# Graph-Based Information Processing: Scaling Laws and Applications

THÈSE N$^{O}$ 5524 (2012)

PRÉSENTÉE LE 19 OCTOBRE 2012
À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE DE THÉORIE DES COMMUNICATIONS
PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Amin KARBASI

acceptée sur proposition du jury:

Prof. P. Thiran, président du jury
Prof. R. Urbanke, Prof. M. Vetterli, directeurs de thèse
Prof. A. Montanari, rapporteur
Prof. D. Shah, rapporteur
Prof. E. Telatar, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2012

*To the memory of my father, for all of his unconditional support.*

# Acknowledgements

Even though only my name appears on the cover of this thesis, many people have contributed to its existence. I owe my gratitude to all of the people who have made this journey nothing short of amazing.

First of all, I would like to express my sincere thanks to my thesis advisors, Prof. Rudiger Urbanke and Prof. Martin Vetterli. Without their constant support and encouragement, this thesis would have not been possible. It is hard to imagine having better advisors, one very keen in theory and the other more inclined towards practice. It is usually said that theory is when we know everything but nothing works and practice is when everything works but no one knows why. Throughout these years, I tried to combine theory with practice. The result was rather dramatic: nothing worked and I didn't know why.

I first met Rudiger in 2004 when I was still an undergraduate student. His course on "Principles of Digital Communication Systems" was one of the main reasons I changed my major from electrical engineering to computer and communication sciences, a decision that I do not regret (so far)! Since then, we have collaborated on various topics and research problems, an effort that has amazingly resulted in zero publications. Rudiger is not only a brilliant advisor but also a great friend. His trip to Iran, as well as all the video clips and photos he left behind, made my wedding even more special. As an additional positive side-effect (depending on whose perspective), whenever we had a disagreement or conflicting opinions, the matter was immediately resolved by me dropping a hint about the clips. Over the past couple of years, he has taught me more than I could have ever imagined: choosing the right research problems, publishing only those results that are worth publishing (this partially explains our astonishing record), and being an independent researcher. I would also like to thank him for his efforts to improve my writing skills, which were non-existent before. For all the above reasons, and numerous others, he will always remain one of the most influential people in my life.

I am deeply indebted to Martin. My first interaction with him was also in 2004 when I mistakenly attended his course on "Advanced Signal Processing" instead of the one I had intended, which was taught by his post-docs on "Statistical Signal Processing". Martin liked me so much that he wrote a very good recommendation letter to make sure that I would spend the next six months as an intern in Tokyo, Japan, as far away as possible from his office. Ever since I started my Ph.D. research, I have enormously benefited from his deep insight and his creative mind. Martin's unique sense of humor leaves no one untouched for more than ten seconds. If there were only one lesson that I learnt from Martin it would be to find my own green island instead of working on others' green transactions. If there were a second lesson it would certainly be again the one I just mentioned as he could have never emphasized enough the importance of this lesson. If there were a third lesson that would be not to take myself too seriously. I should also thank Martin for always trying to keep to a minimum my chaotic movements in research despite my best efforts to the contrary. I will never forget his advice on pursuing both depth and breath in research. Unlike with Rudiger, I managed to write a couple of papers with Martin. I should also mention our regular meetings that usually terminated with Martin saying, "Gooood, can I have a few minutes with Rudi?" To this date, I could not figure out what they were talking about.

I am particularly grateful to Dr. Laurent Massoulié and Dr. Startis Ioannidis, not only for hosting me in the Technicolor Research Lab for six months but also for having the patience to continue our collaborations thereafter. If it was not for them, a substantial part of this thesis would have simply been blank pages. Laurent has the amazing ability to see a research problem from both practical and theoretical points of view. Unlike me, he truly combines theory with practice. Similarly, I was astonished by Stratis' ability to simplify complicated proofs and to write them in comprehensible ways. I have been very fortunate to know them and to learn from them.

I would like to thank my committee members for having accepted to assess my thesis and taking the time to read it through: Prof. Andrea Montanari, Prof. Devavrat Shah, and Prof. Emre Telatar. I would also like to thank Prof. Patrick Thiran for being the president of the jury. I owe sincere and earnest thankfulness to Emre for all the hours we spent discussing a diverse set of subjects ranging from classical music and literature to information theory. I particularly remember one occasion when he read to me a short story, "The Dead", for two hours and explained all the details and hidden angels. We would usually end these meetings with ten to fifteen minutes discussing the research problems. Strangely enough, these few minutes provided me with enough ingredients to solve my problems in the following days. I also had the opportunity to be a teaching assistant for a course on "Dynamical Systems Theory" taught by Patrick. By observing him in the class, I learnt a great deal about teaching skills, for which I am very thankful.

I would like to offer my special thanks to my collaborators, Andrea Montanari and Sewong Oh (for our work on sensor localization), Hessam Salavati and Amin Shokrollahi (for our work on neural associative memories), Mahdi Cheraghchi, Sohiel Mohajer, Ali Hormati, and Venkatesh Saligrama (for our work on group testing), Morteza Zadimoghaddam (for our work on binary identification problem), Laurent Massoulié and Startis Ioannidis (for our work on socila networks and content search through comparisons), Akihiko Sugiyama (for our work on direction of arrival estimation), Reza Parhizkar (for our work on tomography calibration), Arash Amini (for our work on low-rank matrix approximation), and Milan Vojnovic (for our work on job scheduling for cloud computing).

I have been very fortunate to be surrounded by amazing friends who helped me stay sane through these years (at least this is my interpretation). First, Mahdi Jafari for always being available despite his long response-time reaction. Our great friendship has grown dramatically over the last ten years. It is easy to not notice (and I have witnessed many instances) his high intelligence and genuine creativity covered under his humble character. I am deeply grateful for all the time we spent together and all the memories we share. I would like to take this opportunity to thank Soheil Mohajer and Mahdi Cheraghchi on the same basis. I have always appreciated Soheil's acute observations and his quest for knowledge. Similarity, Mahdi's orthogonal comments have always been a source of joy. I particularity like Mahdi's immaculate research style and his high standard for publication. I greatly value my friendship with Hessam Salavati and our ongoing collaboration. Hessam has, among many exceptional traits, a certain growth mindset that allows him to reach significant outcomes through dedicated and hard work. I have also learnt a great deal about neural associative memories from him. Along the same line, I would like to acknowledge my dear friend Saeid Haghighatshoar. His vast knowledge in mathematics and his thirst for learning have always astonished me. Whenever I needed a book I would first check our library and then his office. This acknowledgement is not complete without mentioning the countless hours Reza Parhizkar, Mohammad Karzand, Ali Hormati, Vahid Aref and I wasted on playing foosball and Shelem. Vahid's great wit and unique personality, in addition to his singing skills, make him one of a kind. I would like to particularly thank Reza and Ali for not only being great friends but also for being amazing collaborators.

In addition to those mentioned above, I am very grateful to many friends who helped me with their various forms of support during my graduate study. You are too many to be listed here, but you know who you are!

I am thankful to all my officemates during these years: Hamed Hassani, Christine Neuberg, and Marc Desgroseilliers. One characteristic that they all share is their real comfort with cluttered desks. In addition, Hamed showed, every now and again, his great talent in collecting all of IPG's cups in our office. His talent coupled with Marc's speciality of being the only living person at EPFL who walks barefoot, should make a complete picture of our office over the years.

A good support system is vital for surviving in graduate school. My stay at EPFL was exceptionally comfortable due to Muriel Bardet and Francoise Behn. I particularly enjoyed our gossip sessions during coffee breaks, which were not only a golden source of information but also a crash course in French. I would also like to thank Damir Laurenzi for solving my many embarrassingly silly problems with Unix, despite his saying "no" at first sight (accompanied with Italian swears). I am also indebted to Marisa Wynn for too many reasons. In particular, she helped me find almost all the internships I did. I can never repay her kindness and consideration.

I am obliged to many of my colleagues in Information Processing Group (IPG) and Audiovisual Communications Laboratory (LCAV) who supported me throughout these years. Special thanks go to Shirin Saeedi, Mohammad Karzand, Olivier Leveque, Giovanni Cangiani, Nicolas Macris, Bixio Rimoldi, Andrei Giurgiu, Alla Merzakreeva, Nick Ruozzi, Marc Vuffray, Emmanuel Abbe, Mihailo Kolundzija, Francisco Pinto, Pedro Pinto, Andrea Ridolfi, Jayakrishnan Unnikrishnan, Feng Yang, Yann Barbotin, Ivan Dokmanic, Mitra Fatemi, Marta Martinez Camara, Juri Ranieri, and Farid Movahedi Naini.

Most importantly, I would not have made it this far without the love and patience of my family. I would like to express my heart-felt gratitude to my mother (Soghi), sisters (Elnaz and Sanaz), my father-in-law (Ali), my mother-in-law (Mali), and my brother-in-laws (Shahab, Ehsan, and Amin). I particularly thank my mother for her tireless determination that paved the way for my success in life. Undoubtedly, she is the strongest woman I know. It is hard to live up to the standards put by a mathematician, especially if he happens to be my father-in-law; whatever I prove must seem trivial to him. I am also very grateful to Antonino Catalanotto who has been like a brother to me and who taught me French through really bad jokes. I am also indebted to Giovanni Catalanotto who has always treated me like one of his sons.

Special recognition goes to my lovely wife, Marjan. There are no words to express and convey my gratitude and love for her. I truly thank her for sticking by my side and believing in me, even when I did not believe in myself. A notable example of her faith is when she mistakenly cooks for five people and expects me to finish almost all the food alone. If it was not for her, I would have never learnt to live life to the fullest. Marjan, no matter how many papers I publish, I will always love you.

This thesis is dedicated to my personal hero who taught me the value of work and the importance of character.

# Abstract

We live in a world characterized by massive information transfer and real-time communication. The demand for efficient yet low-complexity algorithms is widespread across different fields, including machine learning, signal processing and communications. Most of the problems that we encounter across these disciplines involves a large number of modules interacting with each other. It is therefore natural to represent these interactions and the flow of information between the modules in terms of a graph. This leads to the study of *graph-based information processing* framework. This framework can be used to gain insight into the development of algorithms for a diverse set of applications.

We investigate the behaviour of large-scale networks (ranging from wireless sensor networks to social networks) as a function of underlying parameters. In particular, we study the scaling laws and applications of graph-based information processing in sensor networks/arrays, sparsity pattern recovery and interactive content search.

In the first part of this thesis, we explore location estimation from incomplete information, a problem that arises often in wireless sensor networks and ultrasound tomography devices. In such applications, the data gathered by the sensors is only useful if we can pinpoint their positions with reasonable accuracy. This problem is particularly challenging when we need to infer the positions based on basic information/interaction such as proximity or incomplete (and often noisy) pairwise distances.

As the sensors deployed in a sensor network are often of low quality and unreliable, we need to devise a mechanism to single out those that do not work properly. In the second part, we frame the network tomography problem as a well-studied inverse problem in statistics, called *group testing*. Group testing involves detecting a small set of defective items in a large population by grouping a subset of items into different pools. The result of each pool is a binary output depending on whether the pool contains a defective item or not. Motivated by the network tomography application, we consider the general framework of group testing with graph constraints. As opposed to conventional group testing where *any* subset of items can be grouped, here a test is admissible if it induces a connected subgraph. Given this constraint, we are interested in bounding the number of pools required to identify the defective items. Once the positions of sensors are known and the defective sensors are identified, we investigate another important feature of networks, namely, *navigability* or how fast nodes can deliver a message from one end to another by means of local operations.

In the final part, we consider navigating through a database of objects utilizing comparisons. Contrary to traditional databases, users do not submit queries that are subsequently matched to objects. Instead, at each step, the database presents two objects to the user, who then selects among the pair the object closest to the target that she has in mind. This process continues until, based on the user's answers, the database can identify the target she has in mind. The search through comparisons amounts to determining which pairs should be presented to the user in order to find the target object as quickly as possible. Interestingly, this problem has a natural connection with the navigability property studied in the second part, which enables us to develop efficient algorithms.

