10 trimaps. The second and the fourth columns are the standard deviation of the preceding columns.

over the whole image and over the challenging boundary (trimaps). Sometimes we see very poor behavior of the alternative methods, which can be attributed to either their over-confidence (as verified below), or the fact that they optimize non-convex objectives and can fail to converge within the given number of iterations. Lastly, capturing high-order interactions leads to higher accuracy (in particular around the boundary) than pairwise potentials only.

Properties of the marginals  We would also like to understand the qualitative characteristics of the resulting marginals of our methods when compared with the traditional techniques. From the discussion on the properties of the inclusive infinite divergence in Chapter 4, we would expect the approximate marginals to avoid assigning low probabilities and rather prefer to err conservatively, i.e., on the side of causing false positives. On the other hand, it is known that the results of belief propagation are often over-confident. For this purpose, we provide a visual comparison in
Figure 5.5: Comparison of inference methods in terms of their accuracy. For each method we optimize the parameters via grid-search, and report leave-one-out cross-validation results. (a) ROC curves for classifying pixels as fore- or background by independently thresholding marginals, averaged over the whole image. (b) Results over the trimaps (blurred boundaries around the ground truth segmentation), focusing on “difficult” pixels. For every algorithm and every of the 10 trimap sizes we report the average area under the curve.

Figure 5.4. Namely, each of the four BP/DR pairs are results using the respective algorithms for the same parameters of the model. We observe exactly what the theory predicts — the distribution obtained using our method is less concentrated around the object and mass is spread around more. The contrast is starkest on Figures 5.4b and 5.4f, where we use a strong pairwise prior (high $\beta$). In Figures 5.4h and 5.4j we have used a weak pairwise prior (low $\beta$), and as expected the resulting marginals are mainly determined by the unary part and the choice of inference procedure does not make a difference. The results in the last column are from the higher order model, with two different values of $\gamma$ (the strength of the higher order potential). We can see that the resulting probabilities better preserve the boundaries of the object and the fine details, which is one of the main benefits of using these models.
5.7 CONCLUSION

In this chapter we have shown how to leverage submodularity, without any factorization assumptions, to obtain efficient approximate inference algorithms for general log-submodular and log-supermodular models. These methods are based on the infinite Rényi divergences, and yield not only fully factorized approximations, but also provable upper or lower bounds on the partition function. To minimize the exclusive divergence for log-submodular distributions we have motivated a greedy algorithm resembling those for submodular maximization. In addition to being very easy to implement, we were also able to leverage the theory of maximization to relate the modes of the approximate and the true distributions. In the case of log-supermodular distributions, we reduced the problem of minimizing the inclusive divergence to the well-studied min-norm problem. This gives us not only an immediate proof that the approximation shares the MAP with the true distribution, but we can immediately leverage the large body of efficient algorithms for this well-studied problem. This let us tackle problems with hundreds of thousands of variables and complex high-degree factors. In the experimental section we have shown that the conjunction of such rich models and our inference method yields approximate marginals that make better trade-off in terms of the ROC curve than traditional schemes. We have also experimentally analysed how our bounds vary with the inclusion of additional information and change in curvature on three different models. Finally, we have also proven theoretical guarantees that connect the curvature with the precision of the approaches, and also gave constant-factor approximations for log-polymatroidal distributions by leveraging the correlation-gap inequality.
INFECTION IN MIXED SUBMODULAR MODELS

Most of the material in this chapter has been already published in the following conference proceedings.


Moreover, while I have contributed to the code to both learn and perform inference in the distributions used in the experimental section, the presented experiments were implemented and executed by Dr. Sebastian Tschiatschek.

6.1 INTRODUCTION

In the previous chapter we have shown how to develop divergence minimization algorithms for log-submodular and log-supermodular distributions without any factorization assumptions. While these families contain models that can capture very useful behaviours, there exist settings where even minor natural modifications to these distributions would render them outside the reach of the proposed algorithms.

For example, we have seen that log-supermodular models are closed under several important operations, and can only model positive correlations. However, this modelling assumption can be too strong, as illustrated in the setting of Rother et al. [135]. Namely, they consider the problem of image segmentation with user feedback. They obtain user input in the form of hints about the class boundaries — the user selects a set of edges with the understanding that the vertices connected by them should be assigned to different labels. This of course motivates the inclusion of repulsive potentials that would negatively correlate the vertices along the boundary. Similarly, Kawahara, Iyer & Bilmes [136] also use log-submodular potentials along the edges of the image, which are automatically computed using an
edge detector. Unfortunately, these extensions can not be easily captured using a log-supermodular model.

Let us also consider the FLID model introduced in Example 5.5. Remember that the main motivation behind the model was that of modelling items that are substitutes of one another — for example, the possession of a smartphone decreases the utility of a camera as both of them have the ability to take images. However, after buying a phone, one might also want to also buy a screen protector for it — in other words, these two items are complements of one another, and we would expect a positive correlation between them. To accommodate such models and capture both of these modalities, we could extend the FLID model with log-supermodular terms, but that would result in a model that does not satisfy the assumptions made by the inference schemes developed in the previous chapter. The fact that the inclusion of such terms can significantly improve the accuracy of the model will be also experimentally corroborated later in this chapter.

In this chapter we will discuss a generalization of the model classes introduced in the previous chapter that can accommodate the two extensions we have motivated above. Specifically, we consider models of the form

$$P(A) = \frac{1}{Z} \exp(F(A) - G(A)),$$  \hspace{1cm} (6.1)

where $F$ and $G$ are both assumed to be submodular functions. We will refer to the class of all such distributions as \textit{mixed submodular models}.

The existing literature has been mostly focused on the problem of computing a MAP configuration of these models. It has been studied by Iyer & Bilmes [137] and Narasimhan & Bilmes [138], who also prove that any set function can be written as a difference of two submodular functions — hence, the model (6.1) can capture \textit{any} distribution over Boolean variables. The complication arises from the fact that, as the authors in [138] note, such a decomposition is in general hard to find. In both papers the authors leverage the differential structure of submodular functions by linearizing $F$ or $G$, and then optimizing the resulting problem. In our approach we have to be able to perform inference after the linearization, so we have to in general linearize both $F$ and $G$. This approach is most related to the ModMod procedure for MAP computation from [137, §4.2].

The remaining of the chapter is organized as follows. We first treat the problem when the energy can be written as the sum of $m$ simpler functions, without making any submodularity assumptions. Specifically, we will show how to minimize the divergences when the energy function is
itself not tractable, but each of its components can be minimized or maximized. Then, we will specialize these bounds to the mixed models we have just introduced. Finally, we will experimentally showcase the benefits of both this extended model family and the resulting approximate marginals on a product recommendation task.

6.2 INFEREN CE IN FACTORIZED MODELS

We begin by analyzing the general case where the energy admits a factorization of the form

\[ F(A) = \sum_{j=1}^{m} F_j(A), \]

for some arbitrary set functions \( F_j : 2^V \rightarrow \mathbb{R} \). In other words, we assume that the distribution factorizes and can be written as

\[ P(A) \propto \prod_{j=1}^{m} \exp(-F_j(A)). \]

We will not make any additional assumptions about \( F_j \), but to efficiently apply the methods we will now develop, one should be able to either minimize (for the inclusive) or maximize (for the exclusive divergence) each \( F_j \). In the remaining of the section we treat \( F \) and each \( F_j \) as being defined over binary vectors \( x \in \mathcal{X} = \{0, 1\}^n \), as that would make the notation much simpler. Like in the previous chapter, as the approximation family \( Q \) we use all fully factorized distributions, i.e., those of the form

\[ Q(x) = \exp(-q^\top x - A(-q)), \]

for some vector of parameters \( q \in \mathbb{R}^n \).

6.2.1 Dual decomposition and expectation propagation

Let us start with the inclusive divergence, which takes the following form

\[ \log Z \leq \inf_{q \in \mathbb{R}^n} A(-q) + \max_{x \in \mathcal{X}} [x^\top q - \sum_{j=1}^{m} F_j(x)]. \quad (6.2) \]

The bound would of course still hold if we compute some upper bound of the maximum. One way to obtain a bound is via duality, i.e., we would like to construct a dual function that always upper bounds the maximum
in (6.2). A principled way to construct such a dual is the method of dual decomposition, which is applicable when we can minimize each $F_j$ individually. It has a long history in optimization (see e.g. Bertsekas [139]), while it was first proposed for the computation of MAP problems in graphical models by Komodakis, Paragios & Tziritas [140] (for a longer survey see [141]). We will now state the theorem, which a straightforward application of dual decomposition, and follows the argument in [141, §4], that provides us a tractable upper bound whenever each component $F_j$ can be efficiently minimized under a linear perturbation.

**Theorem 6.1.** The bound from the inclusive infinite Rényi divergence can be relaxed to

$$\log \mathcal{Z} \leq \inf_{\lambda_j \in \mathbb{R}^n} A(- \sum_{j=1}^{m} \lambda_j) + \sum_{j=1}^{m} \sup_{x \in \mathcal{X}} [\lambda_j^\top x - F_j(x)].$$  

(6.3)

**Proof.** By introducing new variables, we can trivially rewrite (6.2) as

$$A(-q) + \max_{x_j \in \mathcal{X}, x \in \mathbb{R}^n} \sup_{s.t. \ x = x_j} \inf \sum_{j=1}^{m} \lambda_j^\top (x - x_j).$$

After introducing Lagrange multipliers to enforce the constraints, one $\lambda_j \in \mathbb{R}^n$ for each $j$, we can rewrite the above objective as

$$A(-q) + \sup_{x_j \in \mathcal{X}} \inf_{x \in \mathbb{R}^n} \sum_{j=1}^{m} \lambda_j^\top (x - x_j) + q^\top x - \sum_{j=1}^{m} F_j(x),$$

where note that the infimum inside is equal to $-\infty$ unless all $m$ constraints $x_j = x$ hold. If we swap the inf and the sup we will upper-bound the above quantity and obtain

$$A(-q) + \inf_{\lambda_j \in \mathbb{R}^n} \sup_{x_j \in \mathcal{X}} \inf_{x \in \mathbb{R}^n} \sum_{j=1}^{m} \lambda_j^\top (x - x_j) + q^\top x - \sum_{j=1}^{m} F_j(x),$$

which with some algebra can be reduced to

$$A(-q) + \inf_{\lambda_j \in \mathbb{R}^n} \sup_{x \in \mathbb{R}^n} x^\top (q - \sum_{j=1}^{m} \lambda_j) + \sum_{j=1}^{m} [\sup_{x_j \in \mathcal{X}} \lambda_j^\top x_j - F_j(x_j)].$$
Note that if \( q \neq \sum_{j=1}^{m} \lambda_j \) then the supremum with respect to \( x \) will be infinite. Hence, we can easily optimize out \( x \) by making this constraint explicit, which results in in the problem

\[
A(-q) + \inf_{\lambda_j \in \mathbb{R}^n} \sum_{j=1}^{m} \left[ \sup_{x_j \in \mathcal{X}} \lambda_j^\top x_j - F^j(x_j) \right].
\]

If we then substitute for \( q \) we obtain the claimed result.

We can analogously apply the same technique to the lower bound, as formalized in the next theorem.

**Theorem 6.2.** The bound from the exclusive infinite Rényi divergence can be relaxed to

\[
\log \mathcal{Z} \geq \sup_{\lambda_j \in \mathbb{R}^n} A(-\sum_{j=1}^{m} \lambda_j) + \sum_{j=1}^{m} \inf_{x \in \mathcal{X}} [\lambda_j^\top x - F^j(x)].
\] (6.4)

**Proof.** The proof closely follows that of the previous theorem, and is provided only for completeness. Remember that the lower bound has the form

\[
\sup_{q \in \mathbb{R}^n} A(-q) + \inf_{x \in \mathcal{X}} [q^\top x - \sum_{j=1}^{m} F^j(x)],
\]

whose objective we can equivalently write as

\[
A(-q) + \inf_{x \in \mathcal{X}} \inf_{x \in \mathbb{R}^n} \sup_{\lambda_j \in \mathbb{R}^n} \sum_{j=1}^{m} \lambda_j^\top (x_j - x) + q^\top x - \sum_{j=1}^{m} F^j(x).
\]

Swapping the sup and the inf results in the following lower bound

\[
A(-q) + \sup_{\lambda_j \in \mathbb{R}^n} \inf_{x \in \mathcal{X}} \inf_{x \in \mathbb{R}^n} \sum_{i=1}^{m} \lambda_i^\top (x_j - x) + q^\top x - \sum_{j=1}^{m} F^j(x).
\]

Again, optimizing out \( x \) results in the following constrained problem

\[
A(-q) + \sup_{\lambda_j \in \mathbb{R}^n} \sum_{j=1}^{m} \inf_{x_j \in \mathcal{X}} [x_j^\top \lambda_j - F^j(x_j)],
\]

which is exactly what we had to show after substituting for \( q \).

\[ \square \]
Having defined these relaxations, let us discuss the algorithmic aspects of the resulting problems. Note that (6.3) is a convex minimization task whose subgradients can be computed by solving the perturbed MAP problems for each $j$. In contrast, the problem resulting from the exclusive divergence is not that of concave maximization due to the presence of the log-partition function, which is convex. We can nevertheless again obtain gradients by maximizing each $F_j$ separately.

Another strategy of updating these relaxations is using block-coordinate-wise optimization. Namely, at each iteration we propose to minimize (maximize) the objective (6.3) ((6.4)) with respect to a single multiplier $\lambda_j \in \mathbb{R}^n$. If we w.l.o.g. assume that we optimize with respect to the first parameter $\lambda_1$, we obtain the problems

$$\inf_{\lambda_1 \in \mathbb{R}^n} A(-\lambda_1 - \sum_{j=2}^{m} \lambda_j) + \sup_{x \in \mathcal{X}} [x^\top \lambda_1 - F^1(x)],$$

and

$$\sup_{\lambda_1 \in \mathbb{R}^n} A(-\lambda_1 - \sum_{j=2}^{m} \lambda_j) + \inf_{x \in \mathcal{X}} [x^\top \lambda_1 - F^1(x)].$$

If we define the auxiliary distribution

$$\mathbb{P}^{-1}(x) \propto \exp(-F^1(x) - \sum_{j=2}^{m} \lambda_j^\top x),$$

then, in both cases, an optimal $\lambda_1^*$ can be computed from

$$\min_{\mathbb{Q}} D_\infty(\mathbb{P}^{-1} \| \mathbb{Q}),$$

and

$$\min_{\mathbb{Q}} D_\infty(\mathbb{Q} \| \mathbb{P}^{-1})$$

respectively. (6.7)

Specifically, the optimal parameter is equal to $\lambda_1^* = \lambda - \sum_{j=2}^{m} \lambda_j$, if the distribution $\mathbb{Q}^{-1}_*(x) \propto \exp(-\lambda^\top x)$ minimizes the auxiliary problem (6.7).

The above strategy turns out to be a special case of the general expectation propagation (EP) algorithm, which is typically used for the inclusive KL divergence. Namely, Minka [96] has shown that we can generalize EP to any divergence measure whenever we have a distribution of the form

$$\mathbb{P}(x) = \prod_{j=1}^{m} \mathbb{P}_j(x) / \mathcal{Z},$$
for some distributions $P_j$. The main idea is to approximate each $P_j$ with a distribution $Q_j$, which in our case will be a completely factorized distribution, such that the product $\prod_{j=1}^{m} Q_j / Z_Q$ is a good approximation to $P$ in terms of the given divergence. Formally, he suggests to optimize iteratively using the following procedure.

(i) Pick a factor $j \in \{1, 2, \ldots, m\}$.

(ii) Replace the other factors $P_{j'}$ for $j' \neq j$ with their approximations $Q_{j'}$.

(iii) Update $Q_j$ by minimizing the divergence between $P_i \prod_{j' = 1, j' \neq j}^{m} Q_{j'}/Z'$ and $\prod_{j=1}^{m} Q_j / Z_Q$.

To better understand the problem resulting from the last step, let us w.l.o.g. assume that we want to update the first block $Q_1$. Then, if we use the inclusive divergence, we have to solve

$$\min_{Q_1} D_\infty \left( \frac{1}{Z'} P_1 \prod_{j=2}^{m} Q_j \mid \frac{1}{Z} \prod_{j=1}^{m} Q_j \right).$$

If we use fully factorized distributions $Q_j$ parametrized as $Q_j(x) \propto \exp(-\lambda_j^\top x)$ for some $\lambda_j \in \mathbb{R}^n$, then

$$\prod_{j=1}^{m} Q(j) = \exp(-\sum_{j=1}^{m} \lambda_j^\top x),$$

and

$$\log Z_Q = A(-\sum_{j=1}^{m} \lambda_j).$$

Applying Lemma 4.1, the problem reduces to

$$\inf_{\lambda_1 \in \mathbb{R}^n} A(-\sum_{j=1}^{m} \lambda_j) + \max_{x \in X} \left[ \sum_{j=1}^{m} \lambda_j^\top x - (F_1(x) + \sum_{j=1}^{m} \lambda_j^\top x) \right],$$

which is exactly equal to (6.5) if we cancel out the term $\sum_{j=2}^{m} \lambda_j$ from the inner optimization problem. Moreover, if we instead use the exclusive divergence, then we will analogously arrive at equation (6.6). Hence, optimizing the dual decomposition bounds for the infinite divergences yields the same updates as expectation propagation.

It is known that the convergence of the EP algorithm does not always hold, and it is moreover not clear how to choose the order in which we update the blocks. Fortunately, due to the fact that the inclusive divergence is
a convex optimization problem with a special structure, we can guarantee convergence if we choose the blocks in a sequential manner (also known as the Gauss-Seidel rule), or as long as we update each block every \( T \) updates for any \( T \in \mathbb{N} \) (the essentially cyclic rule). Namely, because the function that couples the blocks is a smooth (bounded Hessian eigenvalues), and the non-smooth components are separable, we can use the results of Hong et al. [142]. Specifically, applying [142, Theorem 6.1] we obtain a convergence rate of \( O(1/k) \), where \( k \) is the iteration number. Hence, not only that we are guaranteed that EP with the inclusive divergence will converge, but we also have a provable rate. The only technical condition that we need is that the loss has bounded level sets, which we prove below.

