Active Learning for Source Localization

Master Thesis
Victor Cărbune
April 21, 2014

Supervisors: Yuxin Chen
Dr. Amin Karbasi
Prof. Dr. Andreas Krause

Learning and Adaptive Systems Group
Department of Computer Science, ETH Zürich
Abstract

We propose a novel approach through which we identify the origin of an instance of a diffusion process: disease outbreaks, rumor spreading over social networks, news across the internet or viruses infecting computers within a network.

The model we built creates a synthetic search space by generating cascades according to various parameters from each of the network nodes.

We make use of active learning techniques, such as the EC\textsuperscript{2} or GBS adaptive submodular objective functions, to adaptively select observer nodes and downweight inconsistent hypothesis with the observations.

We thoroughly evaluate the variability of our novel model and compare its performance with the Gaussian maximum likelihood estimator, an alternative solution existing in the research literature.

We further improve the existing solution by showing a performance increase when used in combination with our adaptive placement strategy of observers.

The performance analysis is conducted on a wide variety of situations, generated through synthetic diffusion models and network topologies.
Acknowledgements

I am grateful to Prof. Andreas Krause for giving me the opportunity of pursuing the master thesis within the Learning and Adaptive Systems group, for investing his time to regularly follow up with the progress of the thesis and provide valuable remarks.

I have been extremely lucky to have Yuxin Chen and Dr. Amin Karbasi as direct supervisors, who continuously helped me push the limits of the approach through their vast knowledge within the field of networks and active learning.

I am thankful to Dr. Olivier Byrde and his group, ID HPC, Scientific IT Services, for all the computational resources and support I received as part of the EULER cluster beta program.

I would like to thank my parents, Lucia and Viluţ, and my sister, Maria, for their endless positive energy and support.

I’m very grateful to Sinziana for always being by my side throughout the master’s degree.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Motivation and challenges</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>Contributions</td>
<td>4</td>
</tr>
<tr>
<td><strong>2</strong></td>
<td>Background work</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Source estimator as Gaussian MLE</td>
<td>5</td>
</tr>
<tr>
<td>2.1.1</td>
<td>Result 1: optimal estimator for general trees</td>
<td>6</td>
</tr>
<tr>
<td>2.1.2</td>
<td>Result 2: generalization for arbitrary graphs</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Bayesian Active Learning</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1</td>
<td>Description and notations</td>
<td>8</td>
</tr>
<tr>
<td>2.2.2</td>
<td>Set functions: submodularity</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3</td>
<td>Objective functions: GBS, EC2, VOI</td>
<td>10</td>
</tr>
<tr>
<td><strong>3</strong></td>
<td>Source Estimation using Active Learning</td>
<td>15</td>
</tr>
<tr>
<td>3.1</td>
<td>Preliminaries</td>
<td>16</td>
</tr>
<tr>
<td>3.2</td>
<td>The version space: artificial diffusions</td>
<td>18</td>
</tr>
<tr>
<td>3.3</td>
<td>Reducing the version space: observations</td>
<td>19</td>
</tr>
<tr>
<td>3.4</td>
<td>Accomodating noisy sensors</td>
<td>23</td>
</tr>
<tr>
<td><strong>4</strong></td>
<td>Experimental Results</td>
<td>27</td>
</tr>
<tr>
<td>4.1</td>
<td>Synthetic models and metrics</td>
<td>27</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Network models</td>
<td>27</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Diffusion models</td>
<td>29</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Metrics: hops, rank and nDCG</td>
<td>30</td>
</tr>
<tr>
<td>4.2</td>
<td>Key results</td>
<td>31</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Results I: Known diffusion parameters</td>
<td>32</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Results II: Searching for the unknown</td>
<td>46</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Results III: Variability of the model</td>
<td>51</td>
</tr>
</tbody>
</table>
4.3 Engineering overview ........................................ 55
  4.3.1 Frameworks: SNAP, Eigen and Open MPI ........ 56
  4.3.2 Distributing lazy greedy computations ............. 56

5 Conclusion ..................................................... 59

A Appendix ...................................................... 61
  A.1 Reproducing Results .................................... 61

Bibliography .................................................... 69
Chapter 1

Introduction

Numerous efforts have been required to study the diffusion of various processes, either natural, such as the outbreak of an epidemia or artificial, such as the information spreading over the internet.

More recently, a grown interest has been expressed towards studying the reverse process of identifying the origin of a given diffusion instance, which represents the main topic of the thesis.

Determining the theoretical model of the direct process and the experimental validation is a costly and lengthy process that needs to be customized individually and is highly dependent on external factors.

Therefore, one of the goals of the thesis is to build a model that is able to obtain satisfiable results, even if the theoretical diffusion model is unknown or differs significantly from the true underlying model that has generated the diffusion instance.

In our approach of solving the problem, we employ active learning techniques using several different objective functions, such as $EC^2$ presented in [8], to place observers adaptively and reveal the most valuable information about the diffusion instance, while minimizing the overall cost.

An alternative solution is provided in [16] and further presented in Chapter 2. Throughout the thesis, we will further examine the performance of this solution in comparison to our model and propose an extension for a different approach of placing observers throughout the network.

In the following section, we further present an instance of the problem and discuss several aspects of the key challenges encountered when solving it.
1. Introduction

1.1 Motivation and challenges

The problem of reliably locating the source of a diffusion has many applications in various circumstances.

One common example is the case when a bacteria spreads due to an infection site that is frequently visited by persons in a village and then carried over to other village through movement of individuals. In such a case, it is extremely important to identify the source of infection fast and reliably.

In Figure 1.1 we illustrate two different diffusion instances, with different sources, on top of their common underlying network structure. We defer a formal presentation of the notations and models to Section 3.1.

Despite its simplicity, the example is able to uncover some of the difficulties one encounters when trying to identify the source of an infection or information spread: one can immediately identify that one infection has completely infected all the nodes, while the other only a subset.

Another easy to spot difference is that in the first diffusion the source node is a large hub, with a high degree, while in the other one the source is almost isolated; however, despite its isolation, it manages to infect the whole network.

In order to have a complete overview of the performance of a solution, we believe that the following aspects of the diffusion instance need to be taken into consideration:

- **Topology of the underlying network.** As it naturally arises, the topology has a significant impact on the performance of a given solution. Therefore, we dedicate Section 4.1 to present the main characteristics of the three topologies considered in our experiments

- **Prior knowledge about the diffusion model.** Depending on how long the natural process has been studied, theoretical models of the diffu-
1.1. Motivation and challenges

The diffusion process might exist and ideally, the reverse process, of identifying the source, should take it into consideration.

However, we believe that creating a different reverse model for each existing diffusion model is impossible, due to the variety and variability of the natural phenomena that can generate and influence the diffusion.

Therefore, we are seeking to understand how well our model performs under unknown circumstances.

• The size of the spread within the network. The diffusion might not have affected all nodes, due to its underlying process or simply due to the fact that the algorithm is identifying the source while the infection is still spreading.

If the spread only covers a small percentage of the complete network, tracing the source of the process becomes more difficult: some nodes may or may not be infected and placing an observer might or might not yield relevant information.

The model that we propose is able to efficiently make use of the information that a node is not infected in order to further reduce the search space.

• Which observers are used. The number of observers that can be used is limited and certain restrictions may apply in some situations.

The performance of an estimation strategy is dependent on which observers are selected in order to gain information about the diffusion instance.

Through the experimental setup, we compare the performance of the solutions using adaptive strategies of choosing a subset of observers, as well as fixed policies, such as highest-degree nodes or random.
1.2 Contributions

Throughout the thesis, we analyze and discuss the performance of the solutions with respect to the previous criteria. Our contributions can be summarized as follows:

- **Novel Solution.** We present a novel approach to the problem, in a variety of experimental settings, presenting both its advantages and disadvantages.

- **Observer Placement Techniques.** We present an adaptive technique of selecting observers, using $EC^2$ and GBS as adaptive submodular objective functions.

- **Analysis of Gaussian MLE.** We implement and analyse the performance of the solution presented in [16]. We discuss its advantages and limitations, usually in comparison with the novel approach proposed.

- **Extension of Gaussian MLE.** We further extend the maximum likelihood estimator by making use of adaptive policies for selecting observers, the same as in our model.
In this section of the thesis we will present the foundational theoretical layer of the two solutions that we thoroughly analyze and evaluate experimentally throughout the thesis.

In Section 2.1 we present the theoretical results of Pedro Pinto, Patrick Theran and Martin Vetterli, in their paper “Locating the Source of Diffusion in Large Scale Networks” [16], along with several remarks.

In Section 2.2 we introduce adaptive submodular functions and further present theoretical guarantees of the active learning approach, provided by Daniel Golovin, Andreas Krause and Debajyoti Ray in the paper “Near-optimal Bayesian Active Learning with Noisy Observations” [8].

Through Section 2.2 we provide a preliminary introduction to Chapter 3, where we map the problem of network source localization to the Bayesian active learning setting.

2.1 Source estimator as Gaussian MLE

We start the summary by introducing the model of the diffusion process assumed by the authors: at any given time \( t \), all the vertices in the graph can be in one of the following two states:

- **informed**: the node has already been informed (infected) from a neighbour or it is directly the source.
- **ignorant**: the node has not been not been informed (infected) yet.

Let \( G = (V, E) \) denote the underlying network topology, where \( V \) denotes the set of vertices and \( E \) the set of edges.

The authors further denote \( V(u) \) to be the adjacent vertices of \( u \). Let \( t_u \) be the time when node \( u \) becomes informed, because one of its neighbours
transmits the information for the first time to it.