**Keywords:** Sensor networks, Sparse recovery, Network tomography, Positioning, Calibration, Matrix completion, Group testing, Social networks, Small world, Navigability, Active learning, Binary search, Graphical models.

# Résumé

Le monde actuel est caractérisé par des transferts d'information massifs et des communications en temps réel. La demande pour des algorithmes efficaces et néanmoins peu complexes est présente dans différents domaines, en particulier l'apprentissage automatique, le traitement du signal et les télécommunications. La plupart des problèmes rencontrés dans ces disciplines mettent en jeu un grand nombre de modules interagissant. Il est donc naturel de représenter les interactions entre ces modules et le flot d'informations échangées à l'aide d'un graphe. Ceci nous amène à considérer le traitement de l'information du point de vue des graphes, ainsi qu'à développer des algorithmes pour diverses applications.

Nous explorons le comportement de réseaux à grande échelle (tels les réseaux de capteurs sans fil ou les réseaux sociaux) en fonction des paramètres sous-jacents du problème. En particulier, nous étudions des lois d'échelle et des applications du traitement de l'information basé sur les graphes à des réseaux ou tableaux de capteurs, à l'identification de structures de faible densité, ainsi qu'à la recherche interactive de contenu.

Dans la première partie de cette thèse, nous explorons le problème de localisation à partir d'informations incomplètes, qui surgit souvent dans les réseaux de capteurs sans fil et la tomographie à ultrasons. Dans de telles applications, les données rassemblées par les capteurs sont utiles seulement lorsqu'il est possible de localiser avec précision les positions exactes de ceux-ci. Ce problème se révèle être particulièrement difficile lorsqu'on doit déduire les positions à partir d'informations ou d'interactions simples telles que la proximité ou les distances mutuelles entre capteurs (souvent bruitées).

Du fait que les capteurs déployés dans un réseau sont souvent de faible qualité et pas forcément fiables, il est nécesssaire de développer des mécanismes pour identifier ceux qui ne fonctionnent pas correctement. Dans la seconde partie, nous formulons le problème de tomographie d'un réseau comme un problème de *test de groupe*, un problème bien étudié en statistiques. Le test de groupe consiste en la détection d'un petit ensemble d'éléments défectueux dans une grande population par regroupement des éléments en différents sous-ensembles. Chaque sous-ensemble émet une décision binaire indiquant ou non la présence d'un élément défectueux au sein du groupe. Motivés par l'application à la tomographie des réseaux, nous considérons le cadre général du test de groupe avec des contraintes sur le graphe. Dans notre cas, un test n'est admissible que s'il induit un sous-graphe connecté, contrairement au test de groupe conventionnel, où *n'importe quel* sous-ensemble peut être formé. Etant donné cette contrainte, nous sommes intéressés à majorer le nombre de groupes requis pour identifier les éléments défectueux. Une fois que les positions des capteurs sont connues et que les éléments défectueux sont identifiés, nous étudions une autre propriété importante des réseaux: leur *navigabilité*, à savoir la rapidité à laquelle les noeuds peuvent transmettre un message d'un bout à l'autre du réseau en effectuant seulement des opérations locales.

Dans la dernière partie, nous considérons la navigation dans une base de données au moyen de comparaisons. Contrairement aux bases de données traditionnelles, les utilisateurs ne soumettent pas des requêtes qui sont ensuite mises en correspondance avec des objets. Au lieu de cela, à chaque pas, la base de données présente deux objets à l'utilisateur, qui sélectionne celui le plus proche de la cible visée. Ce processus continue jusqu'à ce que la base de données puisse identifier la cible sur la base des réponses de l'utilisateur. Afin de rendre la recherche par comparaisons efficace, il importe de déterminer quelles paires présenter à l'utilisateur. De manière intéressante, ce problème a un lien naturel avec la propriété de navigabilité étudiée dans la deuxième partie, ce qui nous permet à développer des algorihmes efficaces pour la résolution du problème.

**Mots-clés:** Réseaux de capteurs, Identification de structures de faible densité, Tomographie des réseaux, Positionnement, Calibration, Complétion de matrice, Test de groupe, Réseaux sociaux, Phénomène dit du “petit monde”, Navigabilité, Apprentissage actif, Recherche binaire, Modèles graphiques.

# Contents

**Acknowledgements** **iii**

**Abstract** **vii**

**Contents** **xi**

**1 Introduction** **1**

**I Where Are We?** **9**

**2 Robust Positioning from Incomplete Information** **11**
- 2.1 Related work . . . . . 13
- 2.2 Model definition . . . . . 15
- 2.3 Algorithms . . . . . 17
- 2.4 Main results . . . . . 22
- 2.5 Analysis . . . . . 28
  - 2.5.1 Proof of Theorem 2.1 . . . . . 28
  - 2.5.2 Proof of Theorem 2.3 . . . . . 30
  - 2.5.3 Proof of Theorem 2.4 . . . . . 30
  - 2.5.4 Proof of Theorem 2.5 . . . . . 31

**3 Calibration Through Matrix Completion** **37**
- 3.1 Related Work . . . . . 38
- 3.2 Circular Time of Flight Tomography . . . . . 40
- 3.3 Problem Statement . . . . . 44
- 3.4 Matrix Completion . . . . . 46
- 3.5 Main Results . . . . . 47
- 3.6 Numerical Evaluation . . . . . 48
- 3.7 Analysis . . . . . 50
  - 3.7.1 Proof of Theorem 3.2 . . . . . 50
  - 3.7.2 Proof of Lemma 3.1 . . . . . 54
  - 3.7.3 Proof of Theorem 3.3 . . . . . 54

# II Where Is It? 57

4 Graph-Constrained Group Testing 59
4.1 Related Work . . . 63
4.2 Definitions and Notation . . . 64
4.3 Problem Statement . . . 67
4.4 Main Results . . . 68
4.5 Analysis . . . 72
4.5.1 Proof of Theorems 4.7 and 4.9 . . . 72
4.5.2 Proof of Theorem 4.8 . . . 80
4.5.3 The Erdős-Rényi Random Graph . . . 81
4.5.4 Expander Graphs with Constant Spectral Gap . . . 82

5 Navigability with a Bias 85
5.1 Related Work . . . 87
5.2 Definitions and Notation . . . 87
5.3 Problem Statement . . . 89
5.4 Main Results . . . 90
5.5 Analysis . . . 92
5.5.1 Proof of Theorem 5.2 . . . 92
5.5.2 Proof of Theorem 5.3 . . . 97

# III What Is It? 101

6 Content Search Through Comparisons 103
6.1 Related Work . . . 104
6.2 Definitions and Notation . . . 106
6.3 Problem Statement . . . 107
6.4 Relationship Between SWND and CSTC . . . 109
6.5 Main Results . . . 110
6.6 Learning Algorithm . . . 111
6.7 Analysis . . . 114
6.7.1 Proof of Theorem 6.1 . . . 114
6.7.2 Proof of Theorem 6.2 . . . 115
6.7.3 Proof of Theorem 6.3 . . . 116

7 Comparison-Based Active Learning 119
7.1 Definitions and Preliminaries . . . 120
7.2 Active Learning . . . 121
7.3 An Efficient Adaptive Algorithm . . . 122
7.4 Noisy Comparison Oracle . . . 124
7.5 Numerical Evaluation . . . 125
7.6 Analysis . . . 127
7.6.1 Proof of Lemma 7.3 . . . 127
7.6.2 Proof of Lemma 7.4 . . . 128
7.6.3 Proof of Lemma 7.5 . . . 129
7.6.4 Proof of Theorem 7.6 . . . 129

7.6.5 Proof of Theorem 7.7 . . . . . . . . . . . . . . . . . . . . . . . . . . 130

**8 Conclusion** **133**

**A Morse Code** **137**

**Bibliography** **139**

**Curriculum Vitae** **149**

“·–   ·––– ––– ··– ·–· –· · –·––   ––– ··–·   ·–
– ···· ––– ··– ··· ·– –· –··   –– ·· ·–·· · ···
··· – ·– ·–· – ···   ·–– ·· – ····   ·–   ··· ·· –· ––· ·–·· ·
··· – · ·––· ······”

–·–· ···· ·· –· · ··· ·   ·––· ·–· ––– ··– · ·–· –···

# Introduction

# 1

This thesis describes a framework for “graph-based information processing” that provides a mechanism for exploiting the graph structure in complex systems such as large scale networks. Such systems comprise different modules interacting with each other. One way to visualize these interactions is through a graph-based model where nodes represent the modules and edges the interactions.

An illustrative example is a wireless sensor network deployed in the Swiss Alps for environment monitoring [BISV08]. A wireless sensor network consists of spatially distributed sensors that monitor physical or environmental conditions such as temperature, vibration, and pressure. Moreover, these sensors should cooperatively pass their data through the network to a few locations. Due to energy constraints, each sensor can only communicate with a few of its close-by neighbours. As a result, we have a graph where nodes indicate the sensors and edges the communication links. We can even illustrate more complicated and perhaps more realistic scenarios with the same graph. For instance, two close-by sensors might not be able to detect each other’s presence due to weather conditions (e.g., snow, rain, you name it). Hence, we can associate a probability of detection if two nodes are within the communication range of one another. This probability may well decay as a function of the pairwise distance between the two sensors. To incorporate this phenomenon in our graph-based model, we simply associate a weight (between 0 and 1) to each link. Consequently, we obtain a weighted graph that indicates which nodes communicate with one another and also what the quality of communication is.

Note that in many applications of sensor networks, we have access to many sensors scattered randomly on the field (e.g., dropping them off from an airplane). These sensors are cheap and battery conservative. Once the sensors start working, their first task is to estimate their location. Clearly, without this information the data gathered by the sensors is of little values. For example, the data might indicate that somewhere in the Alps it is snowing (not a surprise!), but we do not know where because we failed to find the positions of the sensors detecting snow. There are many ways to go about solving this problem. We can simply equip each sensor with a GPS device. This not only increases the cost of the device but it is not even appropriate in some situations (e.g., no reception of a GPS signal because there is a big rock in front of the sensor). As a result, some sensors might know their exact locations (and they can live happily until their battery runs down), but most should somehow estimate them. It is not hard to see that if all pairwise distances were known, the positioning problem would be easy to solve. Unfortunately, due to energy constraints, only nearby sensors are able to communicate with one another and thus estimate their pairwise distances. Therefore, most of the pairwise distances are unknown. This uncertainty, coupled with the fact that even the known distances are noisy, makes the localization problem very challenging. Again, we can represent the problem through a graph-based model. Like before, nodes represent the sensors and edges the communication links. This time, however, the weight of a link indicates the pairwise distance between two sensors. To estimate the missing pairwise distances, we can simply calculate the shortest paths on the graph. This way, we will have a rough estimate of all the pairwise distances, based on the ones for which we can calculate the positions.

Another interesting example that shows how the available information is constrained by the graph structure is *network tomography*. Consider again the sensors deployed in the Swiss Alps for environment monitoring. Recall that these sensors are cheap and affected by the changes in their surroundings. This simply implies that every now and then, one or more sensors (hopefully not too many) will malfunction, e.g., drop all the packets. Hence, it is important to single out the defective sensors on a regular basis. One naive way is to test periodically every single sensor. This seems like a waste of energy and time, since only a small fraction of sensors are defective. It also requires direct access to sensors at all times. An alternative way that has recently gained a lot of attention is to make use of collective end-to-end measurements [NT07, CCL$^+$04]. To do so, we can imagine the following protocol. Assume that at a particular node $s$, we wish to identify the set of defective sensors by distributing packets that originate from $s$ in the network. First, $s$ generates a packet containing a time stamp $t$ and sends it to a randomly chosen neighbour, who in turn, decrements the time stamp and forwards the packet to a randomly chosen neighbour. The process continues until the time stamp reaches zero, at which point the packet is sent back to $s$ along the same path it has traversed. This can be achieved for example by storing the route to be followed (which is randomly chosen at $s$) in the packet. By using this procedure, the source node generates a number of independent packets, distributed in the network. Each packet is either returned back to $s$ in a timely manner, or eventually does not reach $s$ due to the presence of a defective sensor within the route. By choosing an appropriate timeout, $s$ can determine the packets that are routed through the defective sensors. Note that this scheme tries to make use of group tests in order to identify the faulty sensors. However, not any group test is possible, i.e., each group here must conform to the constraints imposed by the graph structure of the network.