**Lemma 6.1.** Problem (6.4) has bounded level sets.

**Proof.** We will show that it has one non-empty bounded level set, which implies that all level sets are bounded, as the function is closed [143, Corollary 4D]. First, we can construct a simple lower bound on the objective as follows

\[
A(-\sum_{j=1}^{m} \lambda_j) + \sum_{j=1}^{m} \max_{x \in X} [\lambda_j^\top x - F_j(x)] = \\
\sum_{j=1}^{m} \log(1 + e^{-\sum_{i=1}^{n} \lambda_{j,i}}) + \sum_{j=1}^{m} \max_{x \in \{0,1\}^n} [\lambda_j^\top x - F_j(x)] \geq \\
\sum_{j=1}^{m} \max\{0, -\sum_{i=1}^{n} \lambda_{j,i}\} + \sum_{j=1}^{m} \max_{x \in \{0,1\}^n} [\lambda_j^\top x - F_j(x)] \geq \\
\sum_{j=1}^{m} \max\{0, -\sum_{i=1}^{n} \lambda_{j,i}\} + \sum_{j=1}^{m} [\max_{i=1}^{n} \lambda_{j,i} - \min_{x \in \{0,1\}^n} F_j(x)] = \\
\sum_{j=1}^{m} \max\{0, -\sum_{i=1}^{n} \lambda_{j,i}\} + \sum_{j=1}^{m} [\sum_{i=1}^{n} \max\{0, \lambda_{j,i}\}] - \min_{x \in \{0,1\}^n} F_j(x). 
\]

If we define for each \( i \in \{1,2,\ldots,n\} \) the numbers

\[
K_i = \sum_{j=1}^{m} |\lambda_{j,i}|, \\
\pi^+_i = \sum_{j: \lambda_{j,i} > 0} |\lambda_{j,i}|, \text{ and} \\
\pi^-_i = \sum_{j: \lambda_{j,i} < 0} |\lambda_{j,i}|, 
\]
we have that $\pi_i^- = K_i - \pi_i^+$, and the above bound can be written as

$$\sum_{i=1}^{n} \max\{0, \pi_i^- - \pi_i^+\} + \sum_{i=1}^{n} \pi_i^+ - C = \sum_{i=1}^{n} \max\{0, K_i - 2\pi_i^+\} + \sum_{i=1}^{n} \pi_i^+ - C.$$  

If we fix each $K_i$ and minimize this bound subject to $\pi_i^+ \in [0, K_i]$, we can see from the monotonicity of the objective on $[0, K_i/2]$ and $[K_i/2, K_i]$ that the optimum of $\sum_{i=1}^{n} K_i/2 - C$ is achieved at $\pi_i^+ = K_i/2$. Moreover, the objective value of the vector all zeros is equal to $n \log 2 - C$. Hence, all vectors in the level set containing the zero vector has to be contained in

$$\Lambda = \{(\lambda_1, \lambda_2, \ldots, \lambda_n) | \forall i \in \{1, 2, \ldots, n\}: \sum_{j=1}^{m} |\lambda_{j,i}| \leq 2n \log 2\},$$

which is itself a bounded set (a product of $n$ norm balls), and this completes the proof.

6.2.2 The polyhedral view

We can also arrive at the above bounds by approximating the polyhedra $\mathcal{L}(F)$ and $\mathcal{U}(F)$. Namely, remember the primal polyhedral form of the bounds (Theorems 4.2 and 4.3)

$$\log Z \geq \sup_{(q,c) \in \mathcal{U}(F)} A(-q) - c, \text{ and}$$

$$\log Z \leq \inf_{(q,c) \in \mathcal{L}(F)} A(-q) - c. \quad (6.8)$$

The problem is that we cannot succinctly represent neither $\mathcal{L}(F)$ nor $\mathcal{U}(F)$, so we have to approximate them. Moreover, these approximations need to be inner bounds of $\mathcal{L}(F)$ and $\mathcal{U}(F)$, so that we still have provable bounds on the log-partition function. We can obtain such bounds by leveraging the lower and upper polyhedra of each $F^j$ as

$$\mathcal{L}(F) \supseteq \sum_{j=1}^{m} \mathcal{U}(F^j), \text{ and}$$

$$\mathcal{U}(F) \supseteq \sum_{j=1}^{m} \mathcal{L}(F^j), \quad (6.10)$$

and
where the sum is defined in the Minkowski sense. Hence, plugging in these approximations in the bounds above we obtain the problems

\[
\log Z \leq \inf_{(\lambda_j, c_j) \in \mathcal{L}(F^i)} A\left(-\sum_{j=1}^{m} \lambda_j\right) - \sum_{j=1}^{m} c_j, \quad \text{and} \quad (6.12)
\]

\[
\log Z \geq \sup_{(\lambda_j, c_j) \in \mathcal{U}(F^i)} A\left(-\sum_{j=1}^{m} \lambda_j\right) - \sum_{j=1}^{m} c_j. \quad (6.13)
\]

Similar to the proofs of Theorems 4.2 and 4.3, we can easily see that for any set of vectors \( \lambda_j \in \mathbb{R}^n \) the optimal offsets given by

\[
c_{1,j} = \inf_{x \in \mathcal{X}} [F^i(x) - \lambda_j^\top x], \quad \text{and}
\]

\[
c_{2,j} = \sup_{x \in \mathcal{X}} [F^i(x) - \lambda_j^\top x]
\]

satisfy \((\lambda_j, c_{1,j}) \in \mathcal{U}(F^i)\) and \((\lambda_j, c_{2,j}) \in \mathcal{L}(F^i)\). Plugging these in the inequalities above we arrive at the same bounds that we have derived using dual decomposition.

### 6.3 Inference in Mixed Submodular Models

We now consider the setting of mixed models — when each of component \( F^i \) of \( F \) is either submodular or supermodular. Note that if \( F_j \) is submodular or supermodular, then so is the energy function of \( P_{-j} \) in problem (6.7). Hence, we can easily plug in the inference algorithms we have developed in the previous chapter in the framework we have developed above to perform inference in this model class.

Moreover, it is noteworthy that in this case we can simplify the polyhedral upper bound resulting from the inclusive divergence as follows.

**Lemma 6.2.** Assume that functions \( F^1, F^2, \ldots, F^k \) are submodular and \( F^{k+1} = -H^{k+1}, F^{k+2} = -H^{k+2}, \ldots, F^m = -H^m \) are supermodular. Then, Problem (6.3) achieves the same optimum as

\[
\inf_{\{\lambda_j \in \mathcal{B}(F^i)\}_{j=1}^{k}} \inf_{\{(\lambda_j, c_j) \in \mathcal{U}(H^i)\}_{j=k+1}^{m}} A\left(\sum_{j=k+1}^{m} \lambda_j - \sum_{j=1}^{k} \lambda_j\right) + \sum_{j=k+1}^{m} c_j. \quad (6.14)
\]
Proof. Fix in Problem (6.3) all variables except for w.l.o.g. $\lambda_1$. If $\lambda^*_1$ is optimal, it has to be also optimal for the problem

$$\inf_{\lambda_1 \in \mathbb{R}^n} A(-\lambda_1 - \sum_{j=2}^m \lambda_j) + \sup_{x \in \mathcal{X}} \left[ x^\top \lambda_1 - F^1(x) \right].$$

If we now apply Lemma 5.7, we obtain that there is always an optimum of (6.15) that lies on $B(F^1)$, and we obtain that (6.3) achieves the same optimum as

$$\inf_{\{\lambda_j \in B(F^j)\}_{j=1}^k} \inf_{\{\lambda_j \in \mathbb{R}^n\}_{j=k+1}^m} A(-\sum_{j=2}^m \lambda_j) + \sum_{j=k+1}^m \sup_{x \in \mathcal{X}} [\lambda_j^\top x + H^j(x)].$$

Now, the claim follows by including variables $-c_j$ to capture the suprema and noting that $\mathcal{L}(-H^i) = -\mathcal{U}(H^i)$.

Hence when minimizing the upper bound for the inclusive divergences, we effectively make the following set of approximations

$$\mathcal{L}(F - G) \supseteq \mathcal{L}(\sum_{j=1}^k F^j) - \mathcal{U}(\sum_{j=k+1}^m G^j) \supseteq \sum_{j=1}^k B(F^j) \times \{0\} - \sum_{j=k+1}^m \mathcal{U}(G^j),$$

where the relaxation $\mathcal{L}(\sum_{j=1}^k F^j) \supseteq \sum_{j=1}^k B(F^j) \times \{0\}$ is without loss of precision in the optimization, as argued in the proof of Lemma 6.2.

### 6.3.1 Inference in factorized log-supermodular models

In the special setting where the distribution is purely log-supermodular ($k = m$), we can develop specialized algorithms for the convex problem resulting from the inclusive divergence. Namely, not only we can exactly compute the block-wise updates, but we can also obtain faster convergence rates. Specifically, by leveraging Theorem 5.4, the update with respect to coordinate $\lambda_1$ is given as

$$\lambda_1 \leftarrow \arg\min_{\lambda_1 \in B(F^1) + \sum_{j=2}^m \lambda_j} \frac{1}{2} \|\lambda_1\|^2 - \sum_{j=2}^m \lambda_j,$$

which is equivalent to

$$\lambda_1 \leftarrow \arg\min_{\lambda_1 \in B(F^1)} \| \sum_{j=1}^m \lambda_j \|^2.$$
This is exactly the same as performing coordinate descent for the min-norm problem. This algorithm has been analyzed by Ene & Nguyen [144], who show that the accelerated coordinate descent method of Nesterov [145] has a linear convergence rate on this problem. Hence, if we have only log-supermodular potentials, we can not only compute the exact updates in polynomial time, but we can also obtain provably significant computational speedups.

One downside of the above approach is that it has to be applied sequentially, i.e., one factor has to be updated at a time to ensure convergence. An alternative is to apply an approach used by Jegelka, Bach & Sra [51], which allows to perform message passing in parallel without losing the convergence guarantees. While they consider the minimization of the dual of the min-norm problem \( f_L(x) + \frac{1}{2} \|x\|^2 \), this is equivalent to our setting, due to the equivalence in Theorem 5.4. Their approach assumes that each \( F^j \) is dependent on all variables. Instead, we will generalize their algorithm to the case where \( F^j \) depends only on some subset \( V_j \subseteq V \), so that it is defined as \( F^j : 2^{V_j} \rightarrow \mathbb{R} \). We do this, as it will both result in a faster convergence rate, and have an interpretation in terms of message passing on the factor graph. Namely, the factor graph is the bipartite graph with one node for each element of \( V \) on one side, and one node for each factor \( F^j \) in the other. Then, we connect \( F^j \) to all nodes corresponding to \( V^j \). For any node \( u \) in the factor graph, we shall denote by \( \delta(u) \) its set of neighbours.

Let us first define the resulting problem, which is again derived using dual decomposition. For any \( x \in \mathbb{R}^n \) and any \( A \subseteq V \) we will define by \( x_A \) the vector in \( \mathbb{R}^A \) consisting of the coordinates of \( x \) indexed by \( A \). Note that the index set of \( x_A \) is \( A \) rather than \( \{1, 2, \ldots, |A|\} \).

**Definition 6.1.** For any \( x_A \in \mathbb{R}^A \), where \( A \subseteq V \), define the pair of norms
\[
\|x_A\|^2_G = \sum_{i \in A} \frac{1}{|\delta(i)|} x_i^2, \quad \text{and} \quad \|x_A\|^2_{G^*} = \sum_{i \in A} |\delta(i)| x_i^2.
\]

**Theorem 6.3** (Extension of Jegelka, Bach & Sra [51, Lemma 1]). We have the following duality
\[
\inf_{x \in \mathbb{R}^n} f_L(x) + \frac{1}{2} \|x\|^2 = \sup_{\pi \in \Pi} \sum_{j=1}^m \sup_{q_j \in B(F_j)} -\frac{1}{2} \|q_j - \pi_j\|^2_{G^*}, \quad (6.17)
\]
where the set \( \Pi \) is defined as
\[
\Pi = \{ (\pi_1, \pi_2, \ldots, \pi_m) : \pi_j \in \mathbb{R}^{V_j} \text{ and } (\forall i \in V) \sum_{F_j \in \delta(i)} \pi_{j,i} = 0 \}.
\]
Proof. The proof is a slight modification of that of [51, Lemma 1]. Komodakis, Paragios & Tziritas [140] also do not assume that the factors depend on all variables and obtain similar equations that average each factor by its size. We start, similarly to the other dual decomposition bounds above, by introducing for each component $F_j$ a variable $x_j \in \mathbb{R}^{V_j}$ and writing the problem as

$$\inf_{x \in \mathbb{R}^n} f_L(x) + \frac{1}{2}\|x\|^2 = \inf_{x \in \mathbb{R}^n, x_j \in \mathbb{R}^{V_j}} \sum_{j=1}^{m} (f^j_L(x_j) + \frac{1}{2}\|x_j\|^2_G) \text{ s.t. } x_{V_j} = x_j$$

$$= \inf_{x, x_{V_j}} \sup_{\pi_j} \sum_{j=1}^{m} [f^j_L(x_j) + \frac{1}{2}\|x_j\|^2_G - \pi_j^\top (x_j - x_{V_j})],$$

where $\pi_j$ is the Lagrange multiplier for the constraint $x_j = x_{V_j}$. Using Theorem A.3 theorem we can change the order of optimization. Then, if we focus on the optimization wrt $x$, we see that the Lagrange multipliers have to belong to $\Pi$, as otherwise the optimum will be infinite. Hence, by plugging in the definition of the Lovász extension we have the following problem

$$\sup_{\pi \in \Pi} \sum_{j=1}^{m} \inf_{x_j} \sup_{q_j \in \mathcal{B}(F_j)} [x_j^\top q_j + \frac{1}{2}\|x_j\|^2_G - \pi_j^\top x_j]$$

We can again swap the inf and the sup due to Theorem A.3, and obtain that the minimization with respect to $x_j$ is equivalent to

$$\inf_{x_j \in \mathbb{R}^{V_j}} x_j^\top q_j + \frac{1}{2}\|x_j\|^2_G - \pi_j^\top x_j = \inf_{x_j \in \mathbb{R}^{V_j}} x_j^\top (q_j - \pi_j) + \frac{1}{2}\|x_j\|^2_G,$$

which is exactly the negative of the convex conjugate of $\frac{1}{2}\|\cdot\|^2_G$ evaluated at $\pi_j - q_j$. Because the convex conjugate is equal to $\frac{1}{2}\|\cdot\|^2_{G^*}$ (see e.g. [127, Ex. 3.22]), the above infimum is equal to $-\frac{1}{2}\|q_j - \pi_j\|^2_{G^*}$, which we had to show. \qed

We then alternatingly optimize with respect to $(q_1, q_2, \ldots, q_m) \in \prod_{j=1}^{m} B(F_j)$ and $(\pi_1, \pi_2, \ldots, \pi_m) \in \Pi$. Note that this means that we will update all $q_j$ in parallel in contrast to the approach above. The update with respect to $\pi_j$ amount to simple subtracting of $\frac{1}{m} \sum_{j=1}^{m} q_j$, while the $q_j$ updates simply project onto the base polytope under the $\|\cdot\|_{G^*}$ norm — a separable convex problem solvable using the divide-and-conquer algorithm (Algorithm 3).
As we have already mentioned above, these updates can be written as message passing in the factor graph. Specifically, let us denote the message sent from node \( u \) to node \( w \) at iteration \( t \) by \( \mu_{u \rightarrow w}^t \). The messages from variables \( i \in V \) to factors \( F_j \) are simple sums, similar to those in standard belief propagation

\[
\mu_{i \rightarrow F}^{t+1} = \frac{1}{|\delta(i)|} \sum_{F_j \in \delta(i)} \mu_{F \rightarrow i}^t.
\]

If \( \mu_{\rightarrow F}^t \in \mathbb{R}^V \) are the messages received by factor \( j \) at time \( t \), then it computes

\[
q_{j}^{t+1} = \arg\min_{q_j \in \mathcal{B}(F_j)} \|q_j - (q_j^t - \mu_{\rightarrow F}^t)\|_{G^*}^2,
\]

and sends its neighbours the following messages

\[
\mu_{F_i \rightarrow v}^{t+1} = q_{j,i}^{t+1}.
\]

At every iteration \( t \) the approximate distribution \( Q^t \) can be computed from the messages as

\[
Q^t(x) \propto \prod_{i=1}^m \exp(-x_i \sum_{F_j \in \delta(i)} \mu_{F \rightarrow i}^t).
\]

This parallel algorithm has strong convergence guarantees that depend on the topology of the factor graph, specifically on the maximal connectivity \( \Delta_V = \max_{i \in V} |\delta(i)| \). Based on the analysis of Nishihara, Jegelka & Jordan [146] on block coordinate descent for a similar dual (assuming that all factors depend on all variables, as considered by Jegelka, Bach & Sra [51]), we extend their analysis to obtain a linear convergence rate for our message passing scheme.

**Theorem 6.4** (Extension of [146]). If the graph is \( \Delta_V \)-regular, s.t. every variable appears in exactly \( \Delta_V \) factors, then the message passing algorithm converges linearly with rate \((1 - \frac{1}{|V|\Delta_V})^2\). More specifically

\[
\|q^t - q^*\| \leq 2\|q^0 - q^*\|_\infty \sqrt{\Delta_V E(1 - \frac{1}{|V|^2\Delta_V^2})^t},
\]

where \( q^* \) is the optimal point, \( q^0 \) is the initial point and \( E \) is the number of edges in the factor graph.

The proof of the above theorem is provided in Appendix A.2.
6.4 EXPERIMENTS

In our experiments we consider probabilistic models with an energy function of the form

\[
F(A) = \sum_{i \in A} u_i + \sum_{j=1}^{L} \left( \max_{i \in A} w_{i,j} - \sum_{i \in A} w_{i,j} \right) - \sum_{j=1}^{K} \left( \max_{i \in A} w'_{i,j} - \sum_{i \in A} w'_{i,j} \right), \tag{6.18}
\]

We would like to point out that even though \( \sum_{j=1}^{K} G_j(A) \) is not \( M^{\sharp} \)-concave, each summand \( G_j \) is (Example 2.23), which we will exploit when designing our inference algorithms. The model parameters are the unary weights \( u_i \in \mathbb{R} \), and the weights \( w_{i,j} \geq 0 \) and \( w'_{i,j} \geq 0 \) that define the two facility location potentials. We consider the following four models.

- **Fully factorized model.** We obtain this model by setting \( L = K = 0 \), i.e., \( F(A) = u(A) \). This model serves as a baseline and it models no dependencies among the items.

- **FLID [120].** This is the model from Example 5.5, which we obtain if we fix \( L = 0 \), i.e., it has no log-supermodular component. Tschiatschek, Djolonga & Krause [120] have shown that it performs on par with DPPs on a product recommendation task.

- **Facility location complements model (FLIC).** This model is instantiated from (6.18) by setting \( K = 0 \), it defines a log-supermodular probability distribution. Remember that we have seen such a model also in Section 5.6.1. Because of the FKG inequality (Theorem 5.1), this model can only model positive correlations — in other words, we can only model the complementary properties of the items.