Let \( w \in \mathcal{V}(u) \) be the node from which \( u \) has received the information. Then, each of its neighbours \( v \in \mathcal{V}(u) \setminus w \) will receive the information (become infected) at time \( t_u + \theta_{uv} \), with \( \theta_{uv} \) denoting the random propagation delay associated with edge \( e = (u, v) \in E \). The random variables \( \{\theta_{uv}\} \) have a known, arbitrary joint distribution. The process starts from a source node, \( s \in V \), at an unknown time.

The next step is to establish, under some criteria, the set of nodes where observers will be placed to do measurements. The authors denote the set of \( K \) observers with \( \mathcal{O} = \{o_k\}_{k=1}^K \subset V \).

The authors design an estimator \( \hat{s}(-) \), such that the localization probability
\[
P_{\text{loc}} = P(\hat{s}(\mathcal{O}) = s^*)
\]
where \( s^* \) is the source in the diffusion instance.

The goal of the estimator is to be able to take into account the particularities of the network topology, along with the observed infection times from the placed observers and obtain an accurate estimation of the source location, with theoretical guarantees.

The authors first assume the situation of a general tree network topology, generalizing their results to general network topologies afterwards.

Therefore, let \( T_a \subset G \) be the active subtree of observers \( O_a = \{o_k\}_{k=1}^{K_a} \) that end up receiving the information from the source node.

When an observer is placed, the following information is assumed to be revealed about the ground truth:

- **time**: the time at which the node received it \( \{t_k\}_{k=1}^{K_a} \)
- **direction**: the neighbour from which the information originates, thus imposing an order within the subset \( O_a \)

The authors further proceed to denote the active edges set \( E(T_a) = \{1, 2, ..., E_a\} \) and model the associated propagation delays as random variables, \( \theta_e \) for each \( e \in E \).

The authors further continue with the assumption that these propagation delays are I.I.D. random variables with Gaussian distribution \( \mathcal{N}(\mu, \sigma^2) \), with its moments, mean \( \mu \) and variance \( \sigma^2 \), known.

### 2.1.1 Result 1: optimal estimator for general trees

\[
\hat{s} = \arg\max_{s \in T_a} \mu_s^T \Lambda^{-1}(d - \frac{1}{2} \mu_s)
\]

The notations represent the following:
2.1. Source estimator as Gaussian MLE

- **Observed Delay**: $d_k = t_{k+1} - t_1$
- **Deterministic Delay**: $[\mu_s]_k = \mu(|P(s,o_{k+1}) - |P(s,o_1)|)$
- **Delay Covariance**: $[\lambda]_{k,i} = \sigma^2 \left\{ \begin{array}{ll} |P(o_1,o_{k+1})|, & k = i \\ |P(o_1,o_{k+1}) \cap P(o_1,o_{i+1})|, & k \neq i \end{array} \right.$

In all of the above terms, we have that $k,i = 1,K_a - 1$

The proof of the near-optimality of the estimator and its associated $O(N)$ complexity can be found in the main paper [16] and its supplemental material.

The authors offer further comments about the approach that give an insight into the performance of the estimator:

- **Observers**: should be sparsely positioned.
- **Propagation delays**: should have finite moments.

The authors provide the following generalization to arbitrary graphs.

### 2.1.2 Result 2: generalization for arbitrary graphs

In order to apply the estimator on arbitrary graphs $G = \{V, E\}$ instead of general trees the assumption is that the diffusion tree rooted at a vertex $s \in V$ is the breadth-first search tree $T_{bfs,s}$.

$$\hat{s} = \arg\max_{s \in V} S(s,d,T_{bfs,s}),$$

where $S = \mu_s^T \lambda_s^{-1} (d - \frac{1}{2} \mu_s)$, with the parameters computed with respect to the bread-first search tree rooted at $s$. The complexity of the approach is proved to be $O(N^3)$.

**Remarks**

Even though the authors mention that the Gaussian assumption can be relaxed, we couldn’t find a generalization method from estimating the moments of the distribution directly from the observations.

Our comment is that the assumption is not easily fulfilled in practice, and not having a estimation of the parameters from observations implies a requirement for a preliminary thorough study of the diffusion process in order to obtain approximate estimations of moments of the Gaussian distribution.

One example where such a thorough study has been done is the case-study of the application of the model on the KwaZulu Outbreak provided by the
authors. The moments of the distribution are estimated based on the parameters of the cholera validated through a previous research, which focused on understanding the particularities of the diffusion model of cholera [2].

Aside from the thorough studies involved, many diffusion processes do not necessarily exhibit the property of having Gaussian-distributed random propagation delays.

2.2 Bayesian Active Learning

In the active learning approach, given a set of unlabeled data, the learning algorithm can request labels for particular data, in batches or one by one. The end-goal is to minimize the cost of labeling these subsets of data, while maximizing the performance of the learning algorithm.

The model we built is a natural extension for networks of the Bayesian active learning approach, for which we briefly present here some of the theoretical results obtained in [8].

2.2.1 Description and notations

Given a set of hypothesis \( \mathcal{H} = \{ h_1, h_2, ..., h_n \} \), weighted according to distribution \( P \), in Bayesian active learning we want to distinguish among them by performing a set of tests \( \mathcal{T} = \{ 1, ..., N \} \). Each test has an associated cost of \( c(t) \) and produces an outcome from a finite set \( \mathcal{X} = \{ 1, 2, ..., l \} \). The outcome of each test is modeled by a random variable \( X_t \) taking values in \( \mathcal{X} \).

The goal is to determine an adaptive policy of selecting tests that allows the determination of the true underlying hypothesis \( H \), while minimizing the total cost. Formally, a policy \( \pi \) is a partial mapping from a set of previous observations \( x_A \) to tests, specifying which test to run next.

If we denote with \( \nu(x_A) = \{ h \in \mathcal{H} : P(h|x_A > 0) \} \), then a policy \( \pi \) is called feasible, if upon termination with observations \( x_A \), \( |\nu(x_A)| = 1 \).

Further, the expected cost of a policy \( \pi \) is defined through

\[
c(\pi) = \sum_{h \in \mathcal{H}} P(h) c(\mathcal{T}(\pi, h))
\]

where \( \mathcal{T}(\pi, h) \subseteq \mathcal{T} \) is the set of tests run by policy \( \pi \) in case \( H = h \).

Thus, the goal is to find the feasible policy \( \pi^* \) of minimum expected cost: \( \pi^* = \text{argmin}\{c(\pi) : \pi \text{ is feasible}\} \). Approximating a policy \( \pi \) for which \( c(\pi) \leq c(\pi^*) o(\log(n)) \) has been shown in [3] to be NP-hard.

\(^1\)Noiseless setting, where there exists hypothesis \( h^* \) that agrees with outcomes of all tests
Therefore, some of the approximations are heuristics that select tests greedily according to an objective function, defined on the test set and hypothesis space.

The role of a heuristic function is to compute the marginal gain of a test \( t \in T \), given the set of previous observations \( x_A \), \( \Delta_{Alg}(t|x_A) \).

Thus, the optimal test \( t^* \) to select will then be:

\[
t^* = \arg\max_{t \in T} \Delta_{Alg}(t|x_A)/c(t)
\]

We defer presenting the exact objective functions to 2.2.3, and we proceed by briefly presenting a strong particular structural property of set functions that provides the theoretical guarantees of the heuristic functions used for maximizing greedily the objective function.

### 2.2.2 Set functions: submodularity

The problem of source localization through observer placement in the network shares the common goal of a sensing optimization problem, obtaining the most useful information about the infection, while using the minimum number of sensors.

Extensive work has been done to study the applications of submodularity in such problems [12]. Intuitively, submodular functions exhibit the property of having diminishing returns. In our specific setting, obtaining information about the infection after placing the first observer on the network yields more valuable information than the one obtained after placing the last observer, when almost the entire infection state has been revealed.

Formally, given a set \( S = \{1, 2, ..., n\} \), let \( F : 2^S \to \mathbb{R} \) be a set function defined on the elements of \( S \). Let \( \Delta_F(s|A) = F(\{s\} \cup A) - F(A) \) denote the marginal gain of adding element \( s \in S \) to the set \( A \subset S \).

\( F \) is submodular if the following inequality holds:

\[
\Delta_F(s|A) > \Delta_F(s|B), \forall A \subset B \subset S
\]

Submodular functions are encountered in many natural problems, such as cut functions in graphs and hypergraphs, functions measuring the population protected by sensing in a water distribution system [13], information gain for conditionally independent observations [11] and many others.

Submodularity can be seen as a discrete equivalence of concavity [10]. Several optimization problems can be transformed in an optimization problem.
2. Background work

with a submodular objective function, including the problem solved through this thesis.

\( F \) is considered to be monotonic submodular if the value of the function increases as the set size increases, that is \( F(A) \leq F(B), \forall A \subset B \subset S. \)

For mononotonic submodular functions, the simple greedy optimization algorithm has been proven to be within the NP-hard optimum by a factor of \((1 - \frac{1}{e})\) [15].

The algorithms that employ characteristics of submodular set functions previously described are extremely powerful in selecting a set of fixed \( k \) elements, in a non-adaptive manner. In the current problem of source localization, we wish to adaptively select observers: that is, place the next observer based on the information received from the previous ones.

In such cases, the notion of adaptive submodularity preserves many of the theoretical guarantees of the classical submodular functions, and can be used for building greedy adaptive policies with the same theoretical guarantees with respect to the optimal policy [7].

2.2.3 Objective functions: GBS, EC2, VOI

In the adaptive setting, the outcome of the test is revealed after each selection and is taken into consideration when computing the scores of the remaining tests.

Therefore, the natural approach of extending submodular functions to adaptive policies is to consider the actual realization of the items already selected and compute the expected outcome of the items to be considered, over a given distribution.