Our final example to illustrate the importance of graph structure in information processing is *geographic routing*. Again, let us look back to our sensor network scenario. At the end of the day, the goal of deploying such a complex system is to gather information. To this end, sensors should be able to route collectively the information from one end to the other. As we mentioned earlier, the sensors can only detect their close-by neighbours and are not usually aware of the rest of the network. In such settings, a sensor that holds a message should operate based only on local information. One method that stands out as an efficient routing algorithm is *greedy forwarding*, in which each node forwards the message to the neighbour that is most suitable from a local point of view. The most suitable neighbour can be the one who minimizes the distance to the destination in each step. This algorithm is of course sub-optimal with respect to an algorithm that has the full view of the network and routes the information via shortest paths. However, a greedy forwarding protocol possesses a number of important properties. First, it is a fully distributed algorithm. Second, the system is very robust against changes in the topology of the network; sensors can be added or removed arbitrarily. And third, it is a low complexity algorithm, i.e., each node only needs to know among its neighbours which one is closest to the target.

The issue raised above, namely routing, is also present in the context of social networks such as Facebook or MySpace. These are networks in which nodes are people (or groups of people) and edges represent social interactions such as friendships, professional relationships, or many other types of connections. One of the most important features to appreciate about social networks is the *small-world phenomenon.* To understand the concept, let us conduct the following thought experiment [EK10]. First, find a friend of yours who grew up in another country. Following a path through this friend, to his/her parents, and to their friends, it is very likely that you end up in a different part of the world in only three steps. In my case for example, if I consider my Swiss friend Antonino, his Italian father Geovanni, and his father's friends who live in Sicily, I will end up with the people from another generation who live far from Tehran and have very little in common with me. This idea that the world looks small and it takes a short chain of friends to get from you to anyone else on the planet has been termed the small-world phenomenon or in popular culture referred to as the *six degrees of separation.* It is a matter of common experience to see that sending a message or spreading a rumour across a large network takes only a few steps. As our simple thought experiment and of course many empirical studies reveal, in social networks short paths exist in abundance. What is even more surprising is that people are remarkably good at finding them. Of course if a person knows the structure of an entire network (who is friends with whom) finding short paths is a matter of computation. Participants of a social network such as Facebook, however, do not know the whole network and are probably only aware of a very small part. Yet they are still able to route a message rapidly to the desired target. As Kleinberg observed, it is possible and indeed common to construct a network that has short paths between nodes. Nevertheless finding these short paths for participants lacking the full view of the network is very hard [Kle00]. Hence, not all networks with a small diameter enjoy the small-world phenomenon. As suggested by Kleinberg, a network should possess a particular structure so that nodes can route messages efficiently without a global knowledge of the network.

# Outline

In the following we present a brief summary of the individual problems that are studied in each chapter.

## Robust Positioning from Incomplete Information

Obtaining a good estimation of where wireless devices are located is crucial in wireless network applications including environment monitoring, geographic routing and topology control. When the positions of the devices are unknown and only local distance information is given, we need to infer the positions from these local distance measurements. This problem is particularly challenging when we have access only to measurements that have limited accuracy and are incomplete. In Chapter 2, we look at the problem of localizing wireless devices in an ad-hoc network embedded in a $d$-dimensional Euclidean space. We consider the extreme case of this limitation on the available information, when only the connectivity information is available, i.e., we only know whether a pair of nodes is within a fixed detection range of each other or not, and we have no information about how far apart they are. Furthermore, to account for detection failures, we assume that even if a pair of devices is within the detection range, it fails to detect the presence of one another with some probability and this probability of failure depends on how far apart those devices are. We are mainly interested in answering the following question:

> Given this limited information, how well can we estimate the position of the sensors?

To answer this question, we investigate the performance of a centralized positioning algorithm MDS-MAP introduced by Shang et al. [SRZF03], and its distributed version Hop-TERRAIN introduced by Savarese et al. [SLR02]. In particular, for a network consisting of $n$ devices positioned randomly, we provide a bound on the resulting error for both algorithms. We show that the error is bounded, decreasing at a rate that is proportional to $R_{\text{Critical}}/R$, where $R_{\text{Critical}}$ is the critical detection range when the resulting random network starts to be connected, and $R$ is the detection range of each device.

## Calibration Through Matrix Completion

In Chapter 3, we study the calibration process in circular ultrasound tomography devices based on Time-of-Flight (ToF) measurements. Our goal is to accurately estimate the positions of the sensors deviating from the circumference of a perfect circle. This problem arises in a variety of applications in signal processing such as breast imaging for cancer detection. In the presence of all the pairwise ToF measurements, we can easily estimate the sensor positions using multi-dimensional scaling (MDS) method. In practice however, due to the transient behaviour of the sensors and the beam shaper of the transducers, the ToFs for close-by sensors are unavailable. Furthermore, random malfunctioning of the sensors leads to random missing ToF measurements. In addition to the missing entries, we observe that an unknown time delay is also added to the measurements. The question is:

> Faced with the set of missing ToFs and the unknown time delay, how well can we calibrate a circular ultrasound tomography device?

To answer this question, we incorporate the fact that a matrix defined by all the ToFs measured in a homogeneous medium is of rank at most 4. In order to estimate the missing ToFs, we apply a state-of-the-art low-rank matrix completion algorithm, OptSpace. To find the correct positions of the sensors (our ultimate goal), we then apply MDS. We show analytic bounds on the overall error of the entire process in the presence of noise.

## Graph-Constrained Group Testing

Inverse problems, with the goal of recovering a signal from partial and noisy observations, come in many different formulations and arise in many applications. One important property of an inverse problem is that it should be well-posed, i.e., there should exist a unique and stable solution to the problem. In this respect, prior information about the solution, like sparsity, can be used as a “regularizer” to transform an ill-posed problem into a well-posed one. In Chapter 4, we look at a well-studied inverse problem in statistics, called “group testing”, where we try to identify the set of defective items in a large population. The main idea is that, instead of testing items one by one, it might be possible to pool them in large groups, and then apply the tests on the groups without affecting reliability of the tests. A notable example is the method proposed in World War II to weed out the soldiers carrying syphilis. It was proposed to combine blood samples drawn from individuals and test the combined samples for syphilis. Once a group is tested negative, all the members participating in the group must be negative and this may save a large number of tests. Otherwise, a positive test reveals that at least one of the individuals in the group must be positive. A fundamental question involves

> How to design a pooling strategy that allows for the identification of the exact set of infected items by using as few tests as possible.

Inspired by applications in network tomography, sensor networks and infection propagation, in Chapter 4, we formulate a variation of group testing problems on graphs. Unlike conventional group testing problems, each group here must conform to the constraints imposed by a graph. For instance, items can be associated with vertices and each pool is any set of nodes that is path connected. This way, a test is simply associated with a random walk. In this context, conventional group testing corresponds to the special case of a complete graph on $n$ vertices.

For interesting classes of graphs we obtain a rather surprising result: the number of tests required to identify $d$ defective items is substantially similar to what is required in conventional group testing problems, where no such constraints on pooling is imposed. Specifically, if $T(n)$ corresponds to the mixing time of the graph $G$, it is shown that with $m = O(d^2T^2(n)\log(n/d))$ non-adaptive tests, we can identify the defective items. Consequently, for the Erdős-Rényi random graph $G(n, p)$, as well as expander graphs with constant spectral gap, it follows that $m = O(d^2 \log^3 n)$ non-adaptive tests are sufficient to identify $d$ defective items. We also consider the noisy counterparts and develop parallel results.

## Navigability with a Bias

Routing information through networks is one of the fundamental tasks in many complex systems. When each node has a full view or global map of the network, finding the shortest route is only a matter of computation. In many networks, including social, peer-to-peer and neural networks, even though most nodes are not neighbours of each other, they can still reach one another through a small number of intermediate nodes. In Chapter 5, we look at this phenomenon, known as the *small-world phenomenon*. The fact that nodes can reach each other rather quickly suggests that short routes exist in abundance. Many models in the literature are proposed to capture this property [WS98, Wat99, BC88]. However, as Kleinberg observed [Kle01], the existence of such short paths does not imply that a decentralized algorithm such as greedy forwarding can find them. In Chapter 5, we consider the following general framework for designing small-world networks:

> Given a graph embedded in a metric space, how should we augment this graph by adding edges in order to minimize the expected cost of greedy forwarding over this graph?

We consider the above problem under the scenario of *heterogeneous demand*, i.e., the probability that a message is sent from a node to another is not uniform. We first show that the small-world network design problem under the general heterogeneous demand is NP-hard. We then propose a novel mechanism for edge addition and provide an upper bound on the performance of the greedy forwarding in terms of the bias of the demand distribution, as well as the embedding.

### Content Search Through Comparisons

In the problem of content search through comparisons, a user searching for a target object navigates through a database in the following manner: the user is asked to select the object most similar to her target from a small list of objects. A new object list is then presented to the user based on her earlier selection. This process is repeated until the target is included in the list presented, at which point the search terminates. Now, the main question is the following:

> Which objects should be presented to the user in order to find the target object as quickly as possible?

This problem is strongly related to the small-world network design. More precisely, one way of searching over the database is to establish a small-world network over the target set and employ an oracle to successively progress closer to the target object through greedy forwarding. In Chapter 6, we adopt this approach. Contrary to prior work, which focuses on cases where objects in the database are equally popular, we consider the case where the demand for objects may be heterogeneous, i.e., objects are requested with different frequencies. We provide an algorithm for deciding which sequence of objects to show to the user and bound its cost in numbers of queries. We also provide a lower bound on the cost among all such algorithms. Our upper and lower bounds relate the cost of content search to two important properties of the demand distribution, namely its entropy and its doubling constant. These illustrate interesting connections between content search through comparisons to classic results from information theory. Finally, based on these results, we propose an adaptive learning algorithm for content search that meets the performance guarantees achieved by the above mechanisms.

### Comparison-Based Active Learning

In Chapter 7, we consider again the problem of search through comparisons. Our main objective is to answer the following question:

> In the case of noisy user feedback, is it still possible to navigate through the database robustly?

To answer this question we first propose a new strategy based on rank nets and show that, for target distributions with a bounded *doubling constant*, it finds the target in a number of comparisons close to the entropy of the target distribution, hence of the optimum. We then consider the case of noisy user feedback. In this context, we propose a variant of rank-nets, for which we establish performance bounds within a slowly growing function (doubly logarithmic) of the optimum.

## How to (not) Read This Thesis

The material presented in each of the technical chapters of this thesis (Chapters 2–7) are intended to be self-contained so that they can be read in any order. However, for the interested reader who wants to gain more insight about our work, we modestly suggest the following order. Our first suggestion is to read Chapter 2 before Chapter 3. Both chapters look at the localization problem, albeit in a different context. Following the same philosophy, it is more informative to read Chapter 6 before Chapter 7. To better understand the link between the small network design and content search through comparisons, we believe that Chapter 5 should be skimmed through prior to Chapter 6.

Throughout this thesis, we do our best to follow the same structure. Each chapter begins with an introduction followed by related work. Then the required notation is introduced and the problem setting is stated. Thereafter, the main results are explained in detail. We also provide intuition on the results. The proofs are all presented at the end of the chapters in an attempt to help the reader grasp the main ideas behind our work and not lose sight of the forest for the trees.