- **Facility location diversity and complements model (FLDC).** This model is instantiated from (6.18) by letting both \( L \) and \( K \) be positive. Hence it is a mixed model that generalizes both FLIC and FLID, and should be able to represent both complements and substitutes.

**DATASET** We use the Amazon baby registry dataset collected by Gillenwater et al. [147] for evaluating our methods. This dataset is a standard dataset for benchmarking diversity models and consists of baby registries collected from Amazon. We obtain the post-processed data from Tschiatschek, Djolonga & Krause [120], who split these registries into sub-registries.
according to 13 different product categories, such as safety and carseats. Each such category contains 32 to 100 different items, and there are $\approx 5,000$ to $\approx 13,300$ sub-registries per category.

**Learning** We learned the models described in the previous section using the training data of the different categories. In case of the modular model, the parameters $u$ were set according to the item frequencies in the training data. FLID, FLIC and FLDC were learned using noise contrastive estimation (NCE) [120, 148]. We used stochastic gradient descent for optimizing the NCE objective, created 200,000 noise samples from the modular model and made 100 passes through the data and noise samples. The code was adapted from the one used in [120].

**Evaluation** To evaluate the quality of the marginals from the proposed inference techniques, we repeat the following experiment. We first choose a random basket $B$ from the set of baskets that has at least 2 items in it. Then, we randomly choose $C \subseteq B$ and $D \subseteq V - B$ that satisfy $|C| \in [1, |A|)$ and $|D| = [\lfloor V - B \rfloor / 2]$, and perform inference in the model $P(A \mid A \in [C, D])$ by computing an approximation $Q(A \mid A \in [C, D])$. This task has been designed to capture the following scenario. The user has added some items $C$ to their basket and also shown disinterest in the items in $V - D$. The goal is to recommend them new items from those in the interval $[C, D]$. To evaluate the approximate marginals, we compute the AUC of $B \cap [C, D]$ under the approximate marginals from $Q$. We have found that the inclusive divergence gave better results for FLID, while the exclusive divergence for FLIC and FLDC. For approximate inference in FLIC and FLDC we did not split the log-submodular terms and bounded them as a whole. The results are summarized in Table 6.1. We can observe that FLDC has the highest AUC for all datasets with at most 62 items. For larger datasets, FLDC and FLIC have roughly the same performance and are superior to FLID and the modular model.

6.5 Conclusion

Despite being hard, the combinatorial problems arising from the infinite divergences often possess some structure that we can exploit. In this chapter, we have shown that if the energy function factorizes into simpler terms, we can leverage the dual decomposition framework to obtain relaxed bounds that still provably bound the partition function. Furthermore, we have seen
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Modular</th>
<th>FLID</th>
<th>FLIC</th>
<th>FLDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>safety</td>
<td>0.731304</td>
<td>0.756981</td>
<td>0.731269</td>
<td><strong>0.761168</strong></td>
</tr>
<tr>
<td>furniture</td>
<td>0.701840</td>
<td>0.739646</td>
<td>0.702100</td>
<td><strong>0.759979</strong></td>
</tr>
<tr>
<td>carseats</td>
<td>0.717463</td>
<td>0.770085</td>
<td>0.735472</td>
<td><strong>0.781642</strong></td>
</tr>
<tr>
<td>strollers</td>
<td>0.727055</td>
<td>0.794655</td>
<td>0.827800</td>
<td><strong>0.849767</strong></td>
</tr>
<tr>
<td>health</td>
<td>0.750271</td>
<td>0.754185</td>
<td>0.756873</td>
<td><strong>0.758586</strong></td>
</tr>
<tr>
<td>bath</td>
<td>0.692423</td>
<td>0.705051</td>
<td>0.730443</td>
<td><strong>0.732407</strong></td>
</tr>
<tr>
<td>media</td>
<td>0.666509</td>
<td>0.667848</td>
<td>0.758552</td>
<td><strong>0.780634</strong></td>
</tr>
<tr>
<td>toys</td>
<td>0.724763</td>
<td>0.729089</td>
<td>0.765474</td>
<td><strong>0.777729</strong></td>
</tr>
<tr>
<td>bedding</td>
<td>0.741786</td>
<td>0.744443</td>
<td><strong>0.771159</strong></td>
<td>0.764595</td>
</tr>
<tr>
<td>apparel</td>
<td>0.700694</td>
<td>0.696010</td>
<td>0.778067</td>
<td><strong>0.779665</strong></td>
</tr>
<tr>
<td>diaper</td>
<td>0.685051</td>
<td>0.700543</td>
<td><strong>0.787457</strong></td>
<td>0.787274</td>
</tr>
<tr>
<td>gear</td>
<td>0.687686</td>
<td>0.688116</td>
<td>0.687501</td>
<td><strong>0.688885</strong></td>
</tr>
<tr>
<td>feeding</td>
<td>0.686240</td>
<td>0.686845</td>
<td><strong>0.744043</strong></td>
<td>0.739921</td>
</tr>
</tbody>
</table>

| Average     | 0.708698 | 0.725653 | 0.752016 | **0.766327** |

Table 6.1: The AUC for the four considered models on the product recommendation task. The best result for each dataset is printed in boldface. FLDC has the highest dataset for smaller datasets (≤ 62 items), while FLIC and FLDC have similar AUC in the remaining categories.
how applying coordinate ascent to these relaxed bounds recovers the well-known expectation propagation algorithm. Moreover, thanks to the convex nature of inclusive divergence, we have also shown that in this case the EP algorithm is provably convergent at a rate of $O(1/k)$. We have also introduced mixed submodular models, that are powerful enough to capture any point process, and can be easily fit in the above framework. Namely, we have shown how to leverage the inference methods developed in the previous chapters as subroutines to perform inference in a much richer model family. For purely log-supermodular models, these relaxations are tight, and we have shown how one can develop efficient message-passing algorithms that linearly converge. Finally, we have shown the usefulness of our approach on a product recommendation task.
The material in this chapter has been already published in the following conference proceeding.


7.1 MOTIVATION

The class of models that we discuss next has been introduced in the computer vision literature as a remedy to some of the problems arising from using regular MRFs for semantic image segmentation. Namely, despite their widespread use, the solutions obtained using graph cuts can exhibit some undesirable behaviour. Specifically, they suffer from the so-called "shrinking bias" problem — they prefer segmentations that have short boundaries, i.e., they tend to penalize too strongly the number of neighbouring pixels that have a different label. They can thus have a hard time handling elongated structures, such as tree branches.

As a solution to this problem, Jegelka & Bilmes [149] have suggested a novel method that leverages submodularity in a fundamentally different way. Informally stated, they model the edges, rather than the pixels themselves, to be attractive. They start by clustering the edges, defined as connecting neighbouring pixels on a grid graph, based on their characteristics, e.g., using the color gradient across the endpoints. The underlying assumption is that the object boundaries will be captured by edges that partake in only a few of these clusters. Or stated differently, the cost of the disagreement across some edge should decrease as a function of the number of disagreements that already exist in the cluster where the edge belongs — i.e., the edges should exhibit a diminishing returns property.

To better illustrate the model, consider the configuration in Figure 7.1, where each variable takes on one of two labels indicated by its color, and the edges can belong to one of two clusters (red and blue). The edges that connect vertices with different labels, also said to be cut, are drawn with dashed lines. Note that while three out of four red edges are cut, only a single blue edge is cut. Thus, this configuration should be significantly less
preferred by the model than the alternative where we change the label of $X_{3,1}$. For example, a cost function that captures this above behaviour is

$$\sqrt{\text{#red edges cut}} + \sqrt{\text{#blue edges cut}}.$$

Previous work has only addressed the task of computing the most likely configuration, also called the cooperative cut problem, for a subclass of the models considered here [149, 150]. Instead, we will tackle the problem of approximate inference, which is challenging not just because the costs are defined in terms of the edges, but also because for realistic images the models will have an extremely high order. There has been also work on multi-label log-supermodular models by Zhang, Djolonga & Krause [13], which are a subclass of the class of models considered here. Moreover, their optimization objective is a special case of the upper bound that we derive in Section 7.4.

In what follows, we will show how to use the inclusive and exclusive infinite divergences to obtain both (i) lower and upper bounds on the partition function, and (ii) approximate variational distributions. To design the approximation schemes we will make use of both the polyhedral aspects of submodular functions, and the variational methods for inference in MRFs. We will design a fully convex upper bound that combines optimization over base polyhedra with the tree-reweighted belief propagation or SDP relaxations from Section 3.3.1. The lower bound will be constructed by alternatingly solving submodular minimization, and running the classical mean-field or belief propagation algorithms.

7.2 INTRODUCTION

We address models, which we call cooperative graphical models, that are specified by an undirected graph $G = (V, E)$. The nodes will be w.l.o.g. be taken to be $V = \{1, 2, \ldots, n\}$, and we will also assume that the edges are labelled with numbers from 1 to $m = |E|$. To each node $i \in V$ we associate a random variable $X_i$ that takes values in $\mathcal{L} = \{0, 1, \ldots, k - 1\}$. Then, the models that we consider have the form

$$P(x) = \frac{1}{Z} \exp \left( - \left( \sum_{i=1}^{n} \lambda^\top \iota(x) + F(\Delta(x)) \right) \right) \nu(x), \quad (7.1)$$

where $\lambda \in \mathbb{R}^{nk}$ holds the unary potentials, and the other functions are defined as follows.
Figure 7.1: Example cooperative model. Edge colors indicate the edge cluster. Dotted edges are cut under the current assignment.

- \( \iota: \mathcal{L}^n \to \{0,1\}^{nk} \) collects the unary indicators, i.e., it has elements
  \[
  [\iota(x)]_{i,a} = \mathbb{I}[x_i = a].
  \]

- \( \Delta: \{0,1\}^n \to \{0,1\}^m \) is the disagreement variable, defined as
  \[
  [\Delta(x)]_{ij} = \mathbb{I}[x_i \neq x_j].
  \] (7.2)

In other words, given a configuration \( x \in \mathcal{L}^n \), \( \Delta(x) \) will evaluate to a one on some edge if its endpoints have different values, and zero otherwise.

- \( \nu: \mathcal{L}^n \to \{0,1\} \), called the base measure, allows us to encode constraints, e.g., conditioning on some variables.

If the function \( F \) is modular, we recover the Potts model from Example 3.2. With the help of auxiliary variables, we can also represent the multi-label log-supermodular models from [13], who consider models of the form
\[
P(x) \propto \exp(-G(\iota(x)))
\]
for some submodular function \( G \). Namely, we can introduce \( k \) new variables \( Z_0, Z_1, \ldots, Z_{k-1} \) and fix their values using the base measure
\[
\nu(x, z) = \prod_{j=0}^{k-1} \mathbb{I}[z_i = i].
\]
We can easily see that the disagreement variable on the edges is equal to \( \iota(x) \) using a natural bijection, which completes the argument.

Probabilistic inference in our model class (7.1) is very challenging, since we make no factorization assumption about \( F \). One approach would be to encode \( P(x) \) as an exponential family model via a new variable \( z \in \{0,1\}^m \) and constraints
\[
\nu(x, z) = \mathbb{I}[\Delta(x) = z],
\]
but this in general requires computing exponential-sized sufficient statistics from \( z \). In contrast, we make one additional key assumption that will enable the development of efficiently computable variational lower and upper bounds. We will henceforth assume that \( F: \{0,1\}^m \to \mathbb{R} \) is submodular. In the special case that \( F \) is also
monotone, we recover the model class analyzed for optimization by Jegelka & Bilmes [149]. In our introductory example $F$ is submodular as it is a sum of concave-of-cardinality functions.

Note that even if $F$ is modular, the energy is not necessarily submodular or supermodular — this can be easily seen if $F(\{e\})$ for $e \in E$ evaluates to both positive and negative values. Hence, we can not directly apply the methods we have developed in the previous chapter to this model class.

**Related work** Computing the mode of (7.1) for binary models has been analyzed for the case when both the pairwise interactions $\theta_{i,j}$ are submodular and $f$ is monotone [35]. While the general problem is NP-hard, it can be solved if $F$ is a sum of threshold potentials [150] in time complexity exponential in the number of potentials. Zhang, Djolonga & Krause [13] have analyzed the problem of inference in multi-class log-supermodular models, which can be seen as a special case of the upper bound we will derive.

### 7.3 Inference in Cooperative Graphical Models

As already mentioned in the introduction, to perform inference we will use the infinite Rényi divergences. The approximation family $Q$ will consist of all Potts models (Example 3.2) defined over the same graph as the cooperative graphical model. Specifically, we will assume that the distributions in $Q$ take the form

$$Q(x) = \exp(-\lambda^T i(x) - \Delta(x)^T q) u(x) / Z(q),$$

for some vector $q \in \mathbb{R}^m$. In other words, we keep the same unary potentials as $\mathbb{P}$, but let the parameters $q \in \mathbb{R}^m$ defining the pairwise interactions vary. Another way to look at the above approximation, is that we "linearize" the function $F(\Delta(x))$ to $\Delta(x)^T q$, and moreover, we will optimize over the linearization to find the one that minimizes a distributional divergence. This approximate distribution is an MRF over the graph $G$ that has tied parameters. Specifically, the parameters $\theta \in \mathbb{R}^d$ of the MRF are defined as

$$\theta_{i,a} = -\lambda_{i,a}, \text{ and}$$

$$\theta_{i,j,a,b} = -[a \neq b] q_{\{i,j\}}.$$

Namely, let us define for each $q \in \mathbb{R}^m$ the parameter vector

$$\theta(q) = (-\lambda, -P^T q),$$

(7.4)
where $P \in \{0,1\}^{|E| \times d}$ extracts the disagreements and has for each row \{i,j\} non-zero entries only in positions corresponding to $\phi_{i,j,a,b}$ for $a \neq b$. Then, we can equivalently write $Q$ in the standard MRF form

$$Q(x) = \exp(\theta(q)^\top \phi(x) - A(\theta(q))),$$

where $\phi$ is the over-complete sufficient statistic of the MRF.

Because inference algorithms are typically defined for MRFs in their general form, we will throughout this chapter use $M$ for the marginal polytope of the MRF over $G$ with base measure $\nu$, rather than for the special case (7.3). For any $\mu \in M$ we will define by $\mu \cdot$ the subvector consisting of all unary marginals, and we will collect all pairwise marginals in $\mu \cdot \cdot$. For example, we have that

$$\theta(q)^\top \mu = -\lambda^\top \mu - q^\top P \mu \cdot \cdot.$$

In what follows we will repeatedly use the gradient of $A(\theta(q))$ with respect to the free parameters $q$. Using the chain rule, we can easily compute it as

$$\nabla_q A(-\theta(q)) = \nabla_q A((-\lambda, -P^\top q)) = -P \mu \cdot \cdot.$$ (7.5)

If we have a closer look at the elements of the gradient vector, we see that they are equal to negative probabilities, specifically

$$[P \tau \cdot \cdot]_{\{i,j\}} = \sum_{a \in \mathcal{L}} \sum_{b \in \mathcal{L}} c_{i,j,a,b} = \mathbb{E}_{x \sim Q}[X_i \neq X_j] \in [0,1]^m.$$ (7.6)

Hence, they correspond to negative disagreement probabilities, i.e., we assign to each edge the negative of the probability that its endpoints take on different labels.

### 7.4 Convex Upper Bounds

Remember that the resulting upper bound (4.6) obtained by minimizing the inclusive divergence $D_\infty(P \parallel Q)$ has the form

$$\log Z \leq \inf_{q \in \mathbb{R}^m} A(-\theta(q)) + \sup_{x \in \mathcal{X}} \theta(q)^\top \phi(x) - \lambda^\top t(x) - F(\Delta(x))$$ (7.7)

$$= \inf_{q \in \mathbb{R}^m} A(-\theta(q)) + \sup_{x \in \mathcal{X}} q^\top \Delta(x) - F(\Delta(x)).$$ (7.8)
Unfortunately, the inner maximization problem is exactly the cooperative cut problem, which we cannot solve exactly, so we further relax the above inequality to
\[
\inf_{\mathbf{q} \in \mathbb{R}^d} A(-\theta(\mathbf{q})) + \sup_{\mathbf{y} \in \{0,1\}^m} \mathbf{q}^\top \mathbf{y} - F(\mathbf{y}).
\]
Because the derivatives are inside the unit cube due to (7.6), we can apply Lemma 5.7 and reduce the problem to
\[
\inf_{\mathbf{q} \in B(F)} A(-\theta(\mathbf{q})) = \inf_{\mathbf{q} \in B(F)} \sup_{\mu \in \mathcal{M}} \theta(\mathbf{q})^\top \mu - A^*(\mu).
\]
where we have also used the variational representation (3.12) of the partition function \( A \).

Depending on the graph, we might not be able to optimize the inner problem, so we will approximate both the marginal polytope \( \mathcal{M} \) and entropy \(-A^*\). From all the approximations we have seen in Section 3.3.1, we want to choose a pair \((\overline{\mathcal{M}}, \overline{\mathcal{H}})\) that will not violate the bound on \( \log Z \), which means that we need an approximation that yields an upper bound on \( A(-\theta(\mathbf{q})) \). The methods from Section 3.3.1 that resulted in an upper bound combined an outer approximation \( \mathcal{L} \supseteq \mathcal{M} \) with a function \( \overline{\mathcal{H}} \) that globally dominates the true entropy, in which case the bound becomes
\[
\log Z \leq \inf_{\mathbf{q} \in B(F)} A(-\theta(\mathbf{q})) = \inf_{\mathbf{q} \in B(F)} \sup_{\tau \in \mathcal{L}} -\theta(\mathbf{q})^\top \tau + \overline{\mathcal{H}}(\tau) \tag{7.9}
\]
\[
= \inf_{\mathbf{q} \in B(F)} \sup_{\tau \in \mathcal{L}} -\lambda^\top \tau - \mathbf{q}^\top P\tau_{\cdot \cdot} + \overline{\mathcal{H}}(\tau). \tag{7.10}
\]
Because the inner problem is linear in \( \mathbf{q} \), this is a convex optimization problem over the base polytope. To obtain the gradient with respect to \( \mathbf{q} \) (equal to the negative disagreement probabilities), we have to solve the inner problem. Namely, this subproblem corresponds to performing variational inference in a pairwise model, e.g. via TRWBP or the SDP method from Section 3.3.1.