Considering the the Bayesian active learning setting, let \( P(H, X_1, ..., X_N) \) denote the joint distribution over the hypothesis and test outcomes. Depending on whether the tests are noisy or not, the joint distribution is deterministic or not given \( H \).

We denote through \( x_A \preceq x_B \), the case when for any \( A \subseteq B \) it holds that \( P(x_B|x_A) > 0. \)

Formally, a function \( F : 2^T \times H \rightarrow \mathbb{R} \) is adaptive submodular w.r.t. to a joint distribution \( P \) over the test outcomes and the set of hypothesis, if for any \( x_A \preceq x_B \) and any test \( t \) it holds that \( \Delta(t|x_A) \geq \Delta(t|x_B) \), where

\[
\Delta(t|x_A) = \mathbb{E}_H[F(A \cup \{t\}, H) - F(A,H)|x_A]
\]

In the remaining parts of this section we present the objective functions that we used for exploring the performance of the model proposed in Chapter 3.
Among these, GBS and $EC^2$ are of particular interest, as they are monotonic adaptive submodular and theoretical guarantees exists for the performance of the greedy policy using them.

**Generalized Binary Search (GBS)**

The generalized binary search score function is the expected reduction in the version space probability mass and is formally defined as follows (the expected marginal gain of item $t$ given $x_A$):

$$\Delta_{GBS}(t|x_A) = P(v(x_A)) - \sum_{x \in \mathcal{X}} P(X_t = x|X_A)P(v(x_A, X_t = x))$$

As noted in [8], due to the adaptive submodular property and strong adaptive monotonicity of the underlying function, it has been shown to obtain near-optimal cost [7, 9]:

$$f_{GBS}(S, h) = 1 - P(v(x_S(h))) + P(h)$$

$$c(\pi_{GBS}) \leq (c(\pi^*) \ln(1/p_{min}) + 1),$$

where $p_{min} = \min_{h \in \mathcal{H}} P(h)$ and $\pi^*$ the optimal policy.

Another score function of a test which has been shown to be equivalent to GBS in the noiseless case is information gain, $P$ is deterministic given $\mathcal{H}$ [17].

$$\Delta_{IG}(t|x_A) = H(X_T|x_A) - \mathbb{E}_{x_t|x_A}[H(X_T|x_A, x_t)]$$

**Equivalence Class Edge Cutting ($EC^2$)**

The noisy version of Bayesian active learning is a more realistic setting, admitting that sensors may produce faulty outcomes.

In this case, the noisy active learning problem can be reduced to the noiseless setting by artificially creating noise in the hypothesis space by flipping the outcome of one of the $N$ tests. Each new noisy hypothesis generated in this way has the prior $P'(h') = P(h)/N$, where $h'$ is a noisy copy of $h$ out of total $N$ such copies.

Through this modification of assuming the noise in the hypothesis space, the same guarantees hold with the joint distribution conditioned on hypothesis being deterministic, but the version space can grow significantly larger.

Formally, for each original hypothesis $h_i \in \mathcal{H}$, there now exists an equivalence class $\mathcal{H}_i$, which contains all the noisy copies of the original hypothesis.

The transformed problem is commonly referred as the equivalence class determination, where the set of hypothesis $\mathcal{H}$ is partitioned into a set of $m$ equivalence classes $\{\mathcal{H}_1, ..., \mathcal{H}_m\}$, such that $\mathcal{H} = \bigcup_{i=1}^{m} \mathcal{H}_i$. 
The end goal can be similarly expressed as in the noiseless case, with the difference that, upon termination, the remaining hypothesis should belong to the same class $H_i$.

The $EC^2$ algorithm has been developed in order to overcome some of the performance issues that GBS and IG face when dealing with the problem of differentiating better among classes $H_i$ [8].

$EC^2$ extends the idea of introducing edges between hypotheses [4], but instead of introducing edges between random arbitrary pairs of hypothesis, they are introduced among hypothesis from different classes.

Let $E = \bigcup_{1 \leq i \leq j \leq m} \{\{h, h'\} : h \in H_i, h' \in H_j\}$ the set of edges among hypotheses from different classes. Intuitively, these are the edges that need to be cut, in the sense that among any edge between two hypotheses, one or both of the hypotheses must be eliminated from the version space.

If we define the weight of an edge to be $w : E \to \mathbb{R} \geq 0$ with $w(\{h, h'\}) = P(h) \cdot P(h')$ and extend it to an additive modular function $w(E') = \sum_{e \in E} w(e)$, then objective $f_{EC}$ to be greedily maximized is defined as the weight of the edges cut:

$$f_{EC}(A, h) = w(\bigcup_{t \in A} E_t(h)),$$

where $E_t(h) = \{\{h', h''\} : h'(t) \neq h(t) \lor h''(t) \neq h(t)\}$, are the edges cut by test $t$ under true hypothesis $h$.

By using the structural similarities as with the GBS objective, it is proved that the $f_{EC}$ objective function is adaptive submodular [8]. The marginal gain $\Delta_{EC}(t|x_A)$ can be used to greedily select the next test:

$$\Delta_{EC}(t|x_A) = E_H[f_{EC}(A \cup \{t\}, H) - f_{EC}(A, H)|x_A]$$

The following theoretical guarantee is provided for the adaptive greedy policy $\pi_{EC}$ implemented by $EC^2$:

$$c(\pi_{EC}) \leq (2 \ln \frac{1}{p_{\min}} + 1) c(\pi^*),$$

where $p_{\min} = \min_{h \in H} P(h)$ and $\pi^*$ being the optimal policy for the equivalence class determination problem.

In practice, however, it is impossible to simulate the entire noisy hypothesis search space and set the problem to the deterministic equivalence class determination instance.
Therefore, the authors introduce another random variable, θ to take into account the noisy outcomes of the sensors, as depicted in the graphical model in Figure 2.1.

In the more realistic setting, and the one on top of which we built the model in Chapter 3, we do not expect that upon termination with the observations \( x_T \) to have no uncertainty over the true hypothesis. Therefore, it is possible that in the posterior distribution \( P(H|x_T) \), computed using Bayes rule, we can obtain more than one non-zero hypothesis class. The goal transforms into gathering data to make effective decisions \( d \in \mathcal{D} \) such that each time a loss function \( l(d,h) \) is minimized, where \( \mathcal{D} \) is the set of possible decisions we may make.

We refer the interested reader to further look into \([8]\) for more detailed explanations and proofs and also read the following chapter for the exact mapping of this model to the problem studied by the thesis. Finally, we state the theoretical guarantee important for our model:

**Theorem 2.1** Fix hypothesis \( \mathcal{H} \), tests \( \mathcal{T} \) with costs \( c(t) \) and outcomes in \( \mathcal{X} \), decision set \( \mathcal{D} \) and loss function \( l \). Fix a prior \( P(H,\theta) \) and a function \( x_T : \mathcal{H} \times \text{supp}(\theta) \rightarrow \mathcal{X}^N \) which define the probabilistic noise model. Let \( c(\pi) \) denote the expected cost of \( \pi \) while incurring to find the best decision, i.e. to identify which equivalence class \( \mathcal{H}_d \) the outcome vector \( x_T \) belongs to. Let \( \pi^* \) denote the policy minimizing \( c(\cdot) \) and let \( \pi_{\text{EC}} \) denote the adaptive policy implemented by EC\(^2\). Then it holds that

\[
c(\pi_{\text{EC}}) \leq 2 \ln\left(\frac{1}{p_{\text{min}}'} + 1\right)c(\pi^*),
\]

where \( p_{\text{min}}' = \min_{h \in \mathcal{H}} \{P(h,\theta) : P(h,\theta) > 0\} \)

**Value of Information (VoI)**

The last objective functions that we made use of is the value of information, which is a common approach to the decision problem described above. As noted in \([8]\), this objective function myopically picks tests maximizing:

\[
\Delta_{\text{Vol}}(t|x_A) = \min_{d \in \mathcal{D}} \mathbb{E}_H[l(d,H)|x_A] - \mathbb{E}_{x_i \sim x_i|x_A} \left[ \min_{d \in \mathcal{D}} \mathbb{E}_H[l(d,H)|x_A, x_i] \right]
\]
2. Background work

Intuitively, VoI represents the expected reduction in the expected loss of the best decision due to the observation of $x_t$. 
In the novel approach we propose, we treat the process of identifying the source as an active learning problem, in which information is incrementally revealed about the diffusion instance, through the placement of observers.

The method we built creates an artificial version space on top of the known network topology. The version space consists of clouds of possible diffusion instances, starting from each vertex of the network.

Further, by observing information about the ground truth in a selected subset of the network nodes, we alter the search space according to the observation. Afterwards, we select another node from which to gather information, by maximizing the expected marginal gain with respect to a particular objective function ($EC^2$, $GBS$, $VoI$), which reveals how much information we gain, in expectation over the possible outcomes, by observing the specific node.

In a minimal, step by step, graphical illustration, the process can be depicted as in the following figures:

As with any active learning problem, the first task is to identify a method of scoring how informative and valuable is a test, if labeled. In our case, this means understanding how valuable, at a given state, a specific observation of a node is.

In the following sections we will introduce the mapping of the problem of source identification to the Bayesian Active Learning setting presented in Section 2.2.
3. Source Estimation using Active Learning

Figure 3.1: Initial. The network topology is known.

Figure 3.2: Query 1. The queried node is infected.

Figure 3.3: Query 2. The queried node is clean.

3.1 Preliminaries

As previously, throughout the chapter we will use the term infection, but the reader should not restrict the notion or applicability of the solution solely to disease outbreaks, rather the term should be seen as being interchangeable with rumor spreading within a social network, information dissemination
across the internet through websites and blogs or internet virus spreading within a computer network.