The additional mathematical background required for each chapter is provided when needed, to the extent of not losing focus. For a comprehensive study of the basic tools used, we refer the reader to [CT91] (information theory), [New10] (network models), [MR95, MU05] (probabilistic methods), [KF09] (graphical models), [HLW06] (expander graphs), [AB09] (complexity theory), and [DH99] (group testing).

Each chapter opens up with a quotation. To understand the phrase, the reader needs to be familiar with the **morse code** (see the Appendix). After all, this is a thesis on information processing.

# Part I

# Where Are We?

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

[illegible]

# Robust Positioning from Incomplete Information

# 2

In this chapter [1], we address the problem of positioning when only partial information on pairwise distances is provided. Location estimation of individual nodes is required for many wireless sensor network applications such as environmental monitoring, geographic routing and topology control, to name a few [JH01, Xu02]. In environmental monitoring, for instance, the measurement data by the wireless sensor network is more useful when accompanied by the location information. One way to acquire the positions is to equip all the sensors with a global positioning system (GPS). The use of GPS not only adds considerable cost to the system, but more importantly, it does not work in indoor environments or when the received GPS signal is jammed [CHH02]. Alternatively, we can use an algorithm that derives positions of sensors based on local and basic information such as proximity (which nodes are within communication range of each other) or local distances (pairwise distances between neighbouring sensors).

Two common techniques for obtaining the local distance and connectivity information are Received Signal Strength Indicator (RSSI) and Time Difference of Arrival (TDoA). RSSI is a measurement of the ratio of the power present in a received radio signal and a reference power. Signal power at the receiving end decreases as a function of the distance, and RSSI has the potential to be used to estimate the distance. Altenatively, TDoA techniques use the time difference between the receipt of two different signals with different velocities, for instance ultrasound and radio frequency signals [PCB00, SHS01]. These techniques can be used, independently or together, for distance estimation.

Given a set of such measurements, we want to find the positions. One common approach, known as multi-dimensional scaling (MDS) [BG05], assumes that all pairwise distances are known. However, in almost all practical scenarios such information is unavailable for two major reasons. First, sensors are typically highly resource-constrained (e.g., power) and have limited communication range. Thus, distant sensors cannot communicate to obtain their pairwise distances. Second, due to noise and interference among sensors, there is always the possibility of non-detection or completely incoherent measurements.

---

[1] This chapter is the result of a collaboration with S. Oh and A. Montanari.

Many algorithms have been proposed to resolve these issues by using heuristic approximations to the missing distances, and their success has mostly been measured experimentally. Regarding the mechanisms deployed for estimating sensor locations, one can divide the localization algorithms into two categories: range-based and range-free. In the range-based protocols the absolute point-to-point distance estimates are used for inferring the locations, whereas in the range-free protocols no assumptions about the availability of such information are made and only the connectivity information is provided. As a result, range-free algorithms are more effective in terms of stability and cost, hence more favourable to be deployed in practical settings. In the sensor network literature, "range-free" is also referred to as "connectivity-based", and "range-based" is also referred to as "range-aware".

The theoretical guarantees associated with the performance of the existing methods are, however, of the same interest and complementary in nature. Such analytical bounds on the performance of localization algorithms can provide answers to practical questions: for example, "How large should the radio range be in order to get the reconstruction error within a threshold?" With this motivation in mind, our work takes a step forward in this direction.

We first focus on providing a bound on the performance of a popular localization algorithm MDS-MAP [SRZF03] when applied to sensor localization from only connectivity information. We should stress here that pairwise distances are invariant under rigid transformations (rotation, translation and reflection). Hence, given connectivity information, we can only hope to determine the *configuration* or the relative map of the sensors. In other words, localization is possible only up to rigid transformations. With this point in mind, we prove that using MDS-MAP, we are able to localize sensors up to a bounded error in a connected network where most of distances are missing and only local connectivity information is given.

More precisely, assume that there are $n$ sensors positioned randomly in a $d$-dimensional unit cube with the radio range $R = o(1)$. Further, assume that each pair can detect each other with probability at least $p_0$. Let the $n \times d$ matrices $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ denote the true sensor positions and their estimates by MDS-MAP, respectively. Define $\boldsymbol{L} = \mathbb{I}_{n\times n} - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ where $\mathbb{I}_{n\times n}$ is the identity matrix and $\mathbb{1}_n$ is the all ones vector. The quantity $\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L}$ satisfies nice properties: $(a)$ it is invariant under rigid transformations and $(b)$ if $\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} = \boldsymbol{L}\hat{\boldsymbol{X}}\hat{\boldsymbol{X}}^T\boldsymbol{L}$, then $\boldsymbol{X}$ and $\hat{\boldsymbol{X}}$ are equal up to rigid transformations. Therefore, a natural distance metric is:

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = \frac{1}{n}\left\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\right\|_F,$$

where $\|\cdot\|_F$ denote the Frobenius norm. Using this, we establish an upper bound on the error of MDS-MAP:

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq \frac{R_{\text{MDS}}}{R} + o(1),$$

where $R_{\text{MDS}} = C_d(\ln(n)/(p_0\, n))^{1/d}$ for some constant $C_d$ that only depends on the dimension $d$.

One consequence of the ad-hoc nature of the underlying networks is the lack of a central infrastructure. This hinders the use of a centralized algorithm like MDS-MAP. In particular, centralized algorithms suffer from scalability problems, making them difficult to implement in large scale sensor networks. Centralized algorithms also require higher computational complexity [MFA07]. This leads us to investigate if similar performance guarantees can be obtained in a distributed setting, where each sensor tries to estimate its own *global* position. As mentioned above, this task cannot be accomplished unless some additional information, other than local measurements, is provided. It is well known that in a $d$-dimensional Euclidean space, we need to know the global positions of at least $d+1$ sensors, referred to as *anchors*, in order to uniquely determine the global positions [NN01].

Under such decentralized scenario, we analyze the performance of a popular localization algorithm: HOP-TERRAIN [SLR02]. This algorithm can be seen as a distributed version of MDS-MAP. We prove that HOP-TERRAIN can localize sensors up to a bounded error in a connected network where most of the pairwise distances are unknown and only local connectivity information is given.

Similarly as in the case of MDS-MAP, assume $n$ sensors in a $d$-dimensional unit cube and $d+1$ anchors in general positions. We show that when only connectivity information is available, the Euclidean distance between the estimate $\hat{x}_i$ and the correct position $x_i$ is bounded by

$$\|x_i - \hat{x}_i\| \leq \frac{R_{\text{HOP}}}{R} + o(1),$$

for all $i$ where $R_{\text{HOP}} = C'_d(\log n/(p_0 n))^{\frac{1}{d}}$ for some constant $C'_d$ that only depends on $d$. One main contribution of our work in contrast to previous work is that we consider a general scenario where two sensors within a communication radio range may still fail to detect the presence of each other due to hostile environment or sensor malfunction.

## 2.1 Related work

The localization problem has attracted significant research interests in recent years. A general survey of the area and an overview of recent techniques can be found in [NN01] and [MFA07]. In the case when all pairwise distances are known, the coordinates can be derived by using a classical method known as multidimensional scaling (MDS) [BG05]. The underlying principle of MDS is to convert distances into an inner product matrix, whose singular vectors are the unknown coordinates. In the presence of noise, MDS tolerates errors gracefully due to the overdetermined nature of the problem. However, when most distances are missing, finding coordinates becomes more challenging. Three types of practical centralized algorithms have been proposed in the literature. The first group consists of algorithms that try first to estimate the missing entries of the distance matrix and then apply MDS to the reconstructed distance matrix to find the coordinates of the sensors. MDS-MAP, introduced in [SRZF03] and further studied in [SRZF04], is a well-known example, where the shortest paths are computed in order to approximate the missing distances. The algorithms in the second group mainly consider the sensor localization as a non-convex optimization problem and directly estimate the coordinates of sensors. A famous example of this type is a relaxation to semidefinite programming (SDP)[BY04]. In the third group, the problem is formulated through a stochastic optimization where the main technique used in these algorithms is the stimulated annealing, which is a generalization of the Monte Carlo method in combinatorial optimization [KMV06, KM06].

Perhaps a more practical and interesting case is when there is no central infrastructure. [LR03] identifies a common three-phase structure of three popular distributed sensor-localization algorithms, namely robust positioning [SLR02], ad-hoc positioning [NN03] and N-hop multilateration [SPS03]. Table 2.1 illustrates the structure of these algorithms. In the first phase, nodes share information to collectively determine the distances from

Table 2.1 – Distributed localization algorithm classification [LR03]

| Phase | Robust positioning | Ad-hoc positioning | $N$-hop multilateration |
|---|---|---|---|
| 1. Distance | DV-HOP | Euclidean | Sum-dist |
| 2. Position | Lateration | Lateration | Min-max |
| 3. Refinement | Yes | No | Yes |

each of the nodes to a number of anchors. Anchors are special nodes with a priori knowledge of their own position. In the second phase, nodes determine their position based on the estimated distances to the anchors. In the last phase, the initial estimated positions are iteratively refined. It is empirically demonstrated that these simple three-phase distributed sensor-localization algorithms are robust and energy-efficient [LR03]. However, depending on which method is used in each phase, there are different tradeoffs between localization accuracy, computation complexity and power requirements.

The performances of these algorithms are measured through simulations and little is known about their theoretical performance. A few exceptions are in the following work. In [DJMI$^+$06] the authors use matrix completion methods [Faz02] as a means to reconstruct the distance matrix. The main contribution of their paper is that they are able to provably localize the sensors up to a bounded error. However, their analysis is based on a number of strong assumptions. First, they assume that even far-away sensors have a non-zero probability of detecting their distances. Second, the algorithm explicitly requires the knowledge of detection probabilities between all pairs. Third, their theorem only works when the average degree of the network (i.e., the average number of nodes detected by each sensor) grows linearly with the number of sensors in the network.

Our first result on the analysis of MDS-MAP provides a theoretical guarantee that backs up experimental results. We use shortest paths as our primary guess for the missing entries in the distance matrix and apply MDS to find the relative positions of the nodes up to a rigid motion. In contrast to [DJMI$^+$06], we require significantly weaker assumptions. More specifically, we assume that only neighbouring sensors have information about each other and that only connectivity information is known. Furthermore, for the purpose of estimating the positions, the algorithms presented in this chapter do not require the knowledge of the detection probability. And last, in our analysis we assume that the average degree grows logarithmically, instead of linearly, with the number of sensors, which results in needing many less revealed entries in the distance matrix. On the one hand, we would like to choose the radio range large enough such that the graph is connected. Otherwise, we would be wasting the part of the graph that is not connected to the giant component. On the other hand, we would like to use as small a radio range as possible to save power consumption and operation cost. Hence, choosing a radio range which gives the average number of neighbors of order $\log n$ is desired. We provide the first error bounds on the performance of MDS-MAP.

Of particular interest are the two new results on the performance of sensor localization algorithms. In [JM11], Javanmard et al. proposed a new reconstruction algorithm based on semidefinite programming where lower and upper bounds on the reconstruction errors of their algorithm are established. Similarly, in [KOPV10], through new advances in matrix completion methods [CR08], the authors analysed the performance of OptSpace [KM10], a novel matrix completion algorithm, in localizing the sensors. Interestingly, they did not need to adhere to the assumptions made by [DJMI$^+$06]. However, they have a restrictive assumption about the topology of the network: sensors are scattered inside an annulus.

The above analytical results crucially rely on the fact that there is a central processor with access to the distance measurements. However, centralized algorithms suffer from the scalability problem and require higher computational power. Hence, a distributed algorithm with a similar performance bound is desirable. In our second result, we analyse the reconstruction error of a distributed algorithm. To the best of our knowledge, we show for the first time that HOP-TERRAIN, introduced in [SLR02], achieves a bounded error when only local connectivity information is given.