We would like to point out that the dual of the above problem is closely related to the problems considered by Vilnis et al. [151]. Namely, if we swap the inf and sup of the of the above problem (valid due to Theorem A.3), we obtain
\[
\sup_{\tau \in \mathcal{L}} -\lambda^\top \tau + \overline{\mathcal{H}}(\tau) - f_L(P\tau_{\cdot \cdot}). \tag{7.11}
\]
which is a special case of the problem considered in [151]. Specifically, in their terminology, the Lovász extension applied to the disagreement probabilities acts as a non-smooth non-local energy function. They also make a connection to the posterior regularization framework of Ganchev, Gillenwater & Taskar [152] — namely, we can see the above approach as performing inference in the MRF, but being regularized with the Lovász extension applied to the disagreement probabilities.

### 7.4.1 Algorithms

Let us now discuss two different approaches for solving problem (7.9). Their convergence rates depend on smoothness of the objective in problem (7.9), i.e., on the Lipschitz continuity of its gradients. Informally, the inferred disagreement probabilities should not change by more than $L\|q - q'\|$ if change the parameters from $q$ to $q'$ for some $L \geq 0$. We will first analyze the algorithms in more detail, before we state the conditions that will guarantee the smoothness of problem (7.9) in the next section.

**Frank-Wolfe** Given that we can efficiently solve linear programs over $\mathcal{B}(F)$, the Frank-Wolfe [47] algorithm explained in Section 2.6.1.1 is a natural candidate for solving the problem. For completeness, we present the algorithm when applied to this specific problem in Figure 7.2. From Theorem 2.17 we know that the method has a convergence rate of $O(L/t)$ [48], where $L$ is the assumed smoothness parameter. When taking a step towards $s$, the weight of edge $e_i$ is changed by $s_{e_i} = F(\{e_1, e_2, \ldots, e_i\}) - F(\{e_1, e_2, \ldots, e_{i-1}\})$. Due to the submodularity of $F$, an edge will obtain a higher weight if it appears earlier in the order determined by the disagreement probabilities. Hence, in every iteration, the algorithm will re-adjust the pairwise potentials of $Q$, by encouraging the variables to agree more as a function of their (approximate) disagreement probability.

**Projected Gradient Descent (PGD)** Since we know how to project onto $\mathcal{B}(F)$ using the minimum norm problem (Section 2.6.1), and practically so for many submodular functions $F$, we can alternatively use projected gradient or subgradient descent (PGD). Without smoothness, PGD converges at a rate of $O(1/\sqrt{t})$. If the objective is smooth, we can use an accelerated methods like FISTA [56], which has both a much better $O(L/t^2)$ rate and seems to also experimentally converge faster than many Frank-Wolfe variants.
procedure FW-Inference\( (F, \theta) \)
1. \( q \leftarrow \text{Linear-Oracle}\( (F, 0) \) \)
2. for \( t = 0, 1, \ldots, \max\_\text{steps} \) do
3. \( \hat{A}, \tau \leftarrow \text{VAR-Inference}\( (\theta(q)) \) \)
4. \( s \leftarrow \text{Linear-Oracle}\( (F, \tau) \) \)
5. \( \gamma \leftarrow \text{Compute-Step-Size}\( (q, s) \) \)
6. \( q \leftarrow (1 - \gamma)q + \gamma s \)
7. end for
8. return \( \tau, \hat{A} \)
9. end procedure

Figure 7.2: Inference with Frank-Wolfe, assuming that VAR-Inference guarantees an upper bound.

7.4.2 Convergence

The convergence rates of both algorithms depend on the smoothness parameter \( L \) of the objective. Because we represent the function we optimize as a conjugate of the approximate entropy, it will turn out that we need this approximation to be strongly convex due to a classical duality result in convex optimization.

Theorem 7.1. Problem (7.9) is \( k^2 \sigma \)-smooth over \( B(F) \) if the entropy surrogate \( -H \) is \( \frac{1}{\sigma} \)-strongly convex over \( \overline{M} \).

Proof. Remember that we are optimizing the function

\[
g(q) = \overline{A}(-\theta(q)) = \sup_{\tau \in \overline{M}} -\theta(q)^T \tau + \overline{H}(\tau)
\]

over \( q \in B(F) \). Moreover, remember that the gradient of \( g(q) \) is given as \(-P\tau^{*}\), where \( \tau^{*} \) maximizes the inner problem. Because \( \overline{A} \) is the conjugate of \( -\overline{H} \), it is thus \( \sigma \)-smooth (see e.g. Kakade, Shalev-Shwartz & Tewari [153]). Specifically, this means that for any two parameters \( \theta_1 \in \mathbb{R}^d \) and \( \theta_2 \in \mathbb{R}^d \) with optimal pseudomarginals \( \tau_1 \) and \( \tau_2 \) it hold that

\[
\|\tau_1 - \tau_2\| \leq \sigma \|\theta_1 - \theta_2\|.
\]
But, then for \( \theta_1 = \theta(q_1) \) and \( \theta_2 = \theta(q_2) \) for some \( q_1, q_2 \in \mathbb{R}^m \) with optima \( \tau_1 \) and \( \tau_2 \) respectively, we have that

\[
\|P\tau_1 - P\tau_2\| \leq \|P\| \|\tau_1 - \tau_2\| \\
\leq \sigma \|P\| \|\theta(q_1) - \theta(q_2)\| \\
= \sigma \|P\| \|P^T\| \|q_1 - q_2\| \\
= \|P\|^2 \sigma \|q_1 - q_2\|.
\]

Moreover, from Hölder’s inequality it follows that

\[
\|P\|^2 \leq \|P\|_1 \|P\|_\infty
\]

where \( \|P\|_\infty \) is the maximum row sum, which is \( k(k - 1) \), and \( \|P\|_1 \) is the maximum column sum, which is equal to 1. Combining these two observations completes the proof. \( \Box \)

The benefits of using strongly convex entropy approximations has been discussed in the variational inference literature first by Wainwright [154], who has shown that both the TRWBP and SDP relaxations use strongly convex entropies. London, Huang & Getoor [155] provide an even sharper bound for the tree reweighted entropy, and show how one can strongly convexify any weighted entropy by solving a QP over the weights \( \rho \).

In practice, because the inner problem is typically solved using an iterative algorithm and because the problem is smooth, we obtain speedups by warm-starting the solver with the solution at the previous iterate. We can moreover easily obtain duality certificates using the results of Jaggi [48].

### 7.4.3 Joint optimization

When using weighted entropy approximations in the upper upper bound (see (3.21)), it makes sense to optimize over both the linearization \( g \) and the weights \( \rho \) jointly. Specifically, let \( T \) contain a set of weights \( \rho \) that yield an entropy approximation \( \overline{H} \) upper bounding \( H \). Then, if we expand \( \overline{H} \) in problem (7.9), we obtain

\[
\inf_{\rho \in T} \inf_{q \in B(F)} \sup_{\tau \in L} -\tau^T \theta(q) + \sum_{i \in V} \rho_i H_i(\tau_i) + \sum_{\{i,j\} \in E} \rho_{i,j} H_{i,j}(\tau_{i,j}). \tag{7.12}
\]

Note that inside the supremum, both \( g \) and \( \rho \) appear only linearly, and there is no summand that has terms from both of them. Thus, the problem is convex in \( (q, \rho) \), and we can optimize jointly over both variables. As a
final remark, if we already perform inference in a pairwise model and repeatedly tighten the approximation by optimizing over \( \rho \) via Frank-Wolfe (as suggested in [82]), then the complexity per iteration remains the same even if we use the submodular term \( F \).

### 7.5 Submodular Lower Bounds

Remember that the lower bound (4.5) obtained by minimizing \( D_\infty(Q \| P) \) is given as

\[
\log Z \geq \sup_{q \in \mathbb{R}^m} A(-\theta(q)) + \inf_{x \in X} q^T \Delta(x) - F(\Delta(x)),
\]

which is not violated if we take the infimum over a larger set

\[
\log Z \geq \sup_{q \in \mathbb{R}^m} A(-\theta(q)) + \inf_{y \in \{0,1\}^m} \underbrace{q^T y - F(y)}_{f^\star(q)}.
\]

Using exactly the same argument as in Theorem 4.2 and the variational form ((3.12)) of the log-partition function, we can re-write the above as

\[
\sup_{(q,c) \in U(F)} \sup_{\mu \in M} q^T \theta(\mu) - A^*(\mu) - c. \tag{7.13}
\]

Similarly to the previous section, we can in general neither describe \( M \) nor evaluate \( A^* \), so we have to resort to approximate techniques. In contrast to the upper bounds, we need approximation pairs \( (\overline{M}, \overline{H}) \) of \( (M, -A^*) \) that will yield a lower bound on \( A(-\theta(q)) \). Moreover, because we can not directly work with \( U(F) \), we suggest to instead use the upper bar polyhedron (Definition 2.21) \( \overline{U}(F) \), which results in the following optimization problem

\[
\sup_{(q,c) \in \overline{U}(F)} \sup_{\tau \in \overline{M}} q^T P \tau_{\cdot,\cdot} + a^\top \tau_{\cdot,\cdot} + \overline{H}(\tau) - c. \tag{7.14}
\]

We propose to optimize this lower bound in a block-coordinate-wise manner: first with respect to the supergradient \( (q,c) \in \overline{U}(F) \), and then with respect to the pseudo-marginals \( \tau \) (which amounts to approximate inference in an MRF). Note that the problem is linear in \( (q,c) \), so that we can leverage Lemma 5.4 to solve the minimization over the super-gradients, which we formalize in the following result.
Lemma 7.1. For a fixed $\tau$, the supremum of (7.14) over $(q,c) \in \overline{U}(F)$ is attained at the bar super-gradient $(\bar{s}^A, F(X) - \bar{s}^A(A))$, where $A$ is given as the optimum of the following submodular minimization problem

$$\min_{A \subseteq E} F(A) + m(A),$$

where $m(\cdot)$ is a modular function with elements

$$m(\{e\}) = q_e F(e | V - \{e\}) - q_e F(e | V - \{e\}) - F(\{e\}).$$

In contrast to computing the MAP, the above problem has no constraints and can be easily solved using existing algorithms.

As the approximation algorithm for the inner model, we can always use mean-field, which uses the exact entropy $-A^*$, but uses an inner bound of the marginal polytope $\overline{M} \subseteq M$ containing models for which the entropy is exact. Moreover, Ruozzi [79] has shown that belief-propagation yields a lower bound on the partition function for binary pairwise log-supermodular models. Hence, if the approximate MRF with parameters $\theta(q)$ is log-supermodular, then we can use belief-propagation to solve the inner problem. Remember that the pairwise potentials in the MRF are equal to

$$\theta_{i,j;a,b} = -q_{\{i,j\}}[a \neq b],$$

which will be log-supermodular if $q_{\{i,j\}} \geq 0$. A sufficient condition for log-supermodularity is the function $F$ to be monotone, because in this case all gains will be non-negative, so that each $(q,c) \in \overline{U}(F)$ will satisfy $q \geq 0$ (see Table 2.1). Moreover, in this setting both the mean-field and belief propagation objectives (i.e. computing $\tau$) can be cast as an instance of continuous submodular minimization (see e.g. [110]), which means that they can be solved to arbitrary precision in polynomial time. Unfortunately, problem (7.14) will not be jointly submodular, so we still need to use the block-coordinate ascent method we just outlined.

7.6 APPROXIMATE INFERENCE VIA MAP PERTURBATIONS

For binary models with submodular pairwise potentials and monotone $F$ we can (approximately) solve the MAP problem using the techniques in [35, 150]. Hence, this opens as an alternative approach the Perturb-and-MAP method of Papandreou & Yuille [86], which we have covered in Section 3.3.3. We will namely utilize first-order perturbations, as then the perturbed energies can be still be solved using the aforementioned methods.
7.7 Experiments

Synthetic experiments Our first set of experiments uses complete graphs on \( n \) variables. Throughout all experiments we have sampled the unary potentials as \( \lambda_{i,a} \sim \text{Uniform}(-\alpha, \alpha) \). The edges \( E \) were randomly split into five disjoint buckets \( E_1, E_2, \ldots, E_5 \), and we used

\[
F(\Delta(x)) = \sum_{j=1}^{5} h_j(\Delta(x)_{E_j}) + v^\top \Delta(x).
\]

where \( v \in \mathbb{R}^m \) holds the pairwise Potts potentials, \( \Delta(x)_{E_i} \) are the coordinates of \( \Delta(x) \) corresponding to that edge group, and the functions \( \{h_j\} \) will be defined below. Note that the pairwise potential \( v_{\{i,j\}}[\Delta(x)]_{\{i,j\}} \) is log-supermodular if \( v_{\{i,j\}} \geq 0 \). To perform inference in the linearized pairwise models, which we need as a subroutine, we used: \( \text{trwbp} \) (tree-reweighted belief propagation [82]), \( \text{jtree}^+ \) (exact inference using the junction tree algorithm, upper bound), \( \text{jtree}^- \) (exact inference using the junction tree algorithm, lower bound), \( \text{sdp} \) (the semi-definite relaxation from [83]), \( \text{mf} \) (mean-field), \( \text{bp} \) (belief propagation [78]), \( \text{pmap} \) (perturb-and-MAP with approximate MAP [86]) and \( \text{epmap} \) (perturb-and-MAP with exact MAP [86]). We used \( \text{libDAI} [156] \) and implemented \( \text{sdp} \) using \( \text{cvxpy} [157] \) and \( \text{SCS} [158] \). As a maxflow solver we used [159]. Errors bars denote three standard errors.

Figure 7.3 shows the results for

\[
h_i(y_{E_i}) = w_i \sqrt{\sum_{e \in E_i} y_e / \sqrt{|E_i|}},
\]

with weights \( w_i \sim \text{Uniform}(0, \beta) \). In panel (c) we use mixed (attractive and repulsive) pairwise potentials, chosen as \( v_{\{i,j\}} \sim \text{Uniform}(-\beta, \beta) \). First, the results imply that the methods optimizing the fully convex upper bound yield very good marginal probabilities over a large set of parameter configurations. The estimate of the log-partition function from \( \text{trwbp} \) is also very good, while \( \text{sdp} \) is much worse, which we believe can be attributed to the very loose entropy bound used in the relaxation. The lower bounds (\( \text{bp} \) and \( \text{mf} \)) work well for settings when the pairwise strength \( \beta \) is small compared to the unary strength \( \alpha \). Otherwise, both the bound and the marginals become worse, while \( \text{jtree}^- \) still performs very well. This could be explained by the hardness of the pairwise models obtained after linearizing \( F \). Finally, \( \text{pmap} \) (when applicable) seems very promising for small \( \beta \).
To better understand the regimes when one should use \texttt{trwbp} or \texttt{pmap}, we compare their marginal errors in Figure 7.5. We see that for most parameter configurations, \texttt{trwbp} performs better, and significantly so when the edge interactions are strong.

Finally, we evaluate the effects of the approximate MAP solver for \texttt{pmap} in Figure 7.4. To be able to solve the MAP problem exactly (see [150]), we used $h(y_{E_j}) = \max \{\sum_{e \in E_j} y_e v_e, \sum_{e \in E_j} v_e/2\}$, where $v_e \sim \text{Uniform}(0, \beta)$. As evident from the figure, the gains from the exact solver seem minimal, and it seems that solving the MAP problem approximately does not strongly affect the results.

**An Example from Computer Vision** To demonstrate the scalability of our method and obtain a better qualitative understanding of the resulting marginals, we ran \texttt{trwbp} and \texttt{pmap} on a real world image segmentation task. We use the same setting, data and models as [35], as implemented in the \texttt{pycoop} package. Because \texttt{libDAI} was too slow, we wrote our own TR-WBP implementation. Figure 7.6 shows the results for two specific images (size 305 $\times$ 398 and 214 $\times$ 320). The example in the first row is particularly difficult for pairwise models, but the rich higher-order model has no problem capturing the details even in the challenging shaded regions of the image. The second row shows results for two different model parameters. The second model uses a function $F$ that is closer to being linear, while the first one is more curved (see Table 7.1 for details). We observe that \texttt{trwbp} requires lower temperature parameters (i.e. relatively larger potentials $\theta$ and $F$) than \texttt{pmap}, and that the bottleneck of the complete inference procedure is running the \texttt{trwbp} updates. In other words, the added complexity from our method is minimal and the runtime is dominated by the message passing updates of TRWBP. Hence, any algorithms that speed up TRWBP (e.g., by parallelization or better message scheduling) will result in a direct improvement on the proposed inference procedure.

7.8 Conclusion

We have designed inference techniques for a rich family of discrete multi-label models by carefully combining their submodular properties with classical variational inference techniques. We have shown that minimizing the inclusive divergence results in a fully convex problem that yields an upper bound on the log-partition function. Namely, it was proven that this prob-
lem reduces to the minimization of a convex function, computed using the TRWBP or SDP relaxations, over a base polytope. By minimizing the exclusive divergence, we designed an alternating minimization scheme that combines belief propagation or mean-field with unconstrained submodular minimization. We have numerically shown the efficacy of our schemes, and also applied them to a challenging image segmentation problem.
Figure 7.3: Results on several synthetic models. The methods that optimize the convex upper bound (trwbp, sdp) obtain very good marginals for a large set of parameter settings. Those maximizing the lower bound (bp, mf) fail when there is strong coupling between the edges. In the strong coupling regime the results of pmap also deteriorate, but not as strongly. In (c) bp, pmap, sdp are not applicable.

Figure 7.4: \( \alpha = 2, K_{15} \), model where epmap is applicable. Solving the MAP problem exactly only marginally improves over pmap. The other observations are similar to those in Fig. 7.3b.
**Figure 7.5:** error\textsubscript{pmap} - error\textsubscript{trwbp} on $K_{15}$. Missing entries were not significant at the 0.05 level.

<table>
<thead>
<tr>
<th>Panel</th>
<th>$\theta$</th>
<th>$\lambda$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c)</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>(e)</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>(g)</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>(i)</td>
<td>0.001</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>(d)</td>
<td>0.01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(f)</td>
<td>0.01</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>(h)</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>(j)</td>
<td>0.1</td>
<td>1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

**Table 7.1:** The model of Jegelka & Bilmes [35] has two parameters: $\lambda$, which controls the strength of the unaries, and $\theta$, which controls the curvature of the function $f$. In addition to these parameters, we have another parameter $t$ that controls the inverse temperature of the model, i.e. the complete energy in (7.1) is multiplied by $t$. In the table above we provide all parameters settings for Figure 7.6.
Figure 7.6: Inferred marginals on an image segmentation task. The left column showcases an example that is particularly hard for pairwise models. In the right column we show the results for two different models (the cooperative function $F$ is more curved for model 1). The actual parameter values are given in Table 7.1.
LEARNING OF LOG-SUPERMODULAR MODELS

The material in this chapter has been already published in the following conference proceeding.