We formally establish the **network model**, on top of which we implement our solution and on top of which the diffusion process occurs, as follows: an undirected graph $G = \{V, E\}$, where $V$ denotes the set of vertices of the graph, and $E$ denotes the set of edges. We denote the number of vertices with $N = |V|$ and the number of edges with $M = |E|$.

The **diffusion process model** describes how the infection affects the neighbouring nodes, starting from a particular source node $s \in V$. We defer the formal description of the diffusion processes used to Section 4.1.

However, the **diffusion instance** can be defined regardless of the diffusion process, and it represents a particular realization of the diffusion process. Since the diffusion process can be seen as a graph traversal according to particular rules, the result of the process is a diffusion tree $T_G = \{V_{T_G}, E_{T_G}\} \subset G$, where $E_{T_G}$ are the forward edges, which are used for infecting nodes for the first time, and $V_{T_G}$ is the subset of vertices that were infected.

Let $t_v \in \mathbb{R}_{\geq 0}$ denote the infection time of vertex $v \in V$. Then, $\forall u \in T_V$, and $\forall (u, v) = e \in T_E$, there exists an associated travel time, $\tau_{e} \in \mathbb{R}_{\geq 0}$, such that $t_u = t_v + \tau_{(u,v)}$

Finally, one can represent the diffusion instance simply by the set of pairs of infected nodes and their infection time: $\{(v, t_v)|v \in V, t_v \in \mathbb{R}_{\geq 0}\}$ and further infer the diffusion tree, if needed.

As it is probably already foreseeable by now for the reader, in most real-world scenarios, the resulting diffusion instances cover a small fraction of the network nodes.

We further denote by **ground truth** the realization of the diffusion process for which we are trying to identify the source. Nothing is known about the ground truth at the beginning, and information is revealed through a set of **observers** that are placed throughout the network.

Depending on the placement strategy, information can be gradually revealed by placing observers adaptively or in a single batch, deciding beforehand their precise position. The two methods can also be combined, by adaptively placing batches of observers, but we defer this analysis for future work.

In the active learning setting we define a **test outcome** to be the information revealed by an observer situated at a particular node within the network.

Depending on the model used, the information revealed can be a boolean value, whether the node was infected or not, or the actual infection time.
3. Source Estimation using Active Learning

3.2 The version space: artificial diffusions

The initialization step of our active learning algorithm creates a cloud of possible diffusion instances associated with each node in the network. The instance cloud of a specific node represents our hypothesis search space, should the true diffusion instance start from the respective node.

Each cloud surrounding the specific node represents a hypothesis class. The class label represents the source of the ground truth, and for each cloud is the node from where all the hypotheses start.

Through Figure 3.4 we provide a graphical illustration of how a simple equivalence class looks like.

Given the arbitrary graph $G = \{V, E\}$, and $v \in V$ a vertex in the graph, let $C_v$ represent the cloud of diffusion instances starting from node $v$:

$$C_v = \{h_1, ..., h_n\} \forall v \in V, n \in \mathbb{N},$$

where $n \in \mathbb{N}$ denotes the number of hypotheses artificially generated for each node and is considered to be a fixed parameter of the solution in our experimental setup.

The mapping to the Bayesian active learning framework presented in Section 2.2 is straightforward. The equivalence classes among which the algorithm has to differentiate are $\mathcal{H} = \{ \cup_{v \in V} C_v \}$. 

Figure 3.4: Sample equivalence class for node $v \in V$, $C_v = \{h_1, h_2\}$
Let $S$ denote the random variable of the true source, taking values in the set of vertices $V$. We consider the prior probability to be uniform over the set of vertices in the graph. That is, $P(S = v) = \frac{1}{|V|}, \forall v \in V$.

Since all hypotheses within a cluster have equal importance and a cluster $C_v$ represents the equivalence classes of the case when the source node is the vertex $v$, it follows naturally that the prior probability mass of $h \in C_v$ is

$$P(h) = \frac{1}{n|V|}, \forall h \in C_v, \forall v \in V$$

It is worth noting that the theoretical cost guarantee provided by Theorem 2.1 is directly dependent on the size of the cluster. The more artificial hypothesis we consider, the larger the margin within which the optimal cost is obtained.

### 3.3 Reducing the version space: observations

Let $T = \{T_v | v \in V\}$ denote the set of possible tests. As previously described, each test represents a request for information about the infection state of a specific node in the graph. For simplicity, we consider that running a test $T_v$ incurs a unit cost $c(T_v) = 1$ and produces an outcome from a finite set of possibilities $X_v$, particular for each test.

Thus, the goal of the adaptive policy is to reliably identify the source, while minimizing the number of observers used.

**Observation Outcomes**

The outcome set of a test lies in one of two cases, depending on which information about the infection we consider to be available as soon as an observation is done:

- **State** (booleans): the outcome of a test $T_v$ represents whether the specific node is infected or not.
  $$X_v = \{\text{true, false}\}, v \in V$$

- **Time** (reals): the outcome of a test $T_v$ represents a set of possible real values of the infection time.
  $$X_v = \{t_{h,v} | h \in C_v\} \subset \mathbb{R}_+^\infty, v \in V$$

The two types of outcome have their advantages and disadvantages. By considering only boolean outcomes, the complexity of the computation related to the expected outcome of placing an observer is reduced significantly. However, the expected value is easily misleading, as less information is taken into account.
3. **Source Estimation using Active Learning**

In the case of test outcomes being actual discrete time values, we obtain the information about the possible infection times from the simulated cascades. Because the number of time values can be extremely large, upper bounded by $n \cdot |V|$ due to the fact that the node may be infected in each hypothesis in a cluster, we create a histogram with a maximum of $\sqrt{n \cdot |V|}$ buckets in which we place the infection times.

Let $H$ denote the random variable which is the ground truth and for each test $T_v$, let $X_{T_v}$ denote the random variable of the test outcome, taking values in $X_v$.

**The Underlying Graphical Model**

Due to the underlying network topology, the test outcomes are no longer independent as in the general Bayesian active learning setting. Particularly, the graphical model presented in Figure 2.1 needs to be augmented.

Intuitively, for each pair of node for which there exists a path through which the infection can travel, a dependency edge should be added in the graphical model.

Formally, for each $u, v \in V$ for which there exists a sequence of nodes $(w_k)_{k=1}^n$ such that $(u, w_1), (w_k, w_{k+1})_{k=1}^{n-1}, (w_n, v) \in E$, then a dependency between the random variables $X_{T_u}$ and $X_{T_v}$ should be added to the graphical model.

For undirected graphs, it implies that edges should exist between any two nodes situated in the same connected component. The augmented graphical model is depicted in Figure 3.5.

![Graphical Model for Test Outcomes in a Connected Network](image)

**Figure 3.5:** Graphical Model for Test Outcomes in a Connected Network

Even though the random variables of the test outcomes are no longer independent, according to the results in [8] this constraint can be relaxed, while preserving the same theoretical guarantees.
Graph Weight

Let us consider the *ordered set* of observations $x_A = \{x_{v_1}, x_{v_2}, ..., x_{v_m}\}$, where $v_k \in V$ is a vertex in the graph and $x_{v_k}$ is the observed outcome of $T_{v_k}$, the test associated with observing its infection state. Due to the nature of dependencies illustrated in Figure 3.5, the ordering of the observations is important.

The set $\{v_1, v_2, ..., v_m\} \subset V$ is just a permutation of the vertices in the graph, with the indices denoting the relative order in which the infection state was observed.

Consider $h \in C_v$, for some $v \in V$, then the joint probability of $h$ and the set of observations $x_A$, in the most general case, is:

$$P(h, x_A) = P(h)P(x_A|h)$$

In the case of observations revealing only the state of infection of a node (i.e., infected or not), a more convenient and realistic simplification is to assume that observation outcomes are independent, as in Figure 2.1. If $x_v$ is the outcome of $T_v$, then:

$$P(h, x_A) = P(h) \prod_{T_v \in x_A} P(x_v|h)$$

However, in the general case of observation revealing also infection times, in order to take them properly into account, we use the general model, using all the information available at the time of the observation:

$$P(h, x_A) = P(h)P(x_{v_1}|h)P(x_{v_2}|h, x_{v_1}) \cdots P(x_{v_m}|h, x_{v_1}, x_{v_2}, ..., x_{v_{m-1}})$$

We make use of an adaptive policy obtained by using one of the objective functions described in Section 2.2.3. After running each test, the hypotheses that are inconsistent with the outcome are removed.

In order to describe the adaptive policies using the objective functions described in Section 2.2.3, it is easier to first formally define the weight of graph $G$ as a function of the set of observations $w_G : x_A \rightarrow \mathbb{R}$

Considering the generalized binary search objective function, the graph weight is defined through the sum of joint probabilities over all the hypotheses and observed test outcomes.

$$w_G^{GBS}(x_A) = \sum_{v \in V} \sum_{h \in C_v} P(h, x_A)$$
For the equivalence-class edge-cutting objective function, the graph weight can be seen as the weighted sum of the edges between hypotheses belonging to different equivalence classes.

\[
\sum_{u, v \in E: u \neq v} \sum_{h \in C_u} \sum_{h' \in C_v} \frac{1}{P(h, x_A) P(h', x_A)}
\]

For our last objective function, value of information, the graph weight represents the maximum probability mass lying in a single equivalence class. Intuitively, it is the most probable hypothesis class after the set of observations in \(x_A\).

\[
\max_{h \in C_v} P(h, x_A)
\]

As a last remark related to the weight functions, one can further simplify the \(w^{EC^2}_G\). If we let \(P(C_v, x_A) = \sum_{h \in C_v} P(h, x_A)\), then by rewriting the weight function it yields that \(w^{GBS}_G(x_A) = \sum_{u, v \in V: u \neq v} P(C_u, x_A) P(C_v, x_A)\) and it immediately follows that:

\[
(w^{GBS}_G(x_A))^2 = (\sum_{v \in V} P(C_v, x_A))^2 - \sum_{v \in V} P(C_v, x_A)^2
\]

The simplified version clearly illustrates that the computational cost between the three objective functions is only a constant factor.