## 2.2 Model definition

In this section, we define a probabilistic model considered in this chapter. We assume we have no fine control over the placement of the sensors that we call *unknown nodes* (e.g., the nodes are dropped from an airplane). Hence, $n$ nodes are placed uniformly at random in a $d$-dimensional cube $[0,1]^d$. We assume that there are $m$ special sensors, which we call *anchors*, with a priori knowledge of their own positions in some global coordinate. In practice, it is reasonable to assume that we have some control over the position of anchors. Anchors can be some nodes that are planted on the field before any positioning takes place. Let $V_a = \{1,\dots,m\}$ denote the set of $m$ vertices corresponding to the anchors and $V_u = \{m+1,\dots,m+n\}$ the set of $n$ vertices corresponding to the unknown nodes. We use $x_i$ to denote the position of node $i$ and $\boldsymbol{X} \in \mathbb{R}^{n\times d}$ to denote the position matrix where the $i$-th row corresponds to $x_i$.

In positioning, due to attenuation and power constraints, only measurements between close-by nodes are available. As a result, the pairwise distance measurements can be represented by a random geometric graph $G(n+m,R) = (V,E,P)$, where $V = V_u \cup V_a$, $E \subseteq V \times V$ is a set of edges that connect pairs of sensors that are detected, and $P : E \to \mathbb{R}^+$ is a non-negative real-valued function. Edge weights $P$ is a mapping from an edge $(i,j)$ to a pairwise distance measurement.

A common model for this random geometric graph is the disc model where node $i$ and $j$ are connected if the Euclidean distance $d_{i,j} \equiv \|x_i - x_j\|$ is at most a positive radio range $R$. There are a variety of ways to measure the connectivity between two nodes, including TDoA and RSSI. Due to limited resources, there is a probability of non-detection. Think of RF ranging in the presence of an obstacle or in the case of multiple paths. Depending on the acquisition mechanism, this may result in the absence of measurements.

To model this failure of detection, we assume that two nodes can detect each other with a probability that only depends on the distance $d_{i,j}$. Namely, $(i,j) \in E$ with probability $p(d_{i,j})$ if $d_{i,j} \le R$. The detection probability $p(\cdot) : [0,R] \to [0,1]$ is a non-increasing function of the distance. Our main results on the error achieved by localization algorithms assumes that the detection probability $p(\cdot)$ is lower bounded by a parameter $p_0$ such that $p(z) \ge p_0$ , for $z \le R$. For example, $p(\cdot)$ might be a simple function parameterized by two scalar values $p_0$ and $\beta$:

$$p(z) = \min\left(1, p_0\left(\frac{z}{R}\right)^{-\beta}\right) , \tag{2.1}$$

for some $p_0 \in (0,1]$ and $\beta$. This includes the disc model with perfect detection as a special case (i.e., $p_0 = 1, \beta = 0$).

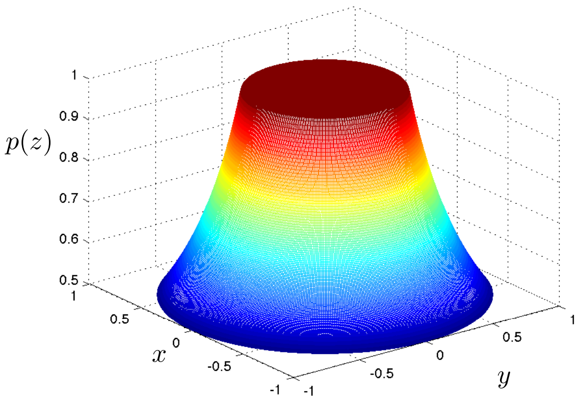


Figure 2.1 – This plot shows how the probability of detection changes as the distance between two sensor changes.

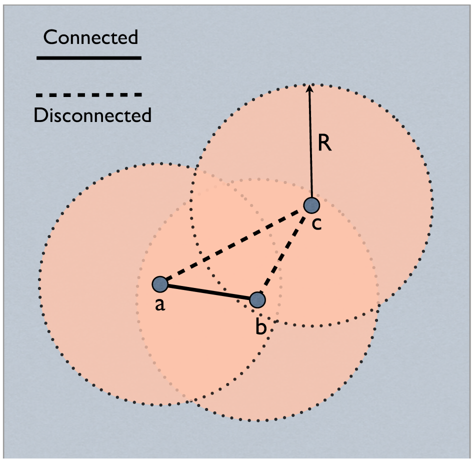


Figure 2.2 – This example shows the model. Nodes $a$ and $b$ are connected since they are within $R$. Although $b$ and $c$ are also within $R$, they are not connected due to detection failure. $a$ and $c$ are not connected because they are far apart.

To each edge $(i, j) \in E$, we associate the distance measurement $\boldsymbol{P}_{i,j}$ between sensors $i$ and $j$. In an ideal case, we have exact distance measurements available for those pairs in $E$. This is called the *range-based model*. Formally, $\boldsymbol{P}_{i,j} = d_{i,j}$ if $(i, j) \in E$ and $*$ otherwise, where a $*$ denotes that the distance measurement is unavailable. In the following, the algorithms presented in this chapter only uses values of $\boldsymbol{P}_{i,j}$'s where it is well defined (i.e. for $(i, j) \in E$, and equivalently for $\boldsymbol{P}_{i,j} \neq *$). In this chapter, we assume that we are given only network connectivity information and no distance information. This is known as the *connectivity-based model*. Formally,

$$\boldsymbol{P}_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ * & \text{otherwise.} \end{cases}$$

In the following, let $\boldsymbol{D}$ denote the $n \times n$ distance matrix where $\boldsymbol{D}_{i,j} = d_{i,j}$, and let $\bar{\boldsymbol{D}}$ denote the $n \times n$ *squared* distance matrix where $\bar{\boldsymbol{D}}_{i,j} = d^2_{i,j}$. By definition, $\bar{\boldsymbol{D}} =$

Table 2.2 – Summary of Notation.

| | | | |
|---|---|---|---|
| $n$ | number of unknown sensors | $V_u$ | set of unknown nodes |
| $m$ | number of anchors | $V_a$ | set of anchors |
| $R$ | communication range | $\mathbb{1}_n$ | all ones vector of size $n$ |
| $\boldsymbol{P}_{i,j}$ | distance measurements | $\widehat{\boldsymbol{D}}$ | estimated squared distance matrix |
| $d_{i,j}$ | Pairwise distance between nodes $i$ and $j$ | $\mathbb{I}_{n\times n}$ | $n \times n$ identity matrix |
| $x_i$ | position of node $i$ | $\hat{x}_i$ | estimated position of node $i$ |
| $p_0$ | minimum detection probability | $\boldsymbol{X}$ | positions matrix |
| $d$ | dimension | $\widehat{\boldsymbol{X}}$ | estimated positions matrix |
| $\bar{\boldsymbol{D}}$ | squared distance matrix | $\hat{d}_{i,j}$ | shortest path between node $i$ and $j$ |
| $\mathsf{O}(d)$ | orthogonal group of $d \times d$ matrices | $\lVert \cdot \rVert_F$ | Frobenius norm |
| $\langle \boldsymbol{A}, \boldsymbol{B} \rangle$ | Frobenius inner product | $\lVert \cdot \rVert_2$ | spectral norm |

$a\mathbb{1}_n^T + \mathbb{1}_n a^T - 2\boldsymbol{X}\boldsymbol{X}^T$, where $a \in \mathbb{R}^n$ is a vector with $a_i = \lVert x_i \rVert^2$ and $\mathbb{1}_n$ is the all ones vector. As $\bar{\boldsymbol{D}}$ is a sum of two rank-1 matrices and a rank-$d$ matrix, its rank is at most $d + 2$.

## 2.3 Algorithms

Depending on the application, we might want a *relative map* or an *absolute map* of the locations. A relative map is a configuration that have the same neighborhood relationships as the underlying graph $G$. In the following we use the terms configuration and relative map interchangeably. An absolute map, on the other hand, determines the absolute geographic coordinates. In this chapter, our objective is two-fold. First, we present a centralized algorithm MDS-MAP, that finds a configuration that best fits the proximity measurements. Then, we discuss its distributed version, HOP-TERRAIN, where the goal is for each sensor to find its absolute position.

### 2.3.1 Centralized Positioning Algorithm: MDS-MAP

The centralized MDS-MAP algorithm, assumes no anchors in the system. We denote the random positions of $n$ sensors by $x_i$'s. MDS-MAP consists of two steps:

**Algorithm 2.1** MDS-MAP [SRZF03].

```
Input: dimension $d$, graph $G = (V, E, P)$.
1: Compute the shortest paths, and let $\widehat{\boldsymbol{D}}$ be the squared shortest paths matrix.
2: Apply MDS to $\widehat{\boldsymbol{D}}$, and let $\widehat{\boldsymbol{X}}$ be the output.
```

**Shortest paths.** The shortest path between nodes $i$ and $j$ in graph $G = (V, E, P)$ is defined as a path between two nodes such that the sum of the proximity measures of its constituent edges is minimized. Let $\hat{d}_{i,j}$ be the computed shortest path between node $i$ and $j$. Then, the squared shortest paths matrix $\widehat{\boldsymbol{D}} \in \mathbb{R}^{n\times n}$ is defined as $\widehat{\boldsymbol{D}}_{ij} = \hat{d}_{i,j}^2$ for $i \neq j$, and 0 for $i = j$.

**Multidimensional scaling.** In step 2, we apply the Multidimensional scaling (MDS) to $\widehat{\boldsymbol{D}}$ to get a good estimate of $\boldsymbol{X}$, specifically, we compute $\widehat{\boldsymbol{X}} = \mathsf{MDS}_d(\widehat{\boldsymbol{D}})$. MDS refers

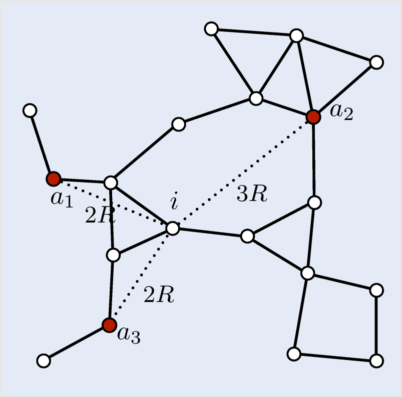


Figure 2.3 – The shortest path between two nodes is defined in terms of the minimum number of hops multiplied by the radio range $R$.

to a set of statistical techniques used in finding the configuration of objects in a low dimensional space such that the measured pairwise distances are preserved [BG05]. It is often used for a visual representation of the proximities between a set of items. Formally, MDS finds a lower dimensional embedding $\hat{x_i}$s that minimize the *stress*:

$$\text{stress} \equiv \sqrt{\frac{\sum_{i \neq j}(f(d_{i,j}) - \hat{d}_{i,j})^2}{\sum_{i \neq j} \hat{d}_{i,j}^2}} \, ,$$

where $d_{i,j}$ is the input similarity (or dissimilarity), $\hat{d}_{i,j} = \|\hat{x_i} - \hat{x_j}\|$ is the Euclidean distance in the lower dimensional embedding, and $f(\cdot)$ is some function on the input data. When MDS perfectly embeds the input data, we will have $f(d_{i,j}) = \hat{d}_{i,j}$ and the stress is zero.