8.1 motivation

In the previous chapters we have seen how submodularity can be used to model useful phenomena, and how it can be furthermore leveraged in the design of efficient approximate inference schemes. However, we have always assumed that the model is given to us and fixed. For example, in the image segmentation experiment in Section 5.6.2, we have narrowed down the model class using our prior knowledge of the problem, but left a few free parameters that were selected using cross validation. This approach can be of course problematic in general — while we would like to encode our biases and prior knowledge, it can be in general very hard to reduce the model class to a handful of numbers that have to be tuned.

Hence, we would like to design a technique that given a model class encoding our prior knowledge, it automatically, and in a computationally efficient way, selects a distribution from this collection that is best suited to our task. In this chapter we will focus on the specific problem of learning log-supermodular models from data. Formally, given (i) a family of submodular functions $F_\theta$ parametrized by some $\theta \in \mathbb{R}^d$ that has to be estimated, and (ii) a collection of sets $A_1, A_2, \ldots, A_m \in 2\{1,2,\ldots,n\}$, we will propose a method how to fit the parameter $\theta$ as to best explain the data.

Perhaps the first strategy one would suggest is to learn the model in a maximum-likelihood manner, by approximating the partition function, and hence the likelihood, using the upper bound from the inclusive divergence. However, Shpakova & Bach [95, §4.1] have shown that the bound is unfortunately too loose and will result in a degenerate solution. We will indeed use the inclusive divergence for learning the model, but in a different fashion — we would like to learn models such that the best approximation $Q$ to $\mathbb{P}$, gives a high-likelihood to the data. This is perhaps the most intu-
itive quantity we should maximize if we use $Q$ as an approximation to $P$. The optimization problem is however very challenging — because we have to evaluate the likelihood of $Q$, which is itself defined only implicitly as a solution to a complex optimization problem. Nevertheless, we will see how this will be indeed possible thanks to the special structure of the solution of the inclusive divergence in the case of log-supermodular models.

Moreover, we will obtain an answer to the intriguing question of how to use submodular minimization in conjunction with deep learning models. Due to the close connection between the inference scheme and the min-norm problem, via our analysis we will enable the incorporation of modules that perform submodular optimization as layers in modern architectures. Specifically, we can design layers that accept as an input a vector parametrizing a submodular function, and output the (smoothed) minimum of the function. This will, perhaps surprisingly, have connections to several structured attention mechanisms that have been recently suggested in the natural language processing literature.

### 8.2 Introduction

To better illustrate our approach, let us consider the image segmentation example from Example 2.15. Remember that we were given as input an RGB representation $x \in \mathbb{R}^{n \times 3}$ of say an image captured by say a dashboard camera, and the goal was to identify the set of pixels $A \subseteq \{1, 2, \ldots, n\}$ that are occupied by pedestrians. The energy function that we have suggested contained a modular function $v(\cdot)$ whose value $v_i = v(\{i\}) \in \mathbb{R}$ for each site $i \in V$ indicated how likely it is that the pixel contains a pedestrian. Furthermore, to encourage spatial consistency we have added a graph-cut component, which has resulted in the following energy function

$$F_{w,v}(A) = \sum_{\{i,j\} \in E} w_{\{i,j\}} \mathbb{1}[|A \cap \{i,j\}| = 1] + \sum_{i \in A} v_i,$$

where $w_{\{i,j\}} \geq 0$ are the pairwise weights. Hence, the family of distributions are those that have the following form

$$P(A \mid w, v) = \exp(-F_{w,v}(A)) / Z(w, v).$$

Having fixed the model class, the goal is to learn the parameters $w$ and $v$ that specify the distribution as a function of $x$ — in other words, we want to learn the conditional distribution $P(A \mid x)$. To this end, we propose
to compute the parameters $w$ and $v$ using a predictor, such as a neural network, i.e., to set them using a mapping $\theta: x \rightarrow (w, v)$. We will assume that the function $\theta$ is itself parametrized, and we will tune these parameters to maximize our objective function. As we have already mentioned in the introduction, given data of the form $(x_1, A_1), (x_2, A_2), \ldots, (x_m, A_m)$ we can not directly optimize

$$\max_{\theta} \sum_{j=1}^{m} \log P(A_j \mid \theta(x_j)),$$

due to the intractability of the likelihood. Hence, we will instead strive to find a mapping $\theta$ so that the best fully factorized approximate distribution

$$Q(\cdot \mid x) = \arg\min_{Q \in \mathcal{Q}} D_\infty(P(\cdot \mid \theta(x)) \parallel Q)$$

gives high likelihood to the training data. In other words, we would like to optimize the following procedure in an end-to-end manner with respect to the parameters of $\theta$.

$$x_i \xrightarrow{\theta(x)} \theta_i \rightarrow P = \exp(-F_{\theta_i}(A)) / \mathcal{Z}(\theta_i)$$
$$\rightarrow Q = \arg\min_{Q \in \mathcal{Q}} D_\infty(P \parallel Q) \quad (8.3)$$
$$\rightarrow Q(A_i).$$

This is a complicated bi-level optimization task (as $Q$ implicitly depends on $\theta$) with an inner variational inference sub-problem. Because we would like to use a first-order optimization method, the main bottleneck is the computation of the partial derivatives of the log-likelihood under the closest fully factorized distribution with respect to the parameters of $\theta$. We approach this problem by leveraging the equivalence between the optimizer of the divergence and the minimum-norm point (Theorem 5.4). Namely, if $y$ is the min-norm point of $F_{w,v}$, then the closest fully factorized distribution $Q$ has marginals $\sigma(-y_i)$, where $\sigma(u) = 1/(1 + e^{-u})$ is the sigmoid function. More importantly, the marginals of $Q$ and the log-likelihood

$$\log Q(A_i \mid x_i) = \sum_{i \in A} \log(\sigma(-y_i)) - \sum_{i \notin A} \log(\sigma(y_i)) \quad (8.4)$$

are simple differentiable functions of the min-norm point. Hence, if we can differentiate the min-norm point, then we can also easily compute the derivative of the log-likelihood using (8.4) and the chain rule. We will thus devote our attention to the analysis of the mapping

$$x \rightarrow \theta(x) \rightarrow F_{\theta(x)} \rightarrow y = \arg\min_{y \in B(F_{\theta(x)})} \|y\|^2. \quad (8.5)$$
Our key technical contribution is the analysis of the Jacobian of this function, or in other words the sensitivity of the min-norm point as a function of the parametrization of the submodular function. For the graph-cut case above we will show how to compute $\partial y / \partial w$ and $\partial y / \partial v$.

**Related Work**  Sensitivity analysis of the set of optimal solutions has a long history in optimization [160]. The problem of argmin-differentiation of the specific case resulting from graph cuts (i.e., Example 2.21) has been considered in the computer vision literature, either by smoothing the objective [161], or by unrolling iterative methods [162]. The idea to train probabilistic models by evaluating them using the marginals produced by an approximate inference algorithm has been studied by Domke [163] for tree-reweighted belief propagation and mean field, and for continuous models by Tappen [164]. These methods either use the implicit function theorem, or unroll iterative optimization algorithms. The benefits of using an inconsistent estimator, which is what we do by optimizing eq. (8.3), at the benefit of using computationally tractable inference methods has been discussed by Wainwright [154]. Amos & Kolter [165] discuss how to efficiently argmin-differentiate quadratic programs by perturbing the KKT conditions, an idea that goes back to Boot [166]. We make an explicit connection to their work in Theorem 8.4. In Section 8.4 we harness the connection between the min-norm problem and isotonic regression, which has been exploited to obtain better duality certificates [20], and by Kumar & Bach [167] to design an active-set algorithm for the min-norm problem. Chakravarti [168] analyzes the sensitivity of the optimal isotonic regression point with respect to perturbations of the input, but does not discuss the directional derivatives of the problem. Recently, Bilmes & Bai [27] have used deep networks to parametrize submodular functions. Discrete optimization is also used in structured prediction [4, 169] for the computation of the loss function.

**Contributions**  We develop a very efficient approximate method (Section 8.4) for the computation of the Jacobian of the min-norm problem inspired by our analysis of the isotonic regression in Section 8.3, where we derive results that might be of independent interest. Even more importantly, from a practical perspective, this Jacobian has a very nice structure and we can multiply with it in linear time. This means that we can efficiently perform back-propagation if we use these layers in a modern deep architectures. In Section 8.5 we show how to compute directional deriva-
tives exactly in polynomial time, and give conditions under which our approximation is correct. This is also an interesting theoretical result as it quantifies the stability of the min-norm point with respect to the model parameters. Lastly, we use our results to learn log-supermodular models in Section 8.6.

8.3 argmin-Differentiation of Isotonic Regression

We will first analyze a simpler problem, i.e., that of isotonic regression, defined as

\[ y(x) = \arg\min_{y \in \mathcal{O}} \frac{1}{2} \|y - x\|^2, \]  

(8.6)

where \( \mathcal{O} = \{y \in \mathbb{R}^n \mid y_i \leq y_{i+1} \text{ for } i = 1, 2, \ldots, n - 1\} \). The connection to our problem will be made clear in Section 8.4, and it essentially follows from the fact that the Lovász extension is linear on \( \mathcal{O} \). In this section, we will be interested in computing the Jacobian \( \frac{\partial y}{\partial x} \), i.e., in understanding how the solution \( y \) changes with respect to the input \( x \). The function is well-defined because of the strict convexity of the objective and the non-empty convex feasible set. Moreover, it can be easily computed in \( O(n) \) time using the pool adjacent violators algorithm (PAVA) [170]. This is a well-studied problem in statistics, see e.g. [171]. To understand the behaviour of \( y(x) \), we will start by stating the optimality conditions of problem (8.6). To simplify the notation, for any \( A \subseteq V \) we will define \( \text{Mean}_x(A) = \frac{1}{|A|} \sum_{i \in A} x_i \). The optimality conditions can be stated via ordered partitions \( \Pi = (B_1, B_2, \ldots, B_m) \) of \( V \), meaning that the sets \( B_i \) are disjoint, \( \bigcup_{j=1}^k B_j = V \), and \( \Pi \) is ordered so that \( 1 + \max_{i \in B_j} i = \min_{i \in B_{j+1}} i \).

Specifically, for any such partition we define \( y_\Pi = (y_1, y_2, \ldots, y_m) \), where \( y_j = \text{Mean}_x(B_j) 1_{|B_j|} \) and \( 1_k = \{1\}^k \) is the vector of all ones. In other words, \( y_\Pi \) is defined by taking block-wise averages of \( x \) with respect to \( \Pi \). By analyzing the KKT conditions of problem (8.6), we obtain the following well-known condition.

**Lemma 8.1** ([170]). An ordered partition \( \Pi = (B_1, B_2, \ldots, B_m) \) is optimal iff the following hold

1. (Primal feasibility) For any two blocks \( B_j \) and \( B_{j+1} \) we have

\[ \text{Mean}_x(B_j) \leq \text{Mean}_x(B_{j+1}). \]  

(8.7)

2. (Dual feasibility) For every block \( B \in \Pi \) and each \( i \in B \) define

\[ \text{Pre}_B(i) = \{j \in B \mid j \leq i\}. \]
Then, the condition reads
\[ \text{Mean}_x(\text{Pre}_B(i)) - \text{Mean}_x(B) \geq 0. \] (8.8)

Points where Equation (8.7) is satisfied are of special interest, because of the following result.

**Lemma 8.2.** Denote by Sym(V) the set of all permutations \( \sigma : V \rightarrow V \) of \( V \). If for some \( B_j \) and \( B_{j+1} \) the first condition is satisfied with equality, we can merge the two sets so that the new coarser partition \( \Pi' \) will also be optimal.

**Proof.** The first condition (Equation (8.7)) is obviously satisfied after merging, so that we only have to prove that dual feasibility (Equation (8.8)) will not be violated. Denote the new merged block by \( B = B_j \cup B_{j+1} \). First, note that by assumption \( \text{Mean}_x(B) = \text{Mean}_x(B_j) = \text{Mean}_x(B_{j+1}) \). For any \( i \in B_j \) we will have \( \text{Mean}_x(\text{Pre}_B(i)) = \text{Mean}_x(\text{Pre}_B(i)) \), so the dual feasibility condition is satisfied from the assumption that the original partition was optimal. On the other hand, for \( i \in B_{j+1} \)

\[
\text{Mean}_x(B_j \cup \text{Pre}_{B_{j+1}}(i)) - \text{Mean}_x(B_j \cup B_{j+1}) = \alpha \text{Mean}_x(B_j) + (1 - \alpha) \text{Mean}_x(\text{Pre}_{B_{j+1}}(i)) - \text{Mean}_x(B_j \cup B_{j+1})
\]

\[
= \alpha \left[ \text{Mean}_x(B_j) - \text{Mean}_x(B_j) \right] + (1 - \alpha) \left[ \text{Mean}_x(\text{Pre}_{B_{j+1}}(i)) - \text{Mean}_x(B_{i+1}) \right] \geq 0,
\]

where \( \alpha = \frac{|B_j|}{|B_j| + |B_{j+1}|} \).

Thus, in the remaining of this section we will assume that the sets \( B_j \) are chosen maximally. We will now introduce a notion that will be crucial in the subsequent analysis.

**Definition 8.1.** For any block \( B \), we say that \( i \in B \) is a breakpoint if it holds that \( \text{Mean}_x(\text{Pre}_B(i)) = \text{Mean}_x(B) \) and \( i \) is not the right end-point of \( B \), i.e., \( i < \max_{i' \in B} i' \).

From an optimization perspective, any breakpoint is equivalent to non-strict complementariness of the corresponding Lagrange multiplier. From a combinatorial perspective, they correspond to positions where we can refine \( \Pi \) into a finer partition \( \Pi' \) that gives rise to the same point, i.e., so that \( y_{\Pi} = y_{\Pi'} \) (if we merge blocks using Lemma 8.2, the point where we merge them will become a breakpoint). We are now ready to discuss the
differentiability of \( y(x) \). Because projecting onto convex sets is a proximal operator and thus non-expansive, we have the following as an immediate consequence of Rademacher’s theorem.

**Lemma 8.3.** The function \( y(x) \) is 1-Lipschitz continuous and differentiable almost everywhere.

We will denote by \( \partial_{x_k}^- \) and \( \partial_{x_k}^+ \) the left and right partial derivatives with respect to \( x_k \). For any index \( i \) we will denote by \( u(k) \) (\( l(k) \)) the breakpoint with the smallest (largest) coordinate larger (smaller) than \( k \). Define it as \(+\infty\) (\(-\infty\)) if no such point exists. Moreover, denote by \( \Pi(z) \) the collection of indices where \( z \) takes on distinct values, i.e., \( \Pi(z) = \bigcup_{i=1}^n \{ i' \in V \mid z_i = z_{i'} \} \).

**Theorem 8.1.** Let \( k \) be any coordinate and let \( B \in \Pi(y(x)) \) be the block containing \( i \). Also define \( B_+ = \{ i \in B \mid i \geq u(k) \} \) and \( B_- = \{ i \in B \mid i \leq l(k) \} \). Hence, for any \( i \in B \)

\[
\partial_{x_k}^+(y_i) = \left[ i \in B \setminus B_- \right] / |B \setminus B_-|, \quad \text{and} \\
\partial_{x_k}^-(y_i) = \left[ i \in B \setminus B_+ \right] / |B \setminus B_+|.
\]

**Proof.** We will show how to compute the derivative in direction \( \partial_{x_k}^+ \), as the other case follows analogously. Let us split \( B \) into \( B_- \) and \( B'_+ = B \setminus B_- = \{ i_1 + 1, i_1 + 2, \ldots, i_1 + m \} \), and consider the partition \( \Pi' \) formed by replacing \( B \) with \( \{ B_-, B'_+ \} \). We will show that the above partition is optimal when we increase \( x_k \) by any sufficiently small perturbation \( \varepsilon > 0 \). Then, the claim immediately follows, because \( y(x) \) is defined by taking averages inside an optimal partition. We thus have to show that both of the conditions in Lemma 8.1 will hold for \( \Pi' \). The first condition obviously holds as the elements of \( \Pi(y(x)) \) correspond to different values, so there is some non-zero gap between the averages of consecutive blocks. Let us denote by \( x_\varepsilon \) the perturbed point, i.e. \( x_{\varepsilon,j} = x_j + \lceil j = k \rceil \varepsilon \). Because \( B_- \) ends with a breakpoint and \( k \notin B_- \), we have that

\[
\text{Mean}_{x_\varepsilon}(B_-) = \text{Mean}_x(B) = \text{Mean}_x(B'_+), \quad (8.9)
\]

which means that the second condition also holds for \( B_- \) as none of its points are perturbed, so that the averages stay the same. On the other hand, for any \( i \in B'_+ \)

\[
\text{Mean}_x(\text{Pre}_B(i)) = (1 - \alpha)\text{Mean}_x(B_-) + \alpha\text{Mean}_x(\text{Pre}_{B'_+}(i)),
\]
where \( \alpha = \frac{i - i_1}{|B|} \in (0, 1] \). Moreover, if we subtract \( \text{Mean}_x(B) \) from both sides of the above equality we have (from Equation (8.9))

\[
\text{Mean}_x(\text{Pre}_B(i)) - \text{Mean}_x(B) = \alpha [\text{Mean}_x(\text{Pre}_{B+}(i)) - \text{Mean}_x(B'_+)].
\]

(8.10)

Hence, for any \( i \in B'_+ \)

\[
\text{Mean}_x(\text{Pre}_{B+}(i)) - \text{Mean}_x(B'_+) = \\
\text{Mean}_x(\text{Pre}_{B+}(i)) + \left\lfloor i \geq k \right\rfloor \frac{\varepsilon}{i - i_1} - \text{Mean}_x(B'_+) - \frac{\varepsilon}{|B'_+|} = \\
\text{Mean}_x(\text{Pre}_{B+}(i)) - \text{Mean}_x(B'_+) + \varepsilon \left[ \left\lfloor i \geq k \right\rfloor \frac{1}{i - i_1} - \frac{1}{m} \right] = \\
\frac{1}{\alpha} \left[ \text{Mean}_x(\text{Pre}_B(i)) - \text{Mean}_x(B) \right] + \varepsilon \left[ \left\lfloor i \geq k \right\rfloor \frac{1}{i - i_1} - \frac{1}{m} \right],
\]

where in the last equality we used Equation (8.10). What remains is to show that the above quantity is always positive except at the end-point. First, if \( i \geq k \), then the \( F \) is strictly positive for all points except for the end-point \( i = i_1 + m \) in which case it evaluates to zero. Now, let \( i < k \). Then, because \( i > l(k) \) it can-not be a breakpoint, so that \( D \geq 0 \). Then, for all \( \varepsilon \) smaller than some sufficiently small \( \delta \) the quantity \( F \) will also be positive, which implies dual feasibility for \( B'_+ \) and completes the proof.