**Adaptive Policies**

The greedy adaptive policies built on top of the weight functions previously defined follow directly, by always selecting the next test that maximizes the expected marginal gain, given the previous observations. Formally,

\[
\Delta^{Alg}_G(T_v | x_A) = w^{Alg}_G(x_A) - \sum_{x \in X_v} P(X_{T_v} = x | x_A) w^{Alg}_G(x_A, T_v = x),
\]

where \(Alg \in \{GBS, EC^2, Vol\}\). The expected marginal gain basically represents the expectation over the possible test outcomes of the change of the graph weight, with respect to a given algorithm.
3.4 Accomodating noisy sensors

An essential aspect that impacts the accuracy of the proposed algorithm still remains to be defined. Particularly, how does the probability mass of a hypothesis change after performing an observation, $T_v$?

The first approach we have experimented with was to consider a noiseless model, under the simplified assumption of independent boolean test outcomes describing the infection of the node, such that $P(h, x_A) = P(h) \Pi_{T_v \in x_A} P(x_v|h)$.

In this noiseless setting, we have that $P(x_v|h) = \begin{cases} 1, & \text{consistency} \\ 0, & \text{otherwise} \end{cases}$, where a boolean observation outcome $x_v \in \{\text{true, false}\}$ is consistent with $h$ if the state of infection of node $v$ in $h$ is the same as $x_v$.

As within the bayesian active learning framework, the algorithm stops when there is at most one equivalence class remaining with hypothesis having positive probability masses.

If we denote through $V(x_A) = \{h \in C_v : P(h, x_A) > 0, v \in V\}$, then the greedy test selection should stop when $|V(x_A)| = 1$.

There are a couple disadvantages with the approach, both due to its noiseless assumption and to the boolean outcomes, as opposed to the real-world scenario, where sensors are noisy and nodes can be infected in discrete time:

- **unused information**: less information is used than it is available when considering only the infection state instead of the infection time.
- **search space reduction**: the search space reduces dramatically with each observation, easily removing the equivalence class in which the source node lies in very early.
- **equivalence class differentiation**: considering the case when at the end of the algorithm there are more than two equivalence classes which have hypotheses with non-zero probability masses, ties need to be broken arbitrarily.

Moving forward and considering more complicated cases leads to the introduction of a noise parameter. We admit that sensors might have noisy outcomes, but move this noisy aspect to the hypothesis search space, by introducing a soft method of downweighing a hypothesis, instead of a harsh removal from the space, as previously.

The solution accommodates both the impossibility of capturing the whole space of possibilities and the situation in which noise is present in the observations, due to the quality of the sensors.

Let $\varepsilon \in (0, \frac{1}{2})$ denote the error tolerance, such that:
3. Source Estimation using Active Learning

\[ P(x_v|h) = \begin{cases} 
1, & h \text{ consistent} \\
\frac{\varepsilon}{1 - \varepsilon}, & h \text{ inconsistent} 
\end{cases} \]

An important situation, not yet analyzed, is the case when test outcomes have real values, containing the infection time of the node tested. Particularly, considering the set of ordered observations (tests) \( x_A = \{x_{v_1}, x_{v_2}, \ldots, x_{v_m}\} \), when is a hypothesis \( h \in \mathcal{H} \) consistent with observation \( x_{v_{m+1}} \)?

Given the graphical model previously illustrated for the problem, an observation is inconsistent not only with respect to a hypothesis, but also with respect to the previous observations \( x_A \).

In the experiments of Chapter 4, we considered that a hypothesis \( h \in \mathcal{H} \) is consistent with test outcome \( t_{v_{m+1}} \), \( h \in \mathcal{H} \in \mathbb{R} \) and previous observations \( x_A \), if

\[ \text{sgn}(t_{v_{m+1}} - t_{v_{m+1}}) = \text{sgn}(t_{v_{m+1}} - t_{v_{m+1}}), \quad v \in V, \text{ with } T_v \in x_A. \]

The variable \( t_{v_{m+1}} \), \( h \) denotes, as previously mentioned, the infection time of node \( v \) in the ground truth \( h_{gt} \). In simple words, the idea behind it is that the relative infection order between nodes is preserved.

One can also further imagine more complicated functions instead of a constant value down-weighting. Particularly, if we consider the case with real infection times, rather than considering that a hypothesis is completely inconsistent if order is not preserved, the down-weighting amount could also be proportional to the number of inconsistencies.

**Identifying the Source Node**

After the set of observations \( x_A \) the source is considered to be the label of the equivalence class in which the most mass is concentrated within. Formally,

\[ s = \arg\max_{v \in V} P(C_v, x_A) = \arg\max_{v \in V} \sum_{h \in C_v} P(h, x_A) \]

**Key Remarks**

The model is basically free of prior assumptions about the diffusion instance that we are searching for, but, should such information exist, it can easily be incorporated within the model in multiple places, extending it.

- **Diffusion Model**: through the way we built the version space, it lacks of dependencies on a particular diffusion model; one can easily plug-in different diffusion models.
3.4. Accomodating noisy sensors

- **Hypothesis Priors**: if one knows that the source is more probable to be in a specific part of the network, the hypothesis classes associated with the respective nodes can be initialized to have non-uniform priors.

- **Diffusion Spread**: if there is existing information about the actual diffusion, it can be incorporated to generate diffusions in the hypothesis search of the same size.

As a short summary to this chapter, the following are the key aspects of the novel approach that we have come up with:

- **Source Estimator**: the model we have built is a novel way of estimating the source of a diffusion and can be used independently of the observer placement strategy.

- **Adaptive Observer Placement**: through hypothesis weighing we provide a method of adaptive placing the observers, one by one, guiding the process to the go to the most informative nodes associated with the source of the infection that we are searching for.

- **Noise Model**: we have built a simplistic model of admitting noise in sensor measurements and incorporated it within the estimation method.

- **Extensibility**: the model is easily extended to incorporate information about particular problems being solved.
Chapter 4

Experimental Results

In this chapter we present the experimental results we have obtained by making use of the theoretical models presented in the previous chapters.

Identifying the source of diffusion is a hard problem and among the two solutions that we have analyzed, we are clearly not stating that one performs on average better than the other, regardless of the conditions.

Rather than establishing such a strict performance comparison, our goal is to use various metrics to understand how the two solutions differ and under which particular circumstances one behaves better than the other.

Stretching the limits of both models aids the thorough understanding of their performance and represents a preliminary step for further developing better theoretical guarantees that can be useful to the problem of source localization through the models.

4.1 Synthetic models and metrics

We first present the particular properties of the network topology models that we have experimented upon, referencing other works for more detailed information.

4.1.1 Network models

Erdős – Rényi

The Erdős – Rényi graph model refers to the $G(n, M)$ random graph model, when a graph selected at random from the set of all possible graphs with $n$ vertices and $M$ edges. It was first studied in 1959 by Erdős and Rényi [5], with the theory surrounding random graphs further extended by them in the following year [6].
4. Experimental Results

The theory surrounding random graphs has grown ever since. In our particular configuration for the thesis, we set $M \sim n \log n$, such that the random network generated is connected.

Another particular aspect with respect to this topology is that the network randomly picked, is not necessarily a realistic model, but it nonetheless provides valuable insight with respect to the performance of the solution.

**Barabasi – Albert**

The Barabasi - Albert model was studied as a network model that closely resembles the graph associated with the world wide web and citation networks [1].

The key principles behind this topology is that the nodes exhibit a power-law degree distribution: that is, the fraction of nodes $F(k)$ having degree $k$ is proportional to $k^{-\gamma}$, where $\gamma$ in our experimental setup lies within $(2, 3)$.

The model can also be seen through the process of preferential attachment, as it exhibits the property that whenever a new node is added to the network, it attaches to already well-connected sites.

It has proven to approximate very well how the www network behaves. One can think at the particular situation of news websites and news aggregators: when a news website appears, most probably a news aggregator will start taking it into consideration, and therefore attach it to an already well-connected site.

**Forest Fire**

The forest fire network topology is a novel topology, shown to approximate the normal growth patterns in social, technological and information network [14]. The key properties of the model are that, over time, graphs *densify* and their *diameters shrink*.

It is worth noting that, according to the authors, none of the existing topologies prior to their research exhibit the properties of modeling how graphs evolve over time - a natural expectation, considering that there never existed so much information about how large graphs evolve, as we have now.

As the name suggests, the generation model makes use of the natural process of fire spreading in forests, and we encourage the reader interested to receive more information on the subject to peruse the paper written by Leskovec et. al in 2007 [14].
4.1. Synthetic models and metrics

4.1.2 Diffusion models

In the experimental setup, we made use of two different diffusion models in order to have a thorough understanding of the two models.

We believe that the diffusion models play a key role in the performance of the source estimation algorithms and we would like to further understand what happens with the two models when one or the other is used.

Susceptible infected

The SI model assumes that each edge \( e \in E \) has an associated probability \( \beta_e \in (0,1) \) to be activated such that the diffusion travels through it, from one node to the other.

If node \( u \in V \) is infected at time \( t_u \), then, at each timestep, its neighbour \( v \in V \) has \( \beta_{(u,v)} \) probability of becoming infected by node \( u \).

If we denote through \( V(u) \subset V \) the current set of infected nodes, one could think about the process as if, at each timestep, with some probability, the infection frontier is advancing to nodes \( v \in V \), for which there exists \( u \in V(u) \) and \( e = (u,v) \in E \), such that \( e \) is activated.