**Algorithm 2.2** Classic Metric MDS [SRZF03].

```
Input: Dimension $d$, estimated squared distance matrix $\widehat{\boldsymbol{D}}$
Output: Estimated positions $\mathsf{MDS}_d(\widehat{\boldsymbol{D}})$
Compute $(-1/2)\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L}$, where $\boldsymbol{L} = \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$;
Compute the best rank-$d$ approximation $\boldsymbol{U}_d\boldsymbol{\Sigma}_d\boldsymbol{U}_d^T$ of $(-1/2)\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L}$;
Return $\mathsf{MDS}_d(\widehat{\boldsymbol{D}}) \equiv \boldsymbol{U}_d\boldsymbol{\Sigma}_d^{1/2}$.
```

In this chapter we use what is called the classic metric MDS [CC01], where $f(\cdot)$ is the identity function and the input dissimilarities correspond to the Euclidean distances such that $d_{i,j} = \|x_i - x_j\|$ for some lower dimensional embedding $\{x_i\}$. This algorithm has been frequently used in positioning applications; and from here on whenever we say MDS we refer to the above algorithm. Let $\mathsf{MDS}_d(\bar{\boldsymbol{D}})$ denote the $n \times d$ matrix returned by MDS when applied to the squared distance matrix $\bar{\boldsymbol{D}}$. Then, in formula, given the singular value decomposition (SVD) of a symmetric and positive definite matrix $(-1/2)\boldsymbol{L}\bar{\boldsymbol{D}}\boldsymbol{L}$ as $(-1/2)\boldsymbol{L}\bar{\boldsymbol{D}}\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T$,

$$\mathsf{MDS}_d(\bar{\boldsymbol{D}}) \equiv \boldsymbol{U}_d\boldsymbol{\Sigma}_d^{1/2} \, ,$$

where $\boldsymbol{U}_d$ denotes the $n \times d$ left singular matrix that corresponds to the first $d$ singular values and $\boldsymbol{\Sigma}_d$ denotes the $d \times d$ diagonal matrix with top $d$ singular values. This is also known as the MDSLocalize algorithm in [DJMI+06]. Note that as the columns of $\boldsymbol{U}$ are orthogonal to $\mathbb{1}_n$ by construction, it follow that $\boldsymbol{L} \cdot \mathsf{MDS}_d(\bar{\boldsymbol{D}}) = \mathsf{MDS}_d(\bar{\boldsymbol{D}})$.

It is crucial that we apply the double scaling by $\boldsymbol{L}$ to the squared distance matrix:

$$\bar{\boldsymbol{D}} = a\mathbb{1}_n^T + \mathbb{1}_n a^T - 2\boldsymbol{X}\boldsymbol{X}^T \, .$$

Since by construction, $\boldsymbol{L}$ is orthogonal to $\mathbb{1}_n$, but preserves the $n-1$ dimensional complementary subspace, the double scaling eliminates the first two terms.

Hence, when MDS is applied to $\bar{\boldsymbol{D}}$ without noise, the configuration of sensors are exactly recovered up to a rigid motion:

$$-(1/2)\boldsymbol{L}\bar{\boldsymbol{D}}\boldsymbol{L} = \boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} \, . \tag{2.2}$$

Note that we only obtain the configuration and not the absolute positions, in the sense that $\mathsf{MDS}_d(\bar{\boldsymbol{D}})$ is one version of infinitely many solutions that matches the distance measurements $\bar{\boldsymbol{D}}$. We introduce a formal definition of rigid transformation and related terms below.

Let $\mathsf{O}(d) = \{\boldsymbol{Q} | \boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{Q}^T\boldsymbol{Q} = \mathbb{I}_d\}$ denote the orthogonal group of $d \times d$ matrices. We say $\boldsymbol{Y} \in \mathbb{R}^{n \times d}$ is a rigid transformation of $\boldsymbol{X}$, if there exists a shift vector $s \in \mathbb{R}^d$ and an orthogonal matrix $\boldsymbol{Q} \in \mathsf{O}(d)$ such that $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Q} + \mathbb{1}_n s^T$ . Here $\boldsymbol{Y}$ is a result of rotating $\boldsymbol{X}$ by $\boldsymbol{Q}$ and then adding a shift by $s$. Similarly, when we say two position matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ are equal up to a rigid transformation, we mean that there exists a rotation $\boldsymbol{Q}$ and a shift $s$ such that $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Q} + \mathbb{1}_n s^T$. Also, we say a function $f(\boldsymbol{X})$ is *invariant* under rigid transformation if and only if for all $\boldsymbol{X}$ and $\boldsymbol{Y}$ that are equal up to a rigid transformation we have $f(\boldsymbol{X}) = f(\boldsymbol{Y})$. Under these definitions, it is clear that $\bar{\boldsymbol{D}}$ is invariant under rigid transformation, as for all $(i,j)$, since $\bar{\boldsymbol{D}}_{ij} = \|x_i - x_j\|^2 = \|(x_i\boldsymbol{Q} + s^T) - (x_j\boldsymbol{Q} + s^T)\|^2$, for any $\boldsymbol{Q} \in \mathsf{O}(d)$ and $s \in \mathbb{R}^d$.

Although MDS works perfectly when $\bar{\boldsymbol{D}}$ is available, in practice not all proximity measurements are available because of the limited radio range $R$. This is why, in the first step, we estimated the unavailable entries of $\bar{\boldsymbol{D}}$ by finding the shortest path between disconnected nodes.

### 2.3.2 Distributed Positioning Algorithm: HOP-TERRAIN

Recall that HOP-TERRAIN is a distributed algorithm that aims at finding the global map. In order to fix the global coordinate system in a $d$ dimensional space, we need to know the positions of at least $d+1$ anchors, nodes with known positions. In this section, we assume that we have $m$ anchors. Based on the robust positioning algorithm introduced in [SLR02], the distributed sensor localization algorithm consists of two steps:

**Algorithm 2.3** MDS-MAP [SRZF03].

```
1: Each node $i$ computes the shortest paths $\{\hat{d}_{i,a} : a \in V_a\}$ between itself and the anchors.
2: Each node $i$ derives an estimated position $\hat{x}_i$ by triangulation with a least squares method.
```

**Distributed shortest paths:** Similar to MDS-MAP, the first step is about finding the shortest path. The difference is that in the first step each of the unknown nodes only estimates the distances between itself and the anchors. These approximate distances will be used in the next triangulation step to derive an estimated position. In other words, the shortest path between an unknown node $i$ and an anchor $a$ in the graph $G$ provides an estimate for the Euclidean distance $d_{i,a} = \|x_i - x_a\|$.

We denote by $\hat{d}_{i,a}$ the computed length of the shortest path. When the corresponding graph is defined as in the *connectivity-based model*, the shortest path $\hat{d}_{i,a}$ is equivalent to the minimum number of hops between two nodes, scaled by the radio range $R$.

In order to compute the number of hops in a distributed way, we use a method similar to DV-HOP [NN03]. Each unknown node maintains a table $\{x_a, h_a\}$ that is initially empty, where $x_a \in \mathbb{R}^d$ refers to the position of the anchor $a$ and $h_a$ to the number of hops from the unknown node to the anchor $a$. First, each of the anchors initiates a broadcast containing its known location and a hop count of one. All of the one-hop neighbors surrounding the anchor, on receiving this broadcast, record the anchor's position and a hop count of one, and then broadcast the anchor's known position and a hop count of two. From then on, whenever a node receives a broadcast, it does one of the two things. If the broadcast refers to an anchor that is already in the record and the hop count is larger than or equal to what is recorded, then the node does nothing. Otherwise, if the broadcast refers to an anchor that is new or has a hop count that is smaller, the node updates its table with this new information on its memory and broadcasts the new information after incrementing the hop count by one. When every node has computed the hop count to all the anchors, the number of hops is multiplied by the radio range $R$. Note that to begin triangulation, not all the hop counts to all the anchors are necessary. A node can start triangulation as soon as it has estimated distances to $d + 1$ anchors.

The above step of computing the minimum number of hops is the same distributed algorithm as described in DV-HOP. However, one difference is that instead of multiplying the number of hops by a fixed radio range $R$, in DV-HOP, the number of hops is multiplied by an average hop distance. The average hop distance is computed from the known pairwise distances between anchors and the number of hops between the anchors. Although numerical simulations show that the average hop distance provides a better estimate, the difference between the computed average hop distance and the radio range $R$ becomes negligible as $n$ grows large.

**Triangulation using least squares.** In the second step, each unknown node $i$ uses a set of estimated distances $\{\hat{d}_{i,a} : a \in V_a\}$ together with the known positions of the anchors, to perform a triangulation. The resulting estimated position is denoted by $\hat{x}_i$. For each node, the triangulation consists in solving a single instance of a least squares problem ($\boldsymbol{A}x = b$) and this process is known as Lateration [SRB01, LR03]. The position vector $x_i$ and the anchor positions $\{x_a : a \in \{1, \dots, m\}\}$ satisfy the following:

$$\begin{aligned} \|x_1 - x_i\|^2 &= d_{i,1}^2 \ , \\ &\vdots \\ \|x_m - x_i\|^2 &= d_{i,m}^2 \ . \end{aligned}$$

Geometrically, the above equalities simply say that the point $x_i$ is the intersection point of $m$ circles centred at $x_1, x_2, \dots, x_m$ (see Figure 2.4). This set of equations can be linearised by subtracting each line from the next line. By reordering the terms, we get a series of

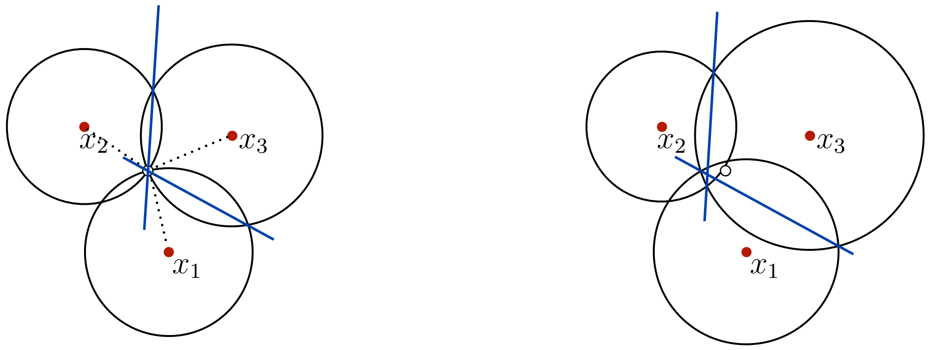


Figure 2.4 – Multilateration with exact distance measurements (left) and with approximate distance measurements (right). Three solid circles denote the anchors (red) and the white circle denotes the unknown nodes. The intersection of the blue lines corresponds to the solution of multilateration.

linear equations in the form $\boldsymbol{A}\,x_i = b_0^{(i)}$, for $\boldsymbol{A} \in \mathbb{R}^{(m-1)\times d}$ and $b \in \mathbb{R}^{m-1}$ defined as

$$\begin{aligned} \boldsymbol{A} &\equiv \begin{bmatrix} 2(x_1 - x_2)^T \\ \vdots \\ 2(x_{m-1} - x_m)^T \end{bmatrix} , \\ b_0^{(i)} &\equiv \begin{bmatrix} \|x_1\|^2 - \|x_2\|^2 + d_{i,2}^2 - d_{i,1}^2 \\ \vdots \\ \|x_{m-1}\|^2 - \|x_m\|^2 + d_{i,m}^2 - d_{i,m-1}^2 \end{bmatrix} . \end{aligned}$$

Note that the matrix $\boldsymbol{A}$ does not depend on the particular unknown node $i$ and all the entries are known accurately to all the nodes. However, the vector $b_0^{(i)}$ is not available at node $i$, because $d_{i,a}$'s are not known. Hence we use an estimation $b^{(i)}$, that is defined from $b_0^{(i)}$ by replacing $d_{i,a}$ by $\hat{d}_{i,a}$ everywhere. Notice that $\hat{d}_{i,a} \geq d_{i,a}$. As a result, the circles centred at $x_1, x_2, \ldots, x_m$ have potentially larger radii. Therefore, the intersection between circles is no longer a single point, but rather a closed area. Then, finding the optimal estimation $\hat{x}_i$ of $x_i$ that minimizes the mean squared error is solved in a closed form using a standard least squares approach:

$$\hat{x}_i = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T b^{(i)} . \tag{2.3}$$