Note that these derivatives agree iff there are no breakpoints, which means that the existence of breakpoints is an isolated phenomenon due to Lemma 8.3. In this case the Jacobian exists and has a very simple block-diagonal form. Namely, it is equal to

\[
\frac{\partial y}{\partial x} = \Lambda(y(x)) \equiv \text{blkdiag}(C_{|B_1|}, C_{|B_2|}, \ldots, C_{|B_m|}),
\]

(8.11)

where \( C_k = 1_{k \times k}/k \) is the averaging matrix with elements \( 1/k \). We will use \( \Lambda(z) \) for the matrix taking block-wise averages with respect to the blocks \( \Pi(z) \). As promised in the introduction, Jacobian multiplication \( \Lambda(y(x))u \) is linear as we only have to perform block-wise averages.
In this section we will assume that we have a function $F_{\theta}$ parametrized by some $\theta \in \mathbb{R}^d$ we are going to learn. For example, we could have a mixture model

$$F_{\theta}(A) = \sum_{j=1}^{d} \theta_j G^j(A), \quad (8.12)$$

for some fixed submodular functions $G^j : 2^V \rightarrow \mathbb{R}$. In this case, to ensure that the resulting function is submodular we also want to enforce $\theta_j \geq 0$ unless $G^j$ is modular. We would like to note that the discussion in this section goes beyond such models. Remember that the min-norm point is defined as

$$y_{\theta} = -\text{argmin}_y f_\theta(y) + \frac{1}{2} \|y\|^2, \quad (8.13)$$

where $f_\theta$ is the Lovász extension\(^1\) of $F_\theta$. Hence, we want to compute $\partial y / \partial \theta$. To make the connection with isotonic regression, remember (Algorithm 1) how we evaluate the Lovász extension at $y$. First, we pick a permutation $\sigma$ that sorts $y$, and then evaluate it as $f_\theta(y) = f_\theta(\sigma)^T y$, where $f_\theta(\sigma)$ has entries

$$[f_\theta(\sigma)]_{\sigma(i)} = F_\theta(\{\sigma(i)\} \cup \{\theta(1), \ldots, \sigma(i-1)\}).$$

Hence, as we have also shown in Lemma 2.4, the Lovász extension is linear on the set of all vectors that are sorted by $\sigma$. Formally, for any permutation $\sigma$ the Lovász extension is equal to $f_\theta(\sigma)^T y$ on the order cone

$$\mathcal{O}(\sigma) = \{y \mid y_{\sigma(n)} \leq y_{\sigma(n-1)} \leq \cdots \leq y_{\sigma(1)}\}.$$  

Given a permutation $\sigma$, if we constrain eq. (8.13) to $\mathcal{O}(\sigma)$ we can replace $f_\theta(y)$ by the linear function $f_\theta(\sigma)^T y$, so that the problem reduces to

$$y_{\theta}(\sigma) = -\text{argmin}_{y \in \mathcal{O}(\sigma)} \frac{1}{2} \|y + f_\theta(\sigma)\|^2, \quad (8.14)$$

which is an instance of isotonic regression if we relabel the elements of $V$ using $\sigma$. Roughly, the idea is to instead differentiate eq. (8.14) with $f_\theta(\sigma)$ computed at the optimal point $y_{\theta}$. However, because we can choose an arbitrary order among the elements with equal values, there will be many permutations that sort $y_{\theta}$, and this extra choice we have seems very problematic. Nevertheless, let us continue with this strategy and analyze the

---

\(^1\) To keep the notation succinct, we use $f_\theta$ instead of $[f_\theta]_L$ for the Lovász extension of $F_\theta$. 
resulting approximations to the Jacobian. We propose the following approxima-
tion to the Jacobian
\[
\frac{\partial y_\theta}{\partial \theta} \approx \hat{J}_\sigma = \Lambda(y_\theta) \times \frac{\partial f_\theta(\sigma)}{\partial \theta}
\]
\[
= \Lambda(y_\theta) \times \left[ \partial_{\theta_1} f_\theta(\sigma) \mid \partial_{\theta_2} f_\theta(\sigma) \mid \cdots \mid \partial_{\theta_d} f_\theta(\sigma) \right],
\]
where \(\Lambda(y_\theta)\) is used as an approximation of a Jacobian which might not
exist. Fortunately, due to the special structure of the linearizations, we have
the following result that the gradient obtained using the above strategy
does not depend on the specific permutation \(\sigma\) that was chosen.

**Theorem 8.2.** If \(\partial_{\theta_k} F(A)\) exists for all \(A \subseteq V\) the approximate Jacobians \(\hat{J}_\sigma\)
are equal and do not depend on the choice of \(\sigma\). Specifically, the \(j\)-th block of any
element \(i \in B \in \Pi(y_\theta)\) is equal to
\[
\frac{1}{|B|} \partial_{\theta_j} F_\theta(B \mid \{i' \mid [y_\theta]_{i'} < [y_\theta]_i\}). \tag{8.15}
\]

**Proof of Theorem 8.2.** We will show that the \(j\)-th column of the approximate
Jacobian, equal to \(\Lambda(y_\theta)\partial_{\theta_j} f_\theta(\sigma)\) does not depend on the choice of \(\sigma\). First,
note that the \(\sigma(i)\)-th coordinate of \(\partial_{\theta_j} f_\theta(\sigma)\) is equal to
\[
[\partial_{\theta_j} f_\theta(\sigma)]_{\sigma(i)} = \partial_{\theta_j} F_\theta(\{\sigma(i)\} \mid \{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\})
\]
\[
= \partial_{\theta_j} F_\theta(\{\sigma(1), \sigma(2), \ldots, \sigma(i)\}) \tag{8.16}
\]
\[
- \partial_{\theta_j} F_\theta(\{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\}).
\]
Let \(B\) be the block where \(i\) belongs to, i.e. \(B = \{i' \mid [y_\theta]_{i'} = [y_\theta]_i\}\). Because
we chose a partition \(\sigma\) that sorts \(y_\theta\), all elements in \(B\) must have their
elements placed consecutively under \( \sigma \), i.e. it there must exist some \( i_1 \) so that \( B = \{ \sigma(i_1 + 1), \sigma(i_2 + 2), \ldots, \sigma(i_1 + |B|) \} \). Then

\[
\text{Mean}_{\theta} f_{\theta}(\sigma)(B) = \frac{1}{|B|} \sum_{i=i_1+1}^{i_1+|B|} \partial_{\theta_j} F_{\theta}(\{\sigma(i)\} | \{\sigma(1), \ldots, \sigma(i_1)\})
\]

\[
= \frac{1}{|B|} \left[ \sum_{i=i_1+1}^{i_1+|B|} \partial_{\theta_j} F_{\theta}(\{\sigma(1), \sigma(2), \ldots, \sigma(i)\}) - \partial_{\theta_j} F_{\theta}(\{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\}) \right]
\]

\[
= \frac{\partial_{\theta_j}}{|B|} F_{\theta}(\{\sigma(1), \ldots, \sigma(i_1 + |B|)\} | \{\sigma(1), \ldots, \sigma(i_1)\})
\]

\[
= \frac{\partial_{\theta_j}}{|B|} F_{\theta}(B \mid \{i' \mid [y_{\theta}]_{i'} < [y_{\theta}]_i\}),
\]

where the third equality follows from the telescoping of the summands.

\[\square\]

**Graph Cuts**  

As a special, but important case, let us analyze how the approximate Jacobian looks like for a cut function (eq. (8.1)), in which case eq. (8.13) reduces to

\[
y(w, v) = -\arg\min_{y \in \mathbb{R}^n} \sum_{\{i,j\} \in E} w_{i,j} |y_i - y_j| + v^T y + \frac{1}{2} \|y\|^2. \quad (8.17)
\]

Let \( \Pi(y(w, v)) = (B_1, B_2, \ldots, B_m) \). For any element \( i \in V \) we will denote by \( \eta(i) \in \{1, 2, \ldots, m\} \) the index of the block where it belongs to. Then, the approximate Jacobian \( \hat{f} \) at \( \theta = (w, v) \) has entries

\[
\hat{\partial}_{v_j}(y_i) = \|\eta(i) = \eta(j)\| / |B_{\eta(k)}|, \quad \text{and}
\]

\[
\hat{\partial}_{w_{i,j}}(y_k) = \begin{cases} 
\text{sign}(y_i - y_j) \frac{1}{|B_{\eta(k)}|} & \text{if } \eta(k) = \eta(i), \text{ or} \\
\text{sign}(y_j - y_i) \frac{1}{|B_{\eta(k)}|} & \text{if } \eta(k) = \eta(j), \text{ and} \\
0 & \text{otherwise},
\end{cases}
\]

where the sign function is defined to be zero if the argument is zero. Intu-itably, increasing the modular term \( v_i \) by \( \delta \) will increase all the coordinates \( B \) in \( y \) that are in the same segment by \( \delta / |B| \). On the other hand, increasing the weight of an edge \( w_{i,j} \) will have no effect if \( i \) and \( j \) are already on \( y \) in
the same segment, and otherwise it will pull the segments containing \(i\) and \(j\) together by increasing the smaller one and decreasing the larger one. In Figure 8.4 we provide a pytorch module that executes the back propagation pass in \(O(|V| + |E|)\) time in about 10 lines of code, and we also derive the approximate Jacobians for concave-of-cardinality and facility location functions.

**Cardinality Functions** Remember that these functions are of the form \(F(A) = h(|A|)\) for some concave \(h\) satisfying \(h(0) = 0\). Because we only evaluate \(h\) at the integers \(\{1, 2, \ldots, n\}\), we can parametrize it by \(h(k) = \sum_{i=1}^{k} \theta_i\) for some reals \(\theta_1 \geq \theta_2 \geq \cdots \geq \theta_n\) (this condition is equivalent to the concavity of \(h\)). Then, \(\partial_{\theta_k} f_\sigma\) is equal to one on coordinate \(\sigma(k)\) and zero elsewhere. Hence, for any block \(B \in \Pi(\mathbf{y}_\theta)\), the column of the approximate Jacobian corresponding to \(\partial_{\theta_k}\) will have values that are identical and equal to \([\sigma(k) \in B]/|B|\). Again, backpropagation takes linear time.

**Facility Location** Consider functions of the form \(F(A) = \max_{i \in A} w_i\) for some non-negative weights \(w_i\). For any weight \(w_k\) the approximate derivative \(\hat{\partial}_{w_k}\) is non-zero only for the block \(B\) is containing \(\sigma(k)\) and the block \(B -\) succeeding it. It can be again computed in linear time.

**8.5 Analysis**

We will now theoretically analyze the conditions under which our approximation is correct, and then give a characterization of the exact directional derivative together with a polynomial algorithm that computes it. The first notion that will have implications for our analysis is that of (in)separability.

**Definition 8.2.** The function \(F: 2^V \to \mathbb{R}\) is said to be separable if there exists some \(B \subseteq V\) such that \(B \not\in \{\emptyset, V\}\) and \(F(V) = F(B) + F(V - B)\).

The term separable is indeed appropriate as it implies that \(F(A) = F(A \cap B) + F((V - B) \cap A)\) for all \(A \subseteq V\) [20, Proposition 4.3], i.e., the function splits as a sum of two functions on disjoint domains. Hence, we can split the problem into two (on \(B\) and \(V - B\)) and analyze them independently. We would like to point out that separability is checkable in cubic time using the algorithm of Queyranne [172]. To simplify the notation, we will assume that we want to compute the derivative at point \(\theta' \in \mathbb{R}^d\) which results in the min-norm point \(y' = y_\theta \in \mathbb{R}^n\). We will furthermore assume that \(y'\) takes on unique values \(\gamma_1 < \gamma_2 < \cdots < \gamma_k\) on sets \(B_1, B_2, \ldots, B_k\).
respectively, and we will define the chain $\emptyset = A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots \subseteq A_k = V$ by $A_j = \bigcup_{j'=1}^j B_{j'}$. A central role in the analysis will be played by the set of active constraints at $y_\theta$, which makes sense given that we expect small perturbations in $\theta'$ to result in small changes in $y_\theta$ as well.

**Definition 8.3.** For any submodular function $F: 2^V \to \mathbb{R}$ and any point $z \in B(F)$ we shall denote by $D_F(z)$ the lattice of tight sets of $z$ on $B(F)$, i.e.

$$D_F(z) = \{ A \subseteq V \mid z(A) = F(A) \}.$$  

The fact that the above set is indeed a lattice is well-known [22]. Moreover, note that $D_F(z) \supseteq \{ \emptyset, V \}$. We will also define $D' = D_{F_\theta'}(y')$, i.e., the lattice of tight sets at the min-norm point.

### 8.5.1 When will the approximate approach work?

We will analyze sufficient conditions so that irrespective of the choice of $\sigma$, the isotonic regression problem eq. (8.14) has no breakpoints, and the left and right derivatives agree. To this end, define for any $j \in \{1, 2, \ldots, k\}$ the following normalized submodular function

$$F_j: 2^{B_j} \to \mathbb{R} \text{ as } F_j(H) = F_\theta'(A_{j-1} \cup H) - F_\theta'(A_{j-1}), \quad (8.18)$$

where we have dropped the dependence on $\theta'$ as it will remain fixed throughout this section.

**Theorem 8.3.** The approximate problem (8.14) is argmin-continuously differentiable irrespective of the chosen permutation $\sigma$ sorting $y_\theta$ if and only if any of the following equivalent conditions hold.

(a) $\arg\min_{H \in B_j} [F_j(H) - F_j(B_j)|H|/|B_j|] = \{ \emptyset, B_j \}$.

(b) $y'_B_j \in \text{relint}(B(F_j))$, i.e. $D_{F_j}(y'_B_j) = \{ \emptyset, B_j \}$, which is only possible if $F_j$ is inseparable.

**Proof.** From Theorem 8.1 we can see that the left and right derivatives will agree if and only if each block $B_j$ has no of the chosen order cone. We will refer to such blocks as being unbreakable. The first condition then follows from the following lemma.

**Lemma 8.4.** Block $B_j$ is unbreakable if and only if there is no $H \notin \{ \emptyset, B_j \}$ satisfying

$$F_j(H)/|H| = F_j(B_j)/|B_j|.$$
Moreover, this condition is equivalent to
\[
\arg\min_{H \in B_j} [F_j(H) - F_j(V)|H|/|B_j|] = \{\emptyset, B_j\}.
\]

Proof. Let w.l.o.g. \(B_j = \{1, 2, \ldots, k\}\) and let \(\sigma: \{1, 2, \ldots, k\} \to \{1, 2, \ldots, k\}\) be the ordering the chosen permutation makes on \(B_j\). Then, at the \(i\)-th position inside \(B_j\), the linearization is equal to
\[
[f(\sigma)]_i = F_j(\{\sigma(i)\} | \{\sigma(1), \sigma(2), \ldots, \sigma(i-1)\}).
\]
Then, due to the telescoping sums we have
\[
\text{Mean}_{f(\sigma)}(\text{Pre}_i(B_j)) = F(\{\sigma(1), \sigma(2), \ldots, \sigma(i)\})/i,
\]
with the special case
\[
\text{Mean}_{f(\sigma)}(\text{Pre}_k(B_j)) = F(B_j)/k = F(B)/|B_j|.
\]
Now, by definition, \(B_j\) is unbreakable if and only if the above two quantities are never equal for \(i < k\) for any possible permutation \(\sigma\), which is exactly what we had to show.

The second condition will follow as a corollary of the following well-known result characterizing the min-norm point.

**Lemma 8.5.** For each \(B_j\) we have that \(y'_B_j = F_j(B_j)/|B_j|\).

Proof. From [22, Theorem 9.1] we know that \(y'(A_j) = F_{\theta'}(A_j)\) for each \(j \in \{1, \ldots, k\}\). Subtracting the equalities for \(j\) and \(j-1\) we have
\[
y(B_j) = F(A_j) - F(A_{j-1}) = F_j(B_j).
\]
Now, the result follows because we have assumed that all elements in \(B_j\) are equal.

**Corollary 5.** Block \(B_j\) is unbreakable iff \(\mathcal{D}_{F_j}(y'_B_j) = \{\emptyset, B_j\}\).

Proof. Assume there exists some other set \(H \subseteq B_j\) that is tight. This implies that \(y'_{B_j}(H) = F_j(H)\). Then, Lemma 8.5 implies that
\[
F_j(B_j)|H|/|B_j| = F_j(H),
\]
which is exactly equal to the first condition.
The fact that the base polytope has any point in the relative interior being equivalent to inseparability is well-known, see e.g. [20, Proposition 4.6] or [22, Theorem 3.36].

In other words, we can equivalently say that the optimum has to lie on the interior of the face.

We would like to point out that one can obtain the same derivatives as the ones suggested in §8.4, if we perturb the KKT conditions, as done by Amos & Kolter [165]. If we use that approach, in addition to the computational challenges, there is the problem of non-uniqueness of the Lagrange multiplier, and moreover, some valid multipliers might be zero for some of the active constraints. This would render the resulting linear system rank deficient, and it is not clear how to proceed. Remember that when we analyzed the isotonic regression problem in §8.3 we had non-differentiability due to the exactly same reason — zero multipliers for active constraints, which in that case correspond to the breakpoints.

**Theorem 8.4.** For a specific Lagrange multiplier there exists a solution to the perturbed KKT conditions derived by Amos & Kolter [165] that gives rise to the approximate Jacobians from Section 8.4. Moreover, this multiplier is unique if the conditions of Theorem 8.3 are satisfied.