In this case, the infection time of a node \( t_v \in \mathbb{N} \) simply denotes the timestep when it got infected.

Propagation delays as Gaussian RVs

The model was already presented in Section 2.1 and is the underlying model for which the process was reversed, by building a maximum-likelihood estimator in [16].

As a brief reminder and comparing with respect to the previously described model, in this scenario there exist a propagation delay \( \theta_{(u,v)} \) associated with each edge \( e = (u,v) \in E \). The propagation delay is sampled according to a normal distribution \( \mathcal{N}(\mu_e, \sigma_e) \).

It is worth noting that when a node \( u \) is infected, its neighbour \( v \) will certainly become infected too, with \( t_v = t_u + \theta_{(u,v)} \).

Practical considerations

Since the maximum likelihood estimator described in [16] needs to know beforehand the parameters of the normal distribution in order to compute scores, we apply an approximation of the geometric distribution with a Gaussian.
4. Experimental Results

This is also the method suggested by the authors in the supplemental material of the paper, whenever the model is applied in the case of a different diffusion model.

In the case of the susceptible-infected model, the propagation delays associated with the edges can be seen as a random variable $\theta$ sampled from a geometric distribution with parameter $\beta \in (0, 1)$.

In this case, the mean and variance of the random variable $\theta$ are given by $\mu = \frac{1}{\beta}$ and $\sigma = \frac{1-\beta}{\beta^2}$. These are the parameters that are being used for the Gaussian maximum-likelihood estimator, whenever the model is applied on a diffusion model of susceptible infected.

4.1.3 Metrics: hops, rank and nDCG

Hops to Real Source

The number of hops on the shortest path between the top ranked source and the actual source of the diffusion represents one of the metrics through which we looked at the performance of the algorithms.

One aspect to consider when looking at this metric is that it’s more important comparatively, rather than absolutely, because the largest shortest path between two nodes differs depending on the topology of the network.

Rank of Real Source

Each of the two estimation methods ends up computing a score for each node in the network and deciding which one is the source simply requires picking the node with the maximum score among these.

The rank metric designates on which position the real source is in the list of nodes sorted descendently by the score they have.

Normalized Discounted Cumulative Gain (nDCG)

If we consider the output of the estimation models from an information retrieval perspective, we can think of the estimation models as returning an list of nodes ordered according to the their score.

Then, we can also consider an ideal ordering, in which each node has as score its distance from the true source, on the shortest path.

The discounted cumulative gain metric takes into consideration both the score of the node and its position in the ordered list (the higher position and higher the score, the better).

If we look at the top items, the $DCG_p$ represents total gain gathered:
4.2 Key results

\[ DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i)}, \]

where \( rel_i \) represents the relevance of item on position \( i \).

Considering an ideal ordering of the nodes we obtain \( IDCG_p \) and then the normalized metric is simply \( nDCG = \frac{DCG_p}{IDCG_p} \).

4.2 Key results

Through several case studies, which cover real-world situations, we are presenting the performance of the models and how our active learning approach can be used sometimes to improve the maximum likelihood estimator from [16], particularly in limit situations.

In order to familiarize the reader with the graphs presented within each case study, we described the key metrics in Section 4.1.3 provide a quick guideline how to look at a graph illustrating comparative results.

![Figure 4.1: Explanatory Plot: Main Algorithms](image)

Each of the case studies presented will contain several graphs similar in structure to the one explained in Figure 4.1. The figure does not contain any reference to any particular metric, rather it highlights the three different approaches we compare.
4. Experimental Results

- **MLE original** represents the maximum-likelihood estimator from [16] with the original suggested observer placement strategy of greedily picking highest degree nodes.

- **MLE extension** represents the same maximum-likelihood estimator, but this time the observers are selected using active learning, and are the same ones used when the $EC^2$ objective function is used for the set of hypothesis built in our approach.

- **Thesis model + adaptive placement** represents the model described in Chapter 3, in combination with using objective functions to adaptively place observers.

- **Thesis model + non-adaptive placement** represents the same model as previously, but this time with a fixed non-adaptive policy, such as random and highest degree nodes.

The performance of any algorithm is more important at the beginning of the graph, when less observers are used, than at the end of the graph. Within this section we will cover the thorough experimental analysis that we have conducted as part of the thesis.

4.2.1 Results I: Known diffusion parameters

In this first set of results we present the performance of our estimation model when they have access to the parameters that generated the diffusion, according to the diffusion model ($\beta$, or $\mu$ and $\sigma$).

The following have been used for each of the experiments in the several case studies discussed:

- **Network**: 100 nodes, a single connected component
- **Equivalence class size**: 100 hypothesis, generated using the SI model
- **Ground truths**: 100 infections starting from random nodes
- **Error bars**: standard error

Whenever the diffusion model is assumed to be **gaussian**, the parameters used for the maximum-likelihood estimator are $\mu = 8, \sigma = 2$, and represent the parameters used for generating the diffusion. As noted in [16], the ratio $\mu/\sigma \geq 4$, ensures that the model is actually realizable.

Whenever the diffusion model is assumed to be geometric, the parameters plugged in to the maximum likelihood estimator are according to the relation specified in Section 4.1.2.

We present in this section the most interesting results that we have obtained in several (extreme) situations we analysed, on one of the network topologies that exhibited the most interesting results.
4.2. Key results

We provide the complete list of figures on the rest of the network topologies, for the interested reader, in the appendix, in Section A.1.
4. Experimental Results

Main Result #1a  
**Model:** Gaussian  
**Spread:** 100%

**Description:** In the case of an infection affecting the entire network, spreading according to the Gaussian joint distributed propagation delays, the MLE always performs better than the approach developed by the thesis, but on Forest Fire it is better with adaptive placement.

**Topology:** Barabasi - Albert

![Graph showing Hops to Real Source (Barabasi)](image)

The results are naturally expected, as the MLE was built specifically reversing the Gaussian case, and due to the size of the infection, all the nodes will provide relevant information about the infection.

![Graph showing Average Rank of Real Source (Barabasi)](image)

When looking at the average rank, it’s interesting to note that, for our model, picking the highest degree observers, even though the barabasi network...
4.2. Key results

topology exhibits large hubs, performs the worst.

The less expected result is the performance obtained by our model when considering the nDCG metric.

Looking towards the end of the graph, the performance is fairly close between the two estimation models. In the end, most probably the source or the nodes very close to it lie within the top 5%.

**Topology:** Forest Fire

When considering the forest fire topology, from around 18% of the observers, the MLE extension performs significantly better, where observers are selected adaptively using active learning. This can be observed on all metrics, which are provided in the appendix.
4. Experimental Results

**Main Result #1b**  **Model**: Geometric($\beta = 0.40$)  **Spread**: 100%

**Description**: As in the previous case, each of the nodes in the network is infected, but the propagation delays are according to the susceptible infected model, as presented in Section 4.1.2.

**Topology**: Forest Fire

![Graph showing hops to real source](image)

The performance of our estimation method is fairly the same as in the previous case, while the performance of the MLE estimator isn’t improving as the number of observers increases. Adaptive placement of observers improves it a bit when looking at the average rank and average hops plots, but for the top 5% observers is not necessarily the case.

![Graph showing nDCG](image)

When looking at the top 5% nodes through the nDCG metric, it seems that
4.2. Key results

non-adaptive policies, random and high-degree, in combination with the source estimation method proposed by the thesis perform well.

This is due to the fact that no matter the node selected, it will yield an infection time associated with the diffusion, guiding the estimation properly.

**Topology: Barabasi - Albert**

When the network topology changes, the main trend are kept, the MLE isn’t able to identify the source, and our estimation method is, but within the objective functions used, the best is now obtained with $EC^2$ and $GBS$. 
4. Experimental Results

Main Result #2a  Model: Gaussian  Spread: 50%

Description: The case of having only half of the network nodes infected is a realistic situation, either due to the diffusion stopping to propagate after some time, or due to running the algorithm in a situation when the diffusion is still spreading.

Topology: Erdos - Renyi

![Graph showing Hops to Real Source (Erdos)](image)

In this case, our model using the VoI objective function to adaptively place observers performs almost as good as the MLE approach, and the best possible approach is the MLE extension.

![Graph showing nDCG @ Top 5% (Erdos)](image)

Through the nDCG metric we can observe that our estimation model with VoI objective places most valuable nodes in the top 5% out of all methods.
4.2. Key results

Our estimation with $EC^2$ or $GBS$ also performs better than the MLE with respect to the nDCG metric, but plugging in adaptive placement of observers still yields better results.

In the plot related to the rank, one can also observe the difference between placing observers adaptively and non-adaptively for our estimation method.

**Topology:** Forest Fire

For the forest fire topology the results are not as good as for the Erdos - Renyi random graph. Even though the VoI objective function reaches similar performance, the GBS and $EC^2$ objectives are outperformed.

The best approach still holds when combining the MLE estimator with the adaptive placement of observers.
4. Experimental Results

**Main Result #2b**  
**Model:** Geometric ($\beta = 0.40$)  
**Spread:** 50%

**Description:** The situation is identical with the previous one, only the diffusion model is now susceptible-infected. The essence of this case is to study the robustness of the models. The gaussian parameters used in the MLE approach are the best approximation of the geometric distribution, as mentioned in Section 4.1.2.

**Topology:** Barabasi - Albert

In this scenario, the estimation strategy proposed in the thesis clearly outperforms the maximum-likelihood estimator, with respect to any of the metrics.

While the performance of the thesis model in this scenario is not as good as the MLE in the Gaussian case, it still provides a fairly good solution.
4.2. Key results

According to the rank of the real source, the MLE performs quite badly, as from 10% of the observers onwards, it actually increases. Using an adaptive placement strategy seems to improve it for another 10%, but in the end reversing the process using an approach assuming a Gaussian fails.