### Computational complexity and network connectivity

**Complexity analysis.** For bounded $d = o(1)$, a single least squares operation has complexity $O(m)$, and applying it $n$ times results in the overall complexity of $O(n\,m)$. No communication between the nodes is necessary for this step. In the MDS-MAP algorithm we require that all-pairs shortest paths be found. This problem has an efficient algorithm whose complexity is $O(n^2 \log n + n|E|)$ [Joh77]. For $R = C(\log n/n)^{1/d}$ with constant $C$, the graph is sparse with $|E| = O(n \log n)$, whence the complexity is $O(n^2 \log n)$. Contrary to MDS-MAP, in HOP-TERRAIN we must only compute the shortest paths between

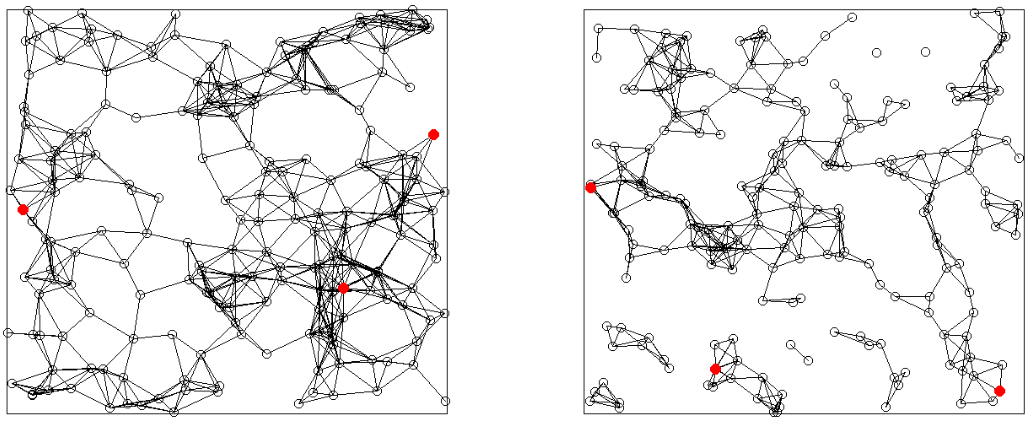


Figure 2.5 – The red vertices indicate the anchors. Under the right scaling of the radio range $R$, the graph stays connected (left figure) whereas otherwise there will be nodes without any means of communication to others (right figure).

the unknown nodes and the anchors. This distributed shortest paths algorithm can be done efficiently with total complexity of $O(n\,m)$.

**Network connectivity.** In general when the graph $G$ is not connected, the localization problem is not well defined, and there are multiple configurations resulting in the same observed proximity measures. For instance if graph $G$ consists of two disconnected components, they can be placed in any way with respect to each other without violating any constraints imposed by $G$.

In this work, we are interested in a scalable system of $n$ unknown nodes for a large value of $n$. As $n$ grows, it is reasonable to assume that the average number of connected neighbors for each node should stay constant. This happens, in our model, if we chose the radio range $R = C/n^{1/d}$. However, in the unit square, assuming sensor positions are drawn uniformly, the random geometric graph is connected, with high probability, if $\pi R^2 > (\log n + c_n)/n$ for $c_n \to \infty$ [GK98]. A similar condition can be derived for generic $d$-dimensions as $C_d R^d > (\log n + c_n)/n$, where $C_d \leq \pi$ is a constant that depends on $d$. Moreover, in case $C_d R^d < (\log n + c_n)/n$, not only the graph is not connected, there will be *isolated nodes* with hight probability. In this case, both MDS-MAP and HOP-TERRAIN algorithms will be in trouble (see Figure 2.5). Hence, we focus in the regime where the average number of connected neighbors is slowly increasing with $n$. Let $R_{\text{critical}}$ be the critical detection range where the resulting graph starts to be connected. Then we are interested in the regime $R = CR_{\text{Critical}}$, for some positive constant $C \geq 1$ such that the graph stays connected with high probability.

## 2.4 Main results

In this section we present our main results regarding the performance of MDS-MAP and HOP-TERRAIN algorithms.

### 2.4.1 MDS-MAP

Our first result establishes an upper bound on the error achieved by MDS-MAP under the *connectivity-based model*. Let $\widehat{\boldsymbol{X}}$ denote an estimate for $\boldsymbol{X}$. Then, we need to define a metric for the distance between the original position matrix $\boldsymbol{X}$ and the estimation $\widehat{\boldsymbol{X}}$, which is invariant under rigid transformation. Recall the matrix $\boldsymbol{L} \equiv \mathbb{I}_n - (1/n)\mathbb{1}_n\mathbb{1}_n^T$ in the MDS algorithm. $\boldsymbol{L}$ is an $n \times n$ symmetric matrix with rank $n-1$, which eliminates the contributions of the translation, in the sense that $\boldsymbol{L}\boldsymbol{X} = \boldsymbol{L}(\boldsymbol{X} + \mathbb{1}s^T)$ for all $s \in R^d$. As we noted earlier, $\boldsymbol{L}$ has the following nice properties:

1. $\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L}$ is invariant under rigid transformation.
2. $\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} = \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}$ implies that $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$ are equal up to a rigid transformation.

As a result, we can define the distance between $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$:

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = \frac{1}{n}\left\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\right\|_F . \tag{2.4}$$

Notice that the factor $(1/n)$ corresponds to the usual normalization by the number of entries in the summation. Furthermore, $d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = 0$ implies that $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$ are equal up to a rigid transformation. With this metric, our main result establishes an upper bound on the resulting error. The proof of this theorem is provided in Section 2.5. We define

$$R_{\text{MDS}} \equiv 32\left(\frac{12\log n}{p_0(n-2)}\right)^{\frac{1}{d}} . \tag{2.5}$$

**Theorem 2.1** (connectivity-based model). *Assume $n$ nodes are distributed uniformly at random in the $[0,1]^d$ hypercube, for a bounded dimension $d \in \{2,3\}$. For a positive radio range $R$ with a minimum detection probability $p_0$, we are given the connectivity information of the nodes according to the range-free model with probabilistic detection. Then, with a probability larger than $1-1/n^4$, the distance between the estimate $\widehat{\boldsymbol{X}}$ produced by* MDS-MAP *and the correct position matrix $\boldsymbol{X}$ is bounded by*

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq \frac{R_{MDS}}{R} + 20R , \tag{2.6}$$

*for $R > (1/p_0)^{1/d}R_{MDS}$, where $d_{\text{inv}}(\cdot)$ is defined in* (2.4) *and $R_{MDS}$ in* (2.5).

The following corollary trivially follows, as for each $(i,j) \in E$, we have $d_{i,j} \leq R$.

**Corollary 2.2** (range-based model). *Under the hypotheses of Theorem 2.1 and in the case of range-based model, with high probability*

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq \frac{R_{MDS}}{R} + 20R .$$

As described in the previous section, we are interested in the regime where $R = C(\log n/n)^{1/d}$ for some constant $C$. Given a small positive constant $\delta$, this implies that MDS-MAP is guaranteed to produce estimated positions that satisfy $d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq \delta$ with a large enough constant $C$ and a large enough $n$. When $p_0$ is fixed and $R = C(\log n/n)^{1/d}$ for some positive parameter $C$, the error bound in (2.6) becomes

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq \frac{C_1}{Cp_0^{1/d}} + C_2C\left(\frac{\log n}{n}\right)^{1/d} ,$$

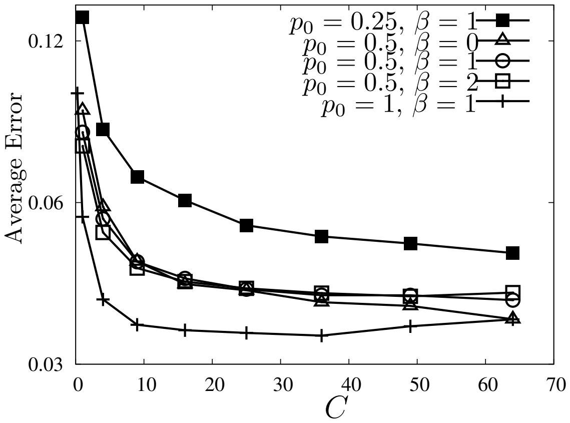


Figure 2.6 – Average error for MDS-MAP with $R = C\sqrt{\log n/n}$ under range-free model.

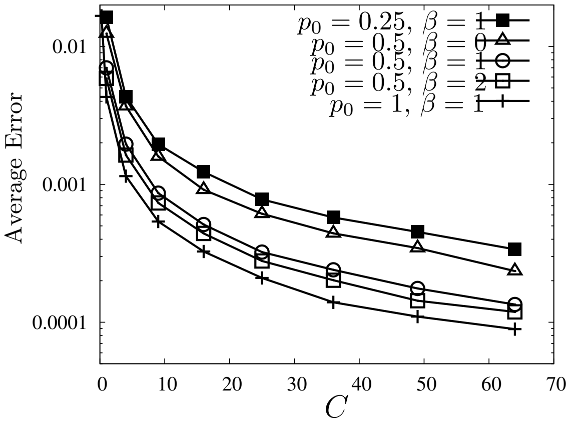


Figure 2.7 – Average error of MDS-MAP under the range-based model.

for some numerical constants $C_1$ and $C_2$. The first term is inversely proportional to $C$ and $p_0^{1/d}$ and is independent of $n$, whereas the second term is linearly dependent on $C$ and vanishes as $n$ grows large. This is illustrated in Figure 2.6, which shows numerical simulations with $n$ sensors randomly distributed in the 2-dimensional unit square, and assuming (2.1). Notice that the resulting error decreases with $p_0$ and is independent of $\beta$.

Even though the upper bounds for both range-free and range-based models have the same form, their behaviours is different as $R$ grows. In the range-free case, up to some point, the performance of MDS-MAP improves as $R$ increases. This is due to the fact that the first and second terms go in opposite directions as a function of $R$. However, in the range-based case (where we measure the pairwise distances exactly if the pair is within a radio range $R$), as $R$ increases, we obtain more measurements with the exact Euclidean distances. As a result, once the radio range increases, the resulting error of MDS-MAP decreases and we do not see the contribution of the second term. This phenomenon is illustrated in Figure 2.7.

Using the above theorem, we can further show that there is a linear transformation $S \in \mathbb{R}^{d\times d}$, such that when applied to the estimations, we get a similar bound in the Frobenius norm of the error in the positions.

**Theorem 2.3.** *Under the hypotheses of Theorem 2.1, with probability larger than* $1-1/n^4$,

$$\min_{S\in\mathbb{R}^{d\times d}} \frac{1}{\sqrt{n}}\|\boldsymbol{L}\boldsymbol{X}-\boldsymbol{L}\widehat{\boldsymbol{X}}\boldsymbol{S}\| \leq \sqrt{6}\left(\frac{R_{MDS}}{R}+20R\right) .$$

Note that although for the sake of simplicity, we focus on $[0,1]^d$ hypercube; our analysis easily generalizes to any bounded convex set and homogeneous Poisson process model with density $\rho = n$. The homogeneous Poisson process model is characterized by the probability that there are exactly $k$ nodes appearing in any region with volume $A$ : $\mathbb{P}(k_A = k) = \frac{(\rho A)^k}{k!}e^{-\rho A}$. Here, $k_A$ is a random variable defined as the number of nodes in a region of volume $A$.

To simplify calculations, we assumed that $d$ is either 2 or 3. However, the analysis easily applies to general $d$ and only the constant in the bound (2.6) would change as long as $d = O(1)$.