**Proof of Theorem 8.4.** Remember that the min-norm point is defined as

$$\arg\min_{y \in \mathbb{R}^n} \frac{1}{2} \|y\|^2$$

subject to $y(A) \leq F_\theta(A)$ for all $A \subseteq V$ and $y(V) = F_\theta(V)$. Introducing multipliers $\lambda_A$ for the inequality constraints and $\mu$ for the equalities we arrive at the Lagrangian

$$L(y, \lambda, \mu) = \frac{1}{2} \|y\|^2 + \sum_{A \subseteq V} \lambda_A (y(A) - F_\theta(A)) + \mu (y(V) - F_\theta(V)).$$

Then, setting $\partial_{y_i} L$ to zero yields the condition

$$\mu + \sum_{A \ni i} \lambda_A = -y_i. \quad (8.19)$$

As $y'$ is the min-norm point for $F_\theta'$, we know that the sets $A_1, A_2, \ldots, A_k = V$ are all tight (see Lemma 8.5). Remember that $y'$ took values $\gamma_1 < \gamma_2 <$
\[ \cdots < \gamma_k \text{ on sets } B_1, B_2, \ldots, B_k \text{ respectively. We suggest the following family of multipliers} \]

\[ \mu = \gamma_k = -\max_i y_i \]

\[ \lambda_A = \begin{cases} 
\gamma_{j+1} - \gamma_j \geq 0 & \text{if for } A = A_j \text{ for } j \in \{1, 2, \ldots, k - 1\}, \text{ or} \\
0 & \text{otherwise.} 
\end{cases} \]

They indeed satisfy the condition in eq. (8.19) because for every \( i \in B_j \) we have that

\[ \mu + \sum_{A \ni i} \lambda_A = \mu + \sum_{j' = j}^{k-1} \lambda_{A_{j'}} = \gamma_k - \sum_{j' = j}^{k-1} \gamma_{j'+1} - \gamma_j' = -\gamma_j' = -y_i. \]

The fact that these multipliers are unique will easily follow if we show that the conditions imply that \( A_1, A_2, \ldots, A_k \) are the only tight sets. Then, it can be easily seen that the only multipliers satisfying eq. (8.19) are those we have just suggested, by doing an induction on \( j \) starting from \( j = k \) (corresponding to \( \nu \)) and going backwards. For the sake of contradiction, assume that the conditions hold and that there exists some \( H \) different from the members of the chain that is tight, i.e. \( y'(H) = F_{\theta'}(H) \) and \( A_{j-1} \subsetneq H \subsetneq A_j \). Then, if we define \( H' = H - A_{j-1} \notin \{\emptyset, B_j\} \), we have that

\[ F_j(H') = F_{\theta'}(H' \cup A_{j-1}) - F_{\theta'}(A_{j-1}) \]
\[ = F_{\theta'}(H) - F_{\theta'}(A_j) \]
\[ = y'(H) - F_{\theta'}(A_j) \]
\[ = y'(H') \frac{|H'|}{|F_{\theta'}(B_j)|/|B_j|}, \]

where the last step follows from Lemma 8.5. However, this means that \( H' \) violates the assumption condition for \( F_j \), which is a contradiction to our assumption.

Create the matrix \( G = \{0, 1\}^{(2^n-1) \times n} \) that has the indicator vectors of the inequality constraints as its rows. Similarly, create the vector \( h \in \mathbb{R}^{2^n-1} \) with elements \( F(A) \), and define \( b = F(V) \). Let us first compute \( \partial y / \partial b \), for which we have to solve the following system

\[ d y + G^\top d \lambda + 1 d \mu = 0 \]
\[ \operatorname{diag}(\lambda^*) G d y + \operatorname{diag}(G y - h) d \lambda = 0 \]
\[ 1^\top d y = 1 \]
We suggest the solution $dy = 1_{B_k}/|B_k|, d\mu = 1/|B_k|$ and $d\lambda$ to be zero except coordinate $d\lambda_{A_{k-1}} = -1/|B_k|$, which can be easily seen to satisfy the conditions. Namely, the second condition translates to $1^T_{A_j}dy = 0$ for $j = 1, 2, \ldots, k-1$, which is true at $y$ is non-zero only on $B_k$. The third condition is obvious, while the first one reads
\[
    dy_i = -\sum_{A \ni i} d\lambda_A + d\mu \\
    = -1/|B_k|[i \in A_{k-1}] + 1/|B_k| \\
    = 1/|B_k|[i \notin A_{k-1}] \\
    = 1/|B_k|[i \in B_k].
\]

Now, let us compute $\partial y/\partial h$, for which we have to solve the following system
\[
    dy + G^T d\lambda + 1d\mu = 0 \\
    \text{diag}(\lambda)Gdy + \text{diag}(Gy - h)d\lambda = \text{diag}(\lambda) \\
    1^T dy = 0
\]
As Ansatz we suggest $d\mu = 0$ and setting only the following entries of $dy$ to non-zero
\[
    dy_{i,A} = \begin{cases} 
        +1/|B_j| & \text{if } i \in B_j \text{ and } A = A_j \text{ for } j < k, \text{ or} \\
        -1/|B_{j+1}| & \text{if } i \in B_{j+1} \text{ and } A = A_j \text{ for } j < k, \text{ or} \\
        0 & \text{otherwise},
    \end{cases}
\]
and
\[
    d\lambda_{A',A} = \begin{cases} 
        -1/|B_j| & \text{if } A' = A = A_j \text{ and } j < k, \text{ or} \\
        +1/|B_{j+1}| + 1/|B_j| & \text{if } A' = A_j \text{ and } A = A_{j+1}, \text{ if } j < k, \text{ or} \\
        0 & \text{otherwise}.
    \end{cases}
\]
Note that the third condition is immediately satisfied for $dy$, as every column sums to zero. The third condition for those rows of $d\lambda$ corresponding to $A_1, A_2, \ldots, A_{k-1}$ reads
\[
    \sum_{i \in A_j} dy_{A',A_j} = [A = A_j],
\]
which is satisfied because the $A_j$-th column of $dy$ is $+1/|B_j|$ on $B_j$ and we do not sum over its other elements which are in $B_{j+1}$. Finally, the first condition is non-vacuous only for the non-zero columns $A_1, A_2, \ldots, A_{k-1}$, in which case it also holds because we have

$$
dy_{i,A_j} = - \sum_{A \ni i} d\lambda_{A,A_j}
= [i \in A_j]/|B_j| - (1/|B_j| + 1/|B_{j+1}|)[i \in A_j+1]
= [i \in B_j]/|B_j| - [i \in B_{j+1}]/|B_{j+1}|.
$$

Then, if these were the actual true Jacobians, for any parameter $\theta_j$ and output $i \in B_j$ we have

$$
\frac{\partial y_i}{\partial \theta_j} = \sum_{A \subseteq V} \frac{\partial y_i}{\partial F_\theta(A)} \times \frac{\partial F_\theta(A)}{\theta_j}
= \sum_{j=1}^{k-1} \frac{\partial y_i}{\partial F_\theta(A_j)} \times \frac{\partial F_\theta(A_j)}{\theta_j}
= \frac{\partial y_i}{\partial F_\theta(A_{j-1})} \times \frac{\partial F_\theta(A_j)}{\theta_j} + \frac{\partial y_i}{\partial F_\theta(A_j)} \times \frac{\partial F_\theta(A_j)}{\theta_j}
= \frac{\partial \theta_j}{|B_j|} F_\theta(B_j).
$$

8.5.2 Exact computation

Unfortunately, computing the gradients exactly seems very complicated for arbitrary parametrizations $F_\theta$, and we will focus our attention to mixture models of the form given in eq. (8.12). The directions $v$ where we will compute the directional derivatives will have in general non-negative components $v_j$, unless $G^i$ is modular. By leveraging the theory of Shapiro [173], and exploiting the structure of both the min-norm point and the polyhedron $B(F_v \mid D')$ we obtain at the following result.
Theorem 8.5. Assume that $F_{\theta'}$ is inseparable and let $v$ be any direction so that $F_v$ is submodular. The directional derivative $\frac{\partial y}{\partial \theta_j}$ at $\theta'$ in direction $v$ is given by the solution of the following problem.

$$\min_{d} \frac{1}{2} \|d\|^2,$$

subject to $d \in B(F_v | D')$, and
$$d(B_j) = F_v(A_j) \text{ for } j \in \{1, 2, \ldots, k\}.$$  

(8.20)

Proof of Theorem 8.5. First, because the function $\theta \rightarrow y_{\theta}$ is continuous under the assumptions ([173, Lemma 2.1]), we know that in a neighbourhood of $\theta'$ no inactive constraints can activate. Hence, we can focus on the active constraints only.

Lemma 8.6. The four assumptions made in [173] hold whenever $F_{\theta'}$ is inseparable.

Proof. We show below why each one holds.

1. We need a neighbourhood $N_{\theta'}$ of $\theta'$ so that for all $\theta'' \in N_{\theta'}$ the base polyhedra are bounded. Obviously holds as $B(F_{\theta}) = \sum_{j=1}^{m} w_j B(G^j)$.

2. We have already discussed the uniqueness of the min-norm point.

3. The Mangasarian-Fromovitz condition is implied by Slater’s condition, which in case holds whenever the function is inseparable [22, Theorem 3.36].

4. Clearly follows from the strong convexity of the objective.

Let us define the shorthands $D' = D_{F_{\theta'}}(y')$ and $\Sigma(v) = B(F_v | D')$. It is easy to see that our definition of $\Sigma(v)$ agrees with that in [173, (2.11)] because the constraints are linear in both $y$ and $\theta$, and there is no term containing both $y$ and $\theta$. Namely, each constraint is of the form

$$g_A(y, \theta) = y^\top 1_A - \sum_{j=1}^{m} \theta_j G^j(A),$$
we have its linear approximation
\[
\alpha_A(u, v) = u^\top \nabla_{y^A} (y', \theta') + v^\top \nabla_{\theta^A} (y', \theta') \\
= u^\top 1_A - \sum_{j=1}^m v_j G_j(A) \\
= u(A) - F_v(A).
\]

Then, the equivalence is clear given the definition [173, (2.11)]
\[
\Sigma(v) = \{ u \in \mathbb{R}^n | \alpha_A(u, v) \leq 0 \text{ for all } A \in D' \} \cap \{ u \in \mathbb{R}^n | \alpha_V(u, v) = 0 \}.
\]

**Lemma 8.7.** A point \( u \in \Sigma(v) \) belongs to \( \Sigma(v) = \operatorname{argmin}_{u \in \Sigma(v)} u^\top y' \) iff it satisfies
\[
u(A_j) = F_v(A_j) \text{ for } j \in \{1, 2, \ldots, k\}.
\]

**Proof.** First, note that \( y^\top \theta : V \to \mathbb{R} \) is monotone non-decreasing on \( D' \) because for any \( i \in B_i \) we have that
\[
\operatorname{dep}(y', i) = \{ A \subseteq V : A \ni i \text{ and } y'(A) = F_\theta(A) \} \subseteq A_i
\]
due to [22, Theorem 9.1(iii)], and the lattice we consider is exactly the one generated from the tight sets of the min-norm point. Then, the result follows from [22, Theorem 3.15].

Now, because the Lagrangian (defined as (3.3) in [173])
\[
\mathcal{L}(y, \theta, \lambda, \nu) = \frac{1}{2} \|y\|^2 + \sum_{A \in D'} \lambda_A (y(A) - \sum_{j=1}^d \theta_j G_j^i(A)) + \nu(y(V)) + \sum_{j=1}^d \theta_j G_j^i(V))
\]
has no terms containing both \( y \) and \( \theta \) we have that
\[
\xi_\lambda(u, v) = \frac{1}{2} u^\top \nabla^2_{yy} \mathcal{L} u + u^\top \nabla^2_{y\theta} \mathcal{L} v + \frac{1}{2} v^\top \nabla^2_{yy} \mathcal{L} v = \frac{1}{2} \|u\|^2,
\]
so that the function is independent of its second argument. Then, \( \xi_{v, h}(u) \) (defined as (4.10) in [173]) reduces to
\[
\xi_{v, h}(u) = \frac{1}{2} \|u\|^2 + z(v, h),
\]
for some function \( z(v, h) \) independent of \( u \). Then, note that

\[
\arg\min_{u \in \Sigma(v)} \zeta_{v,h}(u) = \arg\min_{u \in \Sigma(v)} z(v, h) + \frac{1}{2} \|u\|^2 = \arg\min_{u \in \Sigma(v)} \frac{1}{2} \|u\|^2,
\]

hence the optimum does not depend on the choice of \( h \). Then, the claim follows from [173, Theorem 5.1].

First, note that this is again a min-norm problem, but now defined over a reduced lattice \( D' \) with \( k \) additional equality constraints. Fortunately, due to these additional equalities, we can split the above problem into \( k \) separate min-norm problems. Namely, for each \( j \in \{1, 2, \ldots, k\} \) collect the lattice of tight sets that intersect \( B_j \) as \( D'_j = \{ H \cap B_j \mid H \in D' \} \), and define

\[
G_j: 2^{B_j} \to \mathbb{R} \text{ as } G_j(A) = F_v(A_{j-1} \cup A) - F_v(A_{j-1}),
\]

where note that the parameter vector \( \theta \) is taken as the direction \( v \) in which we want to compute the derivative. Then, the block of the optimal solution of problem (8.20) corresponding to \( B_j \) is equal to

\[
d_{B_j}^* = \arg\min_{y_j \in B(G_j|D'_j)} \frac{1}{2} \|y_j\|^2,
\]

which is again a min-norm point problem where the base polytope is defined using the lattice \( D'_j \). We can then immediately draw a connection with the results from the previous subsection.

**Corollary 6.** If all lattices are trivial eq. (8.21) agrees with the result from the previous section.

**Proof of Corollary 6.** If the lattices are trivial problem (8.21) reduces to

\[
\min_{y_k: y_k^* \mathbf{1} = G_j(B_j)} \|y_j\|^2,
\]

which has as a solution the constant vector \( y_k^* \) with coordinates all equal to

\[
G_j(B_j) / |B_j| = F_v(B_j \mid A_{j-1}) / |B_j|.
\]
How to solve problem (8.21)? Fortunately, the divide-and-conquer algorithm of Groenevelt [53] can be used to find the min-norm point over arbitrary lattices. We can give to the algorithm the lattice using its graph representation from Theorem 2.6. Remember that we have to compute for each $i \in B_j \in V$ the set

$$D'(i) = \bigcap \{ A \in D'_j \mid A \ni i \},$$

which is the unique smallest set in $\arg\min_{H_j \ni i} F_j(H_j) - y'(H_j)$, and can be computed using submodular minimization. To highlight the difference with the approximation from section 8.4, let us consider a very simple case.

**Lemma 8.8.** Assume that $G_j$ is equal to $G_j(A) = [i \in A]$ for some $i \in B_j$. Then, the directional derivative is equal to $1_p/|P|$, where $P$ is the union of $\{i\}$ and its ancestors in $G(D'_j)$.

**Proof.** First note that the suggested solution is feasible, as any set with non-zero elements must contain $i$. Let us denote by $K$ the set of vertices in the same component as $i$, and consider the three sets: $R = K - P$, $K$ and $B_j - K$, all of which are easily seen to be in $D$. We will now prove optimality by showing that the claimed solution is optimal if we only consider the active set corresponding to these three sets. The corresponding Lagrangian is

$$\frac{1}{2} \| u \|^2 + \lambda_1(u(K) - 1) + \lambda_2(u(K - P) - 0) + \lambda_3(u(B_j - K) - 0) + \eta(u^\top 1 - 1).$$

Setting the derivative wrt $u_k$ for $k \in K$ to zero yields the condition

$$u_k = -\lambda_1 - \lambda_2 [k \notin P] - \eta.$$

For any $k \in P$ we have to satisfy $-\lambda_1 - \eta = 1/|P|$, so as Ansatz take $\lambda_1 = 1/|P|$ and $\eta = -2/|P|$. Then, to satisfy the condition for any other $k \in K - P$ we can use $\lambda_2 = -\lambda_1 - \eta = 1/|P|$, which is also positive. Finally, for any $k \notin K$ the condition is satisfied by using the multiplier $\lambda_3 = \eta = 2/|P|$. 

Hence, while the approximate directional derivative would average over all elements in $B_j$, the true one averages only over a subset $P \subseteq B_j$ and is possibly sparser. The above result gives us the exact directional derivatives for the graph cuts, as each component $G_j$ will be either a cut function on two vertices, or a function of the form in Lemma 8.8. In the first case the directional derivative is zero because $0 \in B(G_j) \subseteq B(G_j \mid D'_j)$. In the second case, we can also use a very simple approximation, which generalizes
the result from [174]— given that $F_j$ is separable over $2^{B_j}$ iff the graph is disconnected, we can first separate the graph into connected components, and then take averages within each connected component instead of over all nodes in $B_j$.

8.5.3 *Structured attention and constraints*

Recently, there has been an interest in the design of structured attention mechanisms, which, as we will now show, can be derived and furthermore generalized using the results in this paper. The first mechanism is the *sparsemax* of Martins & Astudillo [175], defined as the projection of onto the probability simplex, which is the base polytope corresponding to the function $G(A) = \min\{|A|, 1\}$. Concurrently with this work, Niculae & Blondel [174] have analyzed the following problem

$$y = \min_{y \in B(G)} f_L(y) + \frac{1}{2} \|y - z\|^2, \quad (8.22)$$

for the special case when $B(G)$ is the simplex and $f_L$ is the Lovász extension of one of two specific submodular functions. We will consider the general case when $G$ can be any concave-of-cardinality function and $F$ is an arbitrary submodular function. Note that, if either $f_L(y)$ or the constraint were not present in problem (8.22), we could have simply leveraged the theory we have developed so far to differentiate it. Fortunately, as done by Niculae & Blondel [174], we can utilize the result of Yu [176] to significantly simplify (8.22). Namely, because projection onto $B(G)$ preserves the order of the coordinates [55, Lemma 1], we can write the optimal solution $y$ to (8.22) as

$$y = \min_{x \in B(G)} \frac{1}{2} \|y - y'\|, \text{ where } y' = \arg\min_y f_L(y) + \frac{1}{2} \|y - z\|^2.$$

We can hence split problem (8.22) into two subtasks — first, compute $y'$ and then project it onto $B(G)$. As each operation can reduces to a minimum-norm problem, we can differentiate each of them separately, and thus solve (8.22) by *stacking* two submodular layers one after the other.

8.6 *Experiments*

We consider the image segmentation tasks from the introduction, where we are given an RGB image $x \in \mathbb{R}^{n \times 3}$ and are supposed to predict those
Table 8.1: Test set results. We see that incorporating a graph cut solver improves both the accuracy and negative log-likelihood (NLL), while having consistent segmentations with fewer foreground objects.