The key insight from this result would be that building an estimator with such a strong assumption about the direct diffusion process is not necessarily robust and can perform badly in conditions where parameter approximation is required.

On the other two network topologies, Forest Fire and Erdos - Renyi, the results are very similar, as trends, to the ones already presented. We provide the rest of the plots and parameters that can be used to reproduce them in the appendix.
4. EXPERIMENTAL RESULTS

**Main Result #3a**  
**Model:** Gaussian  
**Spread:** 10%

**Description:** This situation illustrates a very incipient state of infection, with very few nodes being affected. In this scenario, the adaptive placement of observers strategy places a crucial role, regardless of the estimation model used to solve the problem.

**Topology:** Erdos - Renyi

In the graph above, illustrating the number of hops to the true source, two trends can be easily distinguished: methods making use of adaptive placement of observers and methods who pick observers non-adaptively, based on the highest degree or random.

Clearly, active learning efficiently guides the estimators towards the most
4.2. Key results

relevant nodes. Considering the significant performance improvement of the MLE, active learning guides the process to place earlier observers in nodes that are infected and, thus, have timing information available.

**Topology**: Forest Fire

For the forest fire topology, the best approach is the MLE extension, and among objective functions, the $EC^2$ and $GBS$ objective functions.

The increase in performance when using the MLE in combination with an active learning strategy of placing observers is significant. Also, through the nDCG metric we can see a very strong differentiation between each of the different methods used to solve the problem.
4. Experimental Results

**Main Result #3b**  
**Model:** Geometric(\(\beta = 0.05\))  
**Spread:** 10%

**Description:** This is probably the toughest situation for the MLE estimator. The theoretical diffusion model isn’t realistically represented by a Gaussian, due to the ratio \(\mu/\sigma < 4\). Because of its strong dependency on the Gaussian model, the MLE is unable to properly estimate the source.

**Topology:** Erdos - Renyi

As in some previous plots, the three groups of estimation strategies can be easily distinguished. The difficulty of the situation is highlighted also by our own estimation method, as the average distance in hops to the true source, even when all the observers are used, is still above zero.

The average rank plot for this scenario shows that slowly the source is
4.2. Key results

ranked further and further away in the MLE strategy, while using our estimation it gets quite strong by the end.

Combining with the insight gained from the average hops plot, it looks like our estimation method efficiently identifies the area where the source is, but not exactly the node that is the source.

![Graph showing average rank of real source](image1)

**Topology**: Forest Fire

When the infection spreads on this network topology, our estimator performs slightly better than on the random graph model, with the $EC^2$ and $GBS$ objective functions performing as good with respect to the rank of the true source.

![Graph showing average rank of real source](image2)
4. Experimental Results

4.2.2 Results II: Searching for the unknown

Throughout the previous section, we have observed that the model proposed by the thesis can perform relatively well, but in all the situations presented, parameters were known beforehand.

Therefore, we now assume that the hypotheses search space of the model is not generated according to fixed parameters, rather we sample the parameters from a uniform distribution over a fixed interval.

Due to the fact that we are not aware of a method of estimating $\mu$ and $\sigma$ for the MLE approach, we have considered that it’s not appropriate to randomly sample these parameters, as the authors have never suggested this in their paper [16].

Therefore, we resume this section to compare the relative decrease or increase in performance of the objective functions performing best in the previous section, in combination with the estimation method proposed by the thesis only.

Sampling intervals

Since we don’t know anything about the infection beforehand, we generate our hypothesis search space according to the susceptible-infected model, and we sample uniformly at random $\beta$, the probability of infecting a neighbour node, and $s$, the size of the hypothesis.

The intervals we chose are $\beta \in (0.05, 0.75)$ and $s \in (0.05, 0.50)$. These were chosen empirically, as within this range we have observed them to perform well in the previous section of the results.

While this is definitely not a rigourous approach and we cannot generalize, without theoretical guarantees, that these parameters perform well under any circumstances, we are interested at this stage to understand the cost of this approach and its performance.

Each of the following plots illustrate, comparatively, the two cases:

- fixed parameters (blue)
- uniform parameter sampling (red)

For the fixed parameter values in the plots, they are taken from the corresponding main result in the previous section and the specific objective function (the same diffusion model and the same spread size).
4.2. Key results

**Extended Result #1a**

**Model:** Gaussian  
**Spread:** 100%

**Topology:** Barabasi - Albert

![Graph](image.png)

**Figure 4.2:** VoI objective - Average Rank (left), nDCG @ Top 5% (right)

For the VoI objective, according to both metrics, there’s an incurred cost of sampling uniformly the parameters instead of setting them beforehand.

![Graph](image.png)

**Figure 4.3:** EC² objective - Average Rank (left), nDCG @ Top 5% (right)

Interestingly, for the EC² objective function, through the nDCG metric, there’s actually an improvement in the top 5% nodes.
4. Experimental Results

**Extended Result #1b**  
**Model:** Geometric($\beta = 0.40$)  
**Spread:** 100%

**Topology:** Forest Fire

![Graphs](image)

*Figure 4.4: VoI objective - Average Rank (left), nDCG @ Top 5% (right)*

*Figure 4.5: EC² objective - Average Rank (left), nDCG @ Top 5% (right)*

In this situation, as it can be seen from all of the above plots, sampling parameters clearly helps a lot. Diversifying the search space is essential in some circumstances, but unfortunately, the results are not easily generalizable.
Extended Result #3a  
**Model:** Gaussian  
**Spread:** 10%

\[\text{Average Rank of Source Node (EC2) (Erdos)}\]

\[\text{Average nDCG @ Top 5\% (EC2) (Erdos)}\]

**Figure 4.6:** \(EC^2\) objective - Average Rank (left), nDCG @ Top 5\% (right)

\[\text{Average Rank of Source Node (VOI) (Erdos)}\]

\[\text{Average nDCG @ Top 5\% (VOI) (Erdos)}\]

**Figure 4.7:** \(VoI\) objective - Average Rank (left), nDCG @ Top 5\% (right)

In the case of partial Gaussian diffusion, the sampling of parameters does not have such a strong impact on the performance, considering the average rank metric.

The underlying truth is shown in the nDCG metric, where the costs can be seen and are definitely not negligible.
4. **Experimental Results**

**Extended Result #3b  Model:** Geometric ($\beta = 0.05$)  **Spread:** 10%

![Graph 1](image1)

**Figure 4.8:** $EC^2$ objective - Average Rank (left), nDCG @ Top 5% (right)

![Graph 2](image2)

**Figure 4.9:** VoI objective - Average Rank (left), nDCG @ Top 5% (right)

Through the above graphs, it’s clear that in these limit cases, when the infection has spread only on a tiny fraction of the nodes, sampling parameters has a significant negative impact, due to the fact that very few of the hypotheses are as small as the ground truth and therefore, many inconsistencies appear.

**Remarks**

We have intentionally omitted the case when the diffusion has spread only 50%, as the results don’t bring new information compared to the plots presented above.

As a summary of the section, we believe that sampling parameters provides a key element for generalizing the model to perform well enough in unknown situations, but through the thesis we are not yet able to provide theoretical guarantees for it and instead consider it for future work.
4.2. Key results

4.2.3 Results III: Variability of the model

Through the previous sections we have illustrated how the model can perform by selecting some of the best outcomes among the possible combination of parameters for the model.

In this section we select a couple of comparative graphs illustrating how some model parameters affect its performance when solving the localization problem, while keeping all the others fixed.

Noise

As it is expected intuitively, we observe that the noise impact is higher the more the infection has spread across the network. The number of inconsistencies with the hypothesis search space will easily be higher than if only a small fraction of the nodes are infected, as more information is available.

We restrict the plots only to the $EC^2$ objective function and we note that similar trends can be observed for the others. Among the metrics, the rank of the real source node and the average hops are very similar as well, and therefore we only illustrate the latter.

The network topology also plays a role as to which noise value yields better results, as can be seen from the plots provided below. We have experimented with several values of the noise level $\epsilon \in \{0.01, 0.05, 0.10\}$.

The diffusion model is jointly-Gaussian distributed propagation delays, but using susceptible-infected model yields similar trends.

![Average Rank of Source Node (EC2) (Forest)](image1)

![Average nDCG @ Top 5% (EC2) (Forest)](image2)

**Figure 4.10**: Spread 100%: noise impact for $EC^2$ on forest fire

**Note**: The plot colors are \{red, blue, green\} for $\epsilon \in \{0.01, 0.05, 0.10\}$, respectively.

We note that in Figure 4.10 the noise has a significant impact on the accuracy of the algorithm. Particularly, the extreme values exhibit slightly better results.
As the true infection size decreases, the search space variability decreases as well and a lower impact of the noise value is expected, as it can be observed in Figure 4.12 and in Figure 4.11.

Intuitively, the smaller the ground truth is, the easier it is to have its characteristics captured in the artificial search space that we generate through the equivalence classes from each node.

**Hypothesis Size**

In this subsection we illustrate the impact of the hypothesis size on the accuracy of the algorithm. The size refers to the percentage of nodes infected in the artificial diffusions that we generate from each node in the graph.

For understanding the impact we illustrate the plots in the situation of a susceptible-infected diffusion that has spread entirely on the network.

With some variability, the trend in the rest of the situations, namely other objective functions and ground truth diffusion spread size, is very similar with the ones depicted.
4.2. Key results

Figure 4.13: Forest Fire

Figure 4.14: Erdos - Renyi

Figure 4.15: Barabasi - Albert

Note: The plot colors are \{red, blue, green\} for hypothesis size $\in \{0.05, 0.50, 0.95\}$, respectively. This means that each hypothesis in the search space infects 5%, 50% or 95% of the network nodes.