### 2.4.2 HOP-TERRAIN

Our second result establishes that HOP-TERRAIN [SLR02] achieves an arbitrarily small error for a radio range $R = C(\log n/n)^{1/d}$ with a large enough constant $C$, when we have only the connectivity information as in the case of the *connectivity-based model*. The same bound holds immediately for the *range-based model*, when we have an approximate measurements for the distances, and the same algorithm can be applied without any modification. To compute better estimates for the actual distances between the unknown nodes and the anchors, the extra information can be readily incorporated into the algorithm. We define

$$R_{\mathrm{HOP}} \equiv 12\left(\frac{12\log n}{p_0(n-2)}\right)^{\frac{1}{d}} . \tag{2.7}$$

**Theorem 2.4.** *Assume $n$ sensors and $m$ anchors are distributed uniformly at random in the $[0,1]^d$ hypercube for $d \in \{2,3\}$. For a given radio range $R > (1/p_0)^{1/d}R_{HOP}$, with a minimum detection probability $p_0$, and the number of anchors $m = \Omega(\log n)$, the following is true with probability at least $1-1/n^4$. For all unknown nodes $i \in V_u$, the Euclidean distance between the estimate $\hat{x}_i$ given by* HOP-TERRAIN *and the correct position $x_i$ is bounded by*

$$\|x_i - \hat{x}_i\| \leq \frac{R_{HOP}}{R} + 24R . \tag{2.8}$$

The proof is provided in Section 2.5. As described in the previous section, we are interested in the regime where $R = C(\log n/n)^{1/d}$ for some constant $C$. Given a small positive constant $\delta$, this implies that HOP-TERRAIN is guaranteed to produce estimated positions that satisfy $\|x_i - \hat{x}_i\| \leq \delta$ for all $i$ with a large enough constant $p_0$ and large enough $n$.

When the number of anchors is bounded and the positions of the anchors are chosen randomly, it is possible that, in the triangulation step, we get an ill-conditioned matrix $\boldsymbol{A}^T\boldsymbol{A}$, resulting in a large estimation error. This happens, for instance, if three anchors fall close to a line. However, as mentioned in the introduction, it is reasonable to assume

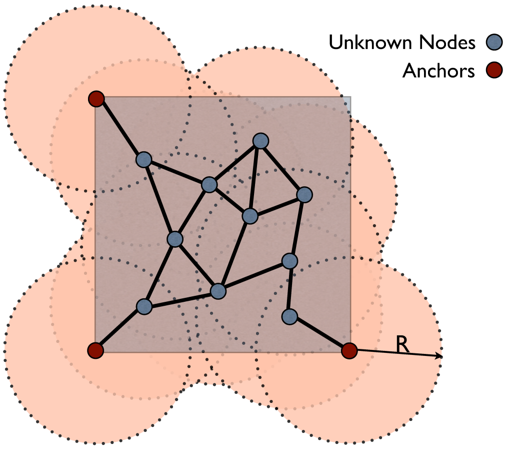


Figure 2.8 – Three anchors in fixed positions ($([0,0],[1,0],[0,1])$) with $d=2$.

that, for the anchors, the system designer has some control over where they are placed. In that case, the next theorem shows that when the positions of anchors are properly chosen, only $d+1$ anchors suffice to get a similar bound. Note that this is the minimum number of anchors necessary for triangulation. For simplicity we assume that one anchor is placed at the origin and $d$ anchors are placed at positions corresponding to $d$-dimensional unit vectors. The position of the $d+1$ anchors are $\{[0,\dots,0],[1,0,\dots,0],\dots\}$. (see figure 2.8)

**Theorem 2.5.** *Under the hypotheses of Theorem 2.4, assume that there are $d+1$ anchors, one of which is placed at the origin, and the position vectors of the $d$ remaining anchors are the $d$-dimensional unit vectors. Then, the following is true with probability at least $1-1/n^4$. For all $i \in V_u$,* Hop-TERRAIN *achieves*

$$\|x_i - \hat{x}_i\| \le 2\frac{R_{HOP}}{R} + 48R\ . \tag{2.9}$$

The proof is provided in Section 2.5. There is nothing particular about the position of the anchors in unit vectors. Any $d+1$ anchors in general position will give similar bounds. The only difference is that the constant term in the definition of $R_{\text{HOP}}$ changes with the anchor positions.

Consider a case where you are designing a sensor network system for environmental monitoring. You want to decide how many sensors to deploy (dropping them from an airplane) and how much radio power each sensors need to be equipped with, in order to be able to determine the positions of the sensors up to an error of $\epsilon$, for any positive $\epsilon$. Then, our main results in (2.8) gives us a guideline for choosing appropriate $n$ and $R$ such that the design goal is met. Precisely, choosing $R=\sqrt{R_{\text{HOP}}/24}$ which minimizes the right-hand side of (2.8) and choosing $n \ge C/(\epsilon^6)$ for some universal constant $C$ achieves the desired design goal. A similar argument is true for a centralized system as well.

There is no known theoretical lower bound for localization under the model considered. One implication of our main results is that we guarantee, with high probability, that for any arbitrarily small constant $\epsilon$, you can achieve error less than $\epsilon$ with radio range $R$ which scales as $R_{\text{critical}}$. You cannot hope to achieve a small error with $R < R_{\text{critical}}$, since the graph starts to be disconnected. However, as we saw in the above example, the dependence on the desired error $\epsilon$ might not be optimal, and there might be better algorithms that can achieve the error with smaller $R$.

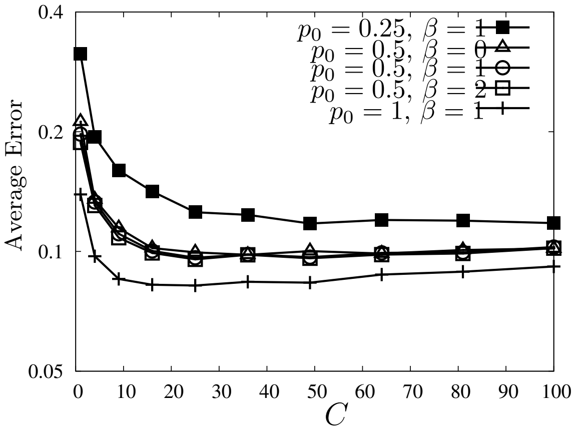


Figure 2.9 – Average error of HOP-TERRAIN for $R = C\sqrt{\log n/n}$ under connectivity-based model.

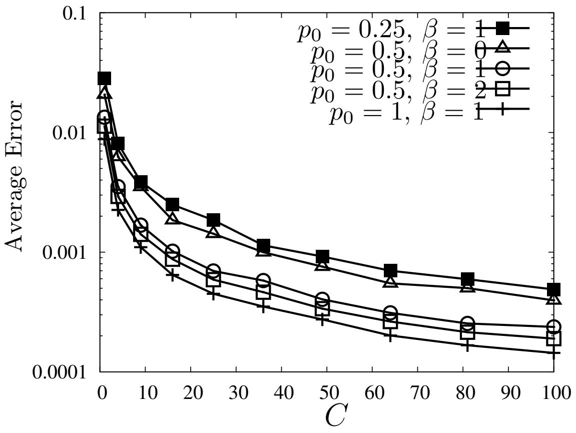


Figure 2.10 – Average error under range-based model.

**Corollary 2.6** (range-based model)**.** *Under the hypotheses of Theorem 2.4 and in the range-based model, with high probability*

$$\|x_i - \hat{x}_i\| \leq \frac{R_{HOP}}{R} + 24R\ .$$

*The similar result holds true when sensors are places deterministically, specifically, under the hypothesis of Theorem 2.5, with high probability,*

$$\|x_i - \hat{x}_i\| \leq 2\frac{R_{HOP}}{R} + 48R\ .$$

As it was the case for MDS-MAP, when $R = C(\log n/n)^{1/d}$ for some positive parameter $C$, the error bound in (2.9) is $\|x_i - \hat{x}_i\| \leq \frac{C_1}{Cp_0^{1/d}} + C_2 C \left(\frac{\log n}{n}\right)^{1/d}$, for some numerical constants $C_1$ and $C_2$. The first term is inversely proportional to $C$ and $p_0^{1/d}$ and is independent of $n$, whereas the second term is linearly dependent in $C$ and vanishes as $n$ grows

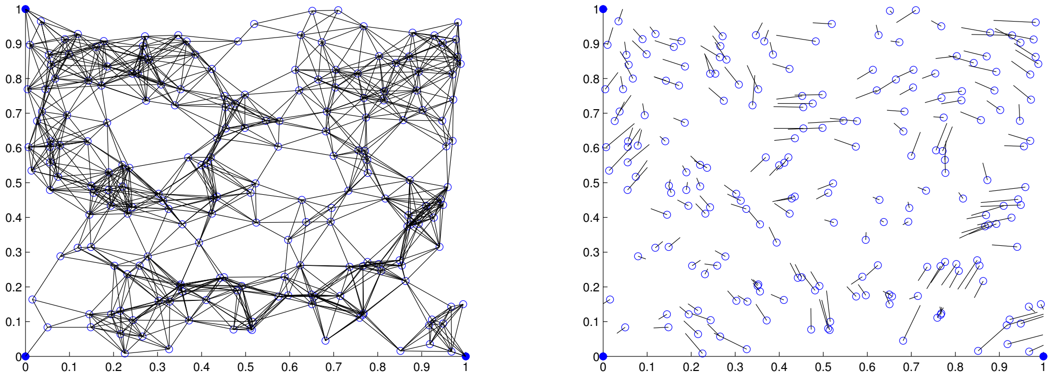


Figure 2.11 – 200 randomly placed nodes with 3 anchors in blue (left). Location estimation using HOP-TERRAIN (right).

large. This is illustrated in Figure 2.9, which shows numerical simulations with $n = 5,000$ sensors in the unit square. To compute the average error we use $\{(1/n)\sum_{i=1}^{n}\|x_i - \hat{x}_i\|^2\}^{1/2}$.

Figure 2.11 shows a network consisting of $n = 200$ nodes placed randomly in the unit circle. The three anchors in fixed positions are displayed by solid blue circles. In this experiment the distance measurements are from the range-based model and the radio range is $\sqrt{0.8\log n/n}$. In the final estimated positions using HOP-TERRAIN, the circles represent the correct positions, and the solid lines represent the differences between the estimates and the correct positions. The average error in this example is 0.075.

## 2.5 Analysis

### 2.5.1 Proof of Theorem 2.1

We start by bounding the distance $d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}})$, as defined in Eq. (2.4). Recall that $\|\boldsymbol{A}\|_F$ denotes the Frobenius norm and $\|\boldsymbol{A}\|_2$ denotes the spectral norm. For a rank $r$ matrix $\boldsymbol{A}$ we have $\|\boldsymbol{A}\|_2 \le \|\boldsymbol{A}\|_F \le \sqrt{r}\|\boldsymbol{A}\|_2$. Since $\boldsymbol{L}(\boldsymbol{X}\boldsymbol{X}^T - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T)\boldsymbol{L}$ has rank at most $2d$, we get

$$\|\boldsymbol{L}(\boldsymbol{X}\boldsymbol{X}^T - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T)\boldsymbol{L}\|_F \le \sqrt{2d}\|\boldsymbol{L}(\boldsymbol{X}\boldsymbol{X}^T - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T)\boldsymbol{L}\|_2 \; .$$

To bound the spectral norm, let $\boldsymbol{M} = -(1/2)\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L}$. Then,

$$\begin{aligned}
\|\boldsymbol{L}(\boldsymbol{X}\boldsymbol{X}^T - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T)\boldsymbol{L}\|_2 &\le \|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{M}\|_2 + \|\boldsymbol{M} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\|_2 \\
&\le (1/2)\|\boldsymbol{L}(-\bar{\boldsymbol{D}} + \widehat{\boldsymbol{D}})\boldsymbol{L}\|_2 \\
&\quad +(1/2)\|\boldsymbol{L}(-\widehat{\boldsymbol{D}} + \bar{\boldsymbol{