<table>
<thead>
<tr>
<th></th>
<th>CNN</th>
<th>CNN+GC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean Std. Dev.</td>
<td>Mean Std. Dev.</td>
</tr>
<tr>
<td>Accuracy</td>
<td>0.8103 0.1391</td>
<td>0.9121 0.1034</td>
</tr>
<tr>
<td>NLL</td>
<td>0.3919 0.1911</td>
<td>0.2681 0.2696</td>
</tr>
<tr>
<td># Fg. Obj.</td>
<td>96.9 65.8</td>
<td>25.3 30.6</td>
</tr>
</tbody>
</table>

Figure 8.1: Test set results. We see that incorporating a graph cut solver improves both the accuracy and negative log-likelihood (NLL), while having consistent segmentations with fewer foreground objects.

pixels $A \subseteq V$ containing the foreground object. We used the Weizmann horse segmentation dataset [3], which we split into training, validation and test splits of sizes 180, 50 and 98 respectively. The implementation was done in pytorch, and to compute the min-norm point we used the algorithm from [60]. To make the problem more realistic and challenging, at training time we randomly selected and revealed only 0.1% of the training set labels. We first trained a convolutional neural network with two hidden layers that directly predicts the per-pixel labels, which we refer to as CNN. Then, we added a second model, which we call CNN+GC, that has the same architecture as the first one, but with an additional graph cut layer, whose weights are parametrized by a convolutional neural network with one hidden layer. Details about the architectures can be found in Figure 8.3. We train the models by maximizing the log-likelihood of the revealed pixels, which corresponds to the variational bi-level strategy (eq. (8.3)) due to equation (8.4). We trained using SGD, Adagrad [177] and Adam [178], and chose the model with the best validation score. As evident from the results presented in Section 8.6, adding the discrete layer improves not only the accuracy (after thresholding the marginals at 0.5) and log-likelihood, but it gives steadier results as it makes predictions with fewer connected components (i.e., foreground objects). Moreover, if we have a look at the predictions themselves in Figure 8.2, we can observe that the optimization layer not only removes spurious predictions, but there is is also a qualitative difference in the marginals as they are spatially more consistent.
Figure 8.2: Comparison of results from both models on four instances from the test set (left: CNN, right: CNN+GC). We can see that adding the graph-cut layers helps not only quantitatively, but also qualitatively, as the predictions are more spatially regular and vary smoothly inside the segments.

8.7 Conclusion

In this chapter we considered the problem of learning log-supermodular models via approximate inference. As the maximum likelihood objective is intractable, we have taken a different approach that is fully aware of the approximate inference method used. Namely, we solve the bi-level optimization problem — find the model parameters $\theta$ such that when we minimize the inclusive divergence on the model, we compute an approximate distribution that gives high likelihood to the observed data. Thanks to the close connection between the inclusive divergence and the min-norm problem, we have reduced our analysis to that of the min-norm point. We have
# The network that predicts the per-pixel scores.
nn.Sequential(nn.Conv2d(3, 32, 3, padding=1),
              nn.ReLU(),
              nn.Conv2d(32, 64, 3, padding=1),
              nn.ReLU(),
              nn.Conv2d(64, 1, 3, padding=1))

# The networks that predict the log of the weights.
nn.Sequential(nn.Conv2d(3, 32, 3, padding=1),
              nn.ReLU(),
              nn.Conv2d(32, 1, 3, padding=1))

Figure 8.3: The architectures used in the experiments.

namely analyzed the sensitivity of the min-norm point for parametric submodular functions and provided both a very easy-to-implement practical approximate algorithm for general objectives, and strong theoretical result characterizing the true directional derivatives for mixtures. Not only useful for learning models, but these results can be also seen as allowing the use of submodular minimization inside modern deep architectures. Moreover, we have shown that several recent structured attention mechanisms can be written as two such layers. Finally, we believe that the theoretical results open the new problem of developing algorithms that can compute not only the min-norm point, but also solve for the associated derivatives.
from prox_tv import tv1w_2d  # The total variation solver.
import numpy as np
import torch
from torch.autograd import Function

class TotalVariation2d(Function):
    """A two dimensional total variation function.

    Specifically, given as input ('x', 'w_r', 'w_c'), the output is

    argmin_z 0.5 |x-z|^2 + \sum_{i, j} w_r_{i,j} |x_{i, j} - x_{i, j + 1}| + \sum_{i, j} w_c_{i,j} |x_{i, j} - x_{i + 1, j}|.""

    def forward(self, x, weights_row, weights_col):
        """Solves the TV problem and returns the solution.

        Arguments
        --------
        x : [m, n] matrix holding the input signal
        weights_row : [m, n - 1] matrix holding the horizontal weights
        weights_col : [m - 1, n] matrix holding the vertical weights

        Returns
        -------
        The TV solution, matrix of size [m, n]"

        opt = tv1w_2d(- x.numpy(), weights_col.numpy(), weights_row.numpy())
        opt = torch.Tensor(opt)
        self.save_for_backward(opt)
        return opt

    def backward(self, grad_output):
        opt, = self.saved_tensors
        grad_x = torch.zeros(opt.size())
        # We always compute this derivative.
        values = np.unique(opt.numpy())
        # One group for each unique value.
        groups = [(opt.numpy() == val).reshape(opt.size()) for val in values]
        for group in groups:
            grad_x.numpy()[group] = - np.mean(grad_output.numpy()[group])
        diff_row = torch.sign(opt[:, :-1] - opt[:, 1:])
        grad_weights_row = diff_row * (grad_x[:, :-1] - grad_x[:, 1:])
        diff_col = torch.sign(opt[:-1, :] - opt[1:, :])
        grad_weights_col = diff_col * (grad_x[:-1, :] - grad_x[1:, :])

        return grad_x, grad_weights_row, grad_weights_col

Figure 8.4: A pytorch function doing the forward and backward passes.
CONCLUSION AND FUTURE WORK

In this thesis, we have both introduced a class of models that can capture very complex behaviour by leveraging submodularity, and shown how to perform variational inference in them via divergence minimization. We have achieved this using the infinite Rényi divergences, which we have shown to have close connections to the continuous extensions, and the lower and upper polyhedra — objects that have been studied extensively for submodular functions.

As the first example, we considered the log-sub- and log-supermodular distributions. Having being often used to capture phenomena of cooperation, diversity and repulsion, they are the family that has been most studied in the existing literature. In this setting, we have also related the approximate inference problems to some classically studied problems, which has not only yielded computational benefits, but also gave us a better understanding of the computed approximations.

We then analyzed a generalization of these model models — the class of mixed submodular models. By assuming a factorization into submodular and supermodular terms, we showed how to leverage the methods for log-submodular and log-supermodular models to handle these models using dual decomposition. Moreover, we have shown the equivalence of this approach to the well-known expectation propagation framework.

The next family we considered are the cooperative graphical models. They utilize submodularity by applying the submodular function not on the variables themselves, but on a vector of boolean quantities computed from them. This has required more sophisticated inference techniques, as the model remains intractable even after linearizing the submodular function. Hence, as part of the inference methods we also to employ algorithms designed for inference in Markov random fields.

Finally, we considered the problem of learning these models. We focused on log-supermodular distributions, whose close connection to the min-norm problem let us formally analyze how the optimal approximate distribution changes as a function of the parameters of the submodular function. This let allowed us to learn models in a bi-level optimization setting, that also takes into account the approximation scheme used at prediction time.
9.1 **Future Work**

As the final section before we conclude the thesis, we would like to propose three possible directions for future work.

**Structured Approximations** In both Chapters 5 and 6 we have used only fully factorized models as approximations. While this might be sufficient for some tasks, it unfortunately does not capture any correlations between the variables. In Section 4.5 we have shown that if we approximate a MRF with a tree-structured distribution, we obtain bounds similar to those from TRWBP. It is thus natural to ask if we can also use such, or even more complex, approximations for some of the model classes we have studied. However, the resulting problems become significantly harder — e.g., if we minimize the inclusive infinite divergence between a log-supermodular model and a tree structured distribution, the inner problem changes from submodular minimization to the NP-hard problem studied by Kawahara, Iyer & Bilmes [136].

**Learning** In Chapter 8 we showed how to learn only log-supermodular distributions. As the minimization of the inclusive divergence is always a convex problem, even when the energy is non-submodular, a similar perturbation analysis might result in practical algorithms for learning an even larger family of discrete models.

**Continuous Models** Submodular functions can be also defined over arbitrary, possibly continuous, lattices, (see e.g. Bach [111] and Topkis [179]). For example, a twice differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is submodular iff its Hessian has only non-positive off-diagonal elements ([179, §3]). It would be interesting to understand if one can also derive useful models with continuous variables, and moreover, if we can again leverage submodularity in the design of approximate inference algorithms.


13. Zhang, J., Djolonga, J. & Krause, A. Higher-Order Inference for Multi-

Submodular Diversity Models Via Noise Contrastive Estimation in Inter-
ational Conference on Artificial Intelligence and Statistics (AISTATS)
(2016).


17. Flake, G. W., Tarjan, R. E. & Tsioutsiouliklis, K. Graph clustering and

18. Dughmi, S. Submodular functions: Extensions, distributions, and al-

19. Iyer, R. & Bilmes, J. Polyhedral aspects of Submodularity, Convexity

20. Bach, F. Learning with submodular functions: a convex optimization

(1972).


23. Agrawal, S., Ding, Y., Saberi, A. & Ye, Y. Correlation robust stochastic
optimization in Proceedings of the twenty-first annual ACM-SIAM sym-
posium on Discrete Algorithms (2010), 1087.


25. Kohli, P., Ladicky, L. & Torr, P. H. Robust higher order potentials for
enforcing label consistency in Computer Vision and Pattern Recognition,

26. Stobbe, P. & Krause, A. Efficient Minimization of Decomposable Submod-
ular Functions in NIPS (2010).


28. Fujishige, S. Polymatroidal dependence structure of a set of random

45. Fujishige, S., Hayashi, T. & Isotani, S. *The minimum-norm-point algorithm applied to submodular function minimization and linear programming* (Kyoto University. Research Institute for Mathematical Sciences [RIMS], 2006).


A.1 Convex Optimization

We will assume all convex functions take on values in the extended real line \( \mathbb{R} \cup \{\infty\} \). Moreover, all convex functions we use are proper (not equal to \( \infty \) everywhere), and closed as they are continuous and defined over closed domains.

**Definition A.1.** The convex conjugate \( f^* \) of a function \( f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\} \) is defined as
\[
  f^*(q) = \sup_{x \in \mathbb{R}^n} x^\top q - f(x). \tag{A.1}
\]

**Definition A.2.** The concave conjugate \( f^* \) of a function \( f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty\} \) is defined as
\[
  f^*(q) = \inf_{x \in \mathbb{R}^n} x^\top q - f(x). \tag{A.2}
\]

We will repeatedly use Fenchel’s duality derive dual problems, which will be useful for both computation and analysis.

**Theorem A.1** (Fenchel duality (Rockafellar [180])). If either \( f \) or \( g \) is continuous at a point where both functions are finite, then
\[
  \inf f(x) + g(-x) = \sup_{x} -g^*(x) - f^*(x).
\]

Moreover \( x^* \) is an minimizes \( f(x) + g(-x) \) iff \( G = \partial f(x^*) \cap \partial g^*(-x^*) \) is non-empty, in which case the elements of \( G \) maximize the supremum.

**Theorem A.2** (Rockafellar [126, Corollary 31.2.1]). Let \( f : \mathbb{R}^n \to \mathbb{R} \) and \( g : \mathbb{R}^m \to \mathbb{R} \) be proper convex functions, and let \( A : \mathbb{R}^n \to \mathbb{R}^m \) be a linear transformation. One has
\[
  \inf_{x} f(x) + g(Ax) = \sup_{y} -f^*(A^\top y) - g^*(-y)
\]
if one of the following conditions hold...
• There exists a point \( x \in \text{relint}(\text{dom}(f)) \) such that \( Ax \in \text{relint}(\text{dom}(g)) \).

• There exists a point \( y \in \text{relint}(\text{dom}(g^*)) \) such that \( A^\top y \in \text{relint}(\text{dom}(f^*)) \).

We will also make use of the following minimax theorem.

**Theorem A.3** (Rockafellar [126, Corollary 37.3.2]). Let \( C \) and \( D \) be non-empty closed sets in \( \mathbb{R}^m \) and \( \mathbb{R}^n \) respectively, and let \( f(x, y): C \times D \rightarrow \mathbb{R} \) be continuous finite concave-convex function. Then, if either \( C \) or \( D \) is bounded, one has

\[
\inf_{y \in D} \sup_{x \in C} f(x, y) = \sup_{x \in C} \inf_{y \in D} f(x, y). \tag{A.3}
\]

### A.2 Proof of Theorem 6.4

We will first define a dual formulation of the minimization problem, for which the claimed message passing scheme does BCD. We will define \( B = \bigotimes_{j=1}^m B(F_j) \), and for any \( y \) in either \( \Pi \) or \( B \) we denote by \( y_{ij} \) the coordinate corresponding to variable \( i \in V_j \) of the \( i \)-th block \((1 \leq i \leq m)\).

Because we consider the \( k \)-regular case, the primal can be written in the following simpler form

\[
\inf_{x \in \mathbb{R}^n} \sum_{j=1}^m f_j^L(x_{V_j}) + \frac{1}{2k} \|x_{V_j}\|^2,
\]

and the decomposed dual becomes the problem of finding the closest points between \( \Pi \) and \( B \).

\[
\sup_{(\pi_1, \ldots, \pi_m) \in \Pi} \sum_{j=1}^m \sup_{q_j \in B(F_j)} - \frac{k}{2} \|q_j - \pi_j\|^2. \tag{A.4}
\]

We now use exactly the same argument as by Nishihara, Jegelka & Jordan [146, §3.3] with some small changes necessary to accommodate our different definitions of \( \Pi \) and the polytopes. Please refer to that paper and references therein for the terminology used in the remaining of the proof. We will show that the Friedrich’s angle between any two faces of \( B \) and \( \Pi \) is at most \( \frac{2 \sqrt{n^2}}{k^2} \), which combined with [146, Theorem 2 and Corollary 5] implies the theorem. To make the notation easier to parse, we will assume that the elements in \( B \) and \( \Pi \) are ordered so that first come the \( |V_1| \) elements corresponding to \( F^1 \), then the \( |V_2| \) elements corresponding to \( F^2 \).
and so forth. Under this ordering, the vector space \( \Pi \) can be written as the nullspace of the following matrix

\[
S = \frac{1}{\sqrt{k}} \left( \begin{array}{c}
S_1 \\
\vdots \\
S_m
\end{array} \right),
\]

where \([S_j]_{i,i'} = \|i = i'\| \). (A.5)

As noted in [146], using [20, Proposition 4.7] we can express the affine hull \( \text{aff}_0(B_z) \) of any face \( B_z \) as (where for each \( j \in \{1, \ldots, m\} \) the sets \( A_{r,1}, \ldots, A_{r,M_r} \) form a partition of \( V_j \))

\[
\text{aff}_0(B_z) = \bigcap_{r=1}^{m} \bigcap_{m=1}^{M_r} \{(q_1, \ldots, q_m) : q_r(A_{r,1} \cup \ldots \cup A_{r,m}) = 0 \}.
\]

This set can be also written as the nullspace of the following matrix

\[
T = \left( \begin{array}{c}
\frac{1^{T}_{A_{1,1}}}{\sqrt{|A_{1,1}|}} \\
\frac{1^{T}_{A_{1,2}}}{\sqrt{|A_{1,2}|}} \\
\vdots \\
\frac{1^{T}_{A_{1,M_1}}}{\sqrt{|A_{1,M_1}|}} \\
\frac{1^{T}_{A_{m,1}}}{\sqrt{|A_{m,1}|}} \\
\frac{1^{T}_{A_{m,2}}}{\sqrt{|A_{m,2}|}} \\
\vdots \\
\frac{1^{T}_{A_{m,M_m}}}{\sqrt{|A_{m,M_m}|}}
\end{array} \right).
\]

To compute the Friedrich’s angle we are interested in the singular values of \( ST^T \) [146][Lemma 6], which is equal to

\[
ST^T = \frac{1}{\sqrt{k}} \left( \begin{array}{c}
\frac{1_{A_{1,1}}}{\sqrt{|A_{1,1}|}} \\
\frac{1_{A_{1,M_1}}}{\sqrt{|A_{1,M_1}|}} \\
\vdots \\
\frac{1_{A_{m,1}}}{\sqrt{|A_{m,1}|}} \\
\frac{1_{A_{m,2}}}{\sqrt{|A_{m,2}|}} \\
\vdots \\
\frac{1_{A_{m,M_m}}}{\sqrt{|A_{m,M_m}|}}
\end{array} \right).
\]

Hence, we have to analyze the eigenvalues of the square matrix \((ST^T)^T (ST^T)\), whose rows and columns are indexed by

\[
\mathcal{I} = \{(j,l) : j \in \{1,2,\ldots,m\}, l \in \{1,2,\ldots,M_j\}\}.
\]
and whose \(((j, l), (j', l'))\) entry is
\[
\frac{1}{k} \frac{|A_{j,l} \cap A_{j',l'}|}{\sqrt{|A_{j,l}| |A_{j',l'}|}}.
\]

We create a graph with vertices \(I\) and we add edges between distinct \((j, l)\) and \((j', l')\) with weight \(|A_{j,l} \cap A_{j',l'}|\) (zero weight means that we do not add that edge). The normalized graph Laplacian of this graph is equal to
\[
[L]_{(j,l),(j',l')} = \begin{cases} 
1 & \text{if } (j, l) = (j', l'), \\
-\frac{1}{k-1} \frac{|A_{j,l} \cap A_{j',l'}|}{\sqrt{|A_{j,l}| |A_{j',l'}|}} & \text{otherwise}.
\end{cases}
\]

Hence, \((ST^T)^T(ST^T) = I - \frac{k-1}{k} \mathcal{L}\). We want to lower-bound the Cheeger constant \(h\) of this graph. Because the spectrum of a graph is the union of the spectra of the connected components we will assume that the graph is connected, as we can apply the same argument to every component. From the definition of \(h\) we have that
\[
h \geq 2 \frac{\text{minimum cut volume}}{\text{volume}}, \text{where volume} = \sum_{i \in V} |\delta(i)|(|\delta(i)| - 1) = |V|(k^2 - k).
\]

What remains is to bound the minimum cut from below. Because the graph is connected, for any cut \(U\) there must exist some \(v\) that is in sets on both sides of the cut. Let \(a\) be the number of sets in \(U\) that contain it, and let \(k - a\) be the number of sets in the complement that contain it. Then, the cut is of size is at least \(a(k - a) \geq k - 1\). Hence
\[
h \geq 2 \frac{k - 1}{|V|(k^2 - k)} = \frac{2}{|V|k}.
\]

And by Cheeger’s inequality the smallest positive eigenvalue of the Laplacian \(\mathcal{L}\) is at least \(\frac{2}{|V|^2 k^2}\), which from the relationship \((ST^T)^T(ST^T) = I - \frac{k-1}{k} \mathcal{L}\) implies that the squared Friedrich’s angle \(c_F^2\) between \(\Pi\) and \(B\) is at most
\[
c_F^2 \leq c_F^2 = 1 - \frac{k - 1}{k} \pi_2 \leq 1 - \frac{1}{|V|^2 k^2},
\]
which completes the proof.