The results revealed by these plots is that the performance of the algorithm is significantly better if the search space represents small possible diffusion
clouds within each node, rather than having a realistic hypothesis of the same size of the ground truth that we are trying to identify.

Intuitively, this comes from the fact that the larger the artificial hypothesis, the higher the chances of test outcomes being inconsistent with more of them, due to the exponential size of the search space.

**Network Nodes**

As intuitively expected, the network size has an impact on the performance of the algorithm. In this section we illustrate the impact of keeping the same parameters of the model, while growing the network size from 100 nodes to 500 nodes.

There is one theoretical aspect to be thought of, however, revealed in Section 2.2.3 regarding the theoretical cost guarantees of \( EC^2 \): the lower the prior probability of the hypothesis, the larger the margin within the optimal cost.

Since growing the network size directly decreases the prior probability of each of the hypothesis, we expect the cost to possibly grow. As with the previous sections, we illustrate the performance change for two metrics, on the different network topologies.

![Figure 4.16: Forest Fire](image)

**Note:** The plot colors are \{red, blue, green\} for network size \( \in \{100, 240, 500\} \), respectively.

While we don’t have a theoretical explanation, it’s interesting to observe that the algorithm is fairly robust when it comes to increasing the network size for the Erdos - Renyi random graph.

For the other topologies, the rank of the true source from 60% observers onwards is pretty much the same, but then looking at the nDCG metric we identify the issue: while it’s among the top nodes returned, the estimation method isn’t able to easily differentiate between them.
4.3 Engineering overview

From our understanding of the model at the time of this writing, we expect that growing other parameters, the cluster size for example, as a function of the network size, likely keeps its accuracy similar.

4.3 Engineering overview

The engineering elements of the thesis project have played an important role throughout the whole period and therefore this section is presenting the key elements through which we were able to easily scale the implementation and evaluate quickly the performance of the algorithms in so many situations.

Having a scalable implementation quickly would not have been possible without several open source frameworks, which we further mention in the following section. As a result of the benefit we have had from the numerous open tools over the years, we will do our best to keep an open source version of the implementation in order to easily reproduce the results presented within the thesis.

As part of the thesis software, we also provide the first open source imple-
4. Experimental Results

mentation of the maximum-likelihood estimator developed in [16].

Aside from the scope of reproducing the results, ideally the platform we built could be extended and improved in order to have a unique benchmark for evaluating future methods of solving the problem of source localization.

4.3.1 Frameworks: SNAP, Eigen and Open MPI

We have made use of the network generators available through the Stanford Network Analysis Platform \(^1\) and we have also borrowed the implementation of the susceptible-infected cascade generation.

Aside from generators, the SNAP framework has also provided several useful algorithms, such as computing shortest paths in graphs.

In order to easily implement maximum-likelihood estimator, the Eigen \(^2\) template library for linear algebra was extremely useful and straightforward to use.

Not depending on heavy software, and being able to simply plug-in a couple of C++ header files for linear algebra operations was a very effective speed-up on the implementation side.

Lastly, scalability was achieved through the Open MPI \(^3\) project, which is an open source implementation of the MPI-2 standard\(^4\). The reason we used the message passing interface instead of map reduce is that the Brutus and Euler clusters at ETH have native support for this paradigm.

We also wanted to make sure that that the implementation doesn’t use more I/O than needed, which happens often when solving problem using the MR paradigm through the intermediary phases between multiple map reduces. In the next section we briefly illustrate the way we parallelized computations through multiple nodes.

4.3.2 Distributing lazy greedy computations

The cores requested on the cluster are split into P worker nodes and 1 central node collecting partial scores computed by each of the nodes, according to the objective function.

Each worker node is responsible for:

- Reading the network topology from the centralized file
- Creating the clusters of hypotheses associated with N/P nodes

---

\(^1\)http://snap.stanford.edu  
\(^2\)http://eigen.tuxfamily.org  
\(^3\)http://www.open-mpi.org  
\(^4\)http://www.mpi-forum.org
4.3. Engineering overview

**Figure 4.19:** Scaling Lazy Greedy through Computing Nodes

- Listening to requests from the Master node for recomputing partial scores for a particular node, with respect to a given objective function and its instantiated clusters
- Listening to requests for centralizing information about the current weight of the graph and distributing the partial mass contained within the node to the master node

The **master node** is responsible for:

- Reading the network topology from the centralized file
- Managing the greedy priority queue
- Requesting recomputations for specific nodes in the priority queue, facilitating a lazy greedy approach which, even though it has theoretically the same complexity, in practice yields faster computations due to the adaptive submodularity of EC² and GBS
- Computing a centralized score for a test, after receiving the partial scores from each of the worker nodes
- Processing results and outputting the final scoring of the estimation methods
- Computes mean and standard error for each of the variables analyzed in the graphs from the results section

While this approach definitely has its drawbacks, for example the division of nodes per cluster should be as uniform as possible (therefore the number of compute cores requested can be optimized according to the total number of nodes in the graph), it provided a satisfactory speed for the purpose of the thesis and evaluating the results with the preliminary model that we built.
Chapter 5

Conclusion

Summary. Throughout the thesis, we have tried understanding what chances of success exist, in solving the problem of source localization, through an active learning approach.

We have had a strong interest in varying the conditions in which the diffusion happens, through multiple network topologies and a set of other controlled parameters.

The model that we have built has a number of key advantages, through which the approach has proven to have a high degree of generalization:

- any theoretical diffusion model can be easily plugged-in
- the cost of the adaptive policy for placing observers is theoretically guaranteed to be in the range of the cost for the optimal one
- fairly robust, when a different diffusion model is used than the one that generated the diffusion

The most important conclusion is that through the variety of experiments we have undertaken, we understood that the problem of source localization is highly dependent on many factors and, as it is the case in the field of machine learning, there’s no free lunch.

Among the two models analysed, it cannot be stated that one performs better than the other, regardless of the theoretical diffusion process and its realization, or the underlying network topology.

Future work. Even though, through the thesis, we have answers for most of the initial questions, we ended up with even more additional questions.

Particularly, while we have proved experimentally that the model can perform extremely well, there are a couple of items missing in the puzzle.
5. Conclusion

Within the model that we built, more complex probability functions for the noisy downweighing step might help, as well as more complex functions for sampling parameters, instead of sampling them at random.

Theoretical guarantees is also a must-have component, before deploying the model in a real-world scenario. Due to the restricted timeframe of the thesis, we postponed developing theoretical guarantees in order to first understand whether there’s any chance of success with the model.

Since the graph is assumed to be known beforehand, one could introduce a training phase with various synthetic diffusions on the particular topology and tune the model parameters such that it performs the best and, only after that, localize the source in the real-world environment.

We hope that our approach might engage other researchers as well and we are looking forward to further develop and generalize the framework.
Appendix A

A.1 Reproducing Results

Through this section of the appendix we intend to illustrate a complete overview of all the relevant plots associated with the main results from Section 4.2.1. This includes the rest of plots on other network topologies not included due to space constraints.

In this section we also include the parameters to reproduce the experiments. However, please refer to the help instructions or README file of the most recent version of the software associated with thesis for the precise method of specifying these parameters ¹.

Short reminder on what the parameters represent:

- \( N \) = network nodes
- \( CS \) = count of artificial diffusions, hypothesis, starting from each node
- \( CB \) = percentage size of the total network of each hypothesis
- \( \beta \) = edge activation probability in the geometric diffusion
- \( \epsilon \) = noise value used for downweighing hypothesis when inconsistent
- \( \mu, \sigma \) = moments of the normal distribution used for MLE

The rest of the figures and results present in the thesis can easily reproduced as most of them have as base case one of the main results, and one of the parameters changed (e.g. noise level, network size etc.)

¹ [http://github.com/vcarbune/netloc](http://github.com/vcarbune/netloc), at the time of writing
A. Appendix

Main Result #1a

Spread: 100%

Model: Gaussian

Params: $N = 100$, $CS = 100$, $CB = 0.50$, $\epsilon = 0.01$, $\beta = 0.125$, $\mu = 8$, $\sigma = 2$
A.1. Reproducing Results

Main Result #1b  
**Spread:** 100%  
**Model:** Geometric

**Params:** $N = 100, CS = 100, CB = 0.05, \varepsilon = 0.01, \beta = 0.40, \mu = 2.5, \sigma = 1.93$
A. Appendix

Main Result #2a  
**Spread:** 50%  
**Model:** Gaussian

**Params:** $N = 100, CS = 100, CB = 0.05, \epsilon = 0.01, \beta = 0.125, \mu = 8, \sigma = 2$
**A.1. Reproducing Results**

**Main Result #2b Spread: 50%**  
**Model:** Geometric  
**Params:** \( N = 100, \) \( CS = 100, \) \( CB = 0.05, \) \( \epsilon = 0.01, \) \( \beta = 0.125, \) \( \mu = 8, \) \( \sigma = 2 \)
Main Result #3a  
**Spread**: 10%  
**Model**: Gaussian

**Params**: $N = 100$, $CS = 100$, $CB = 0.05$, $\epsilon = 0.01$, $\beta = 0.125$, $\mu = 8$, $\sigma = 2$
A.1. Reproducing Results

**Main Result #3b**

**Spread:** 10%

**Model:** Geometric

**Params:** $N = 100$, $CS = 100$, $CB = 0.05$, $\epsilon = 0.01$, $\beta = 0.05$, $\mu = 20$, $\sigma = 19.49$


Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor’s thesis, Master’s thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

Active Learning for Source Localization

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s): Carbune
First name(s): Victor

With my signature I confirm that
- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date
Zurich, 21 April 2014

Signature(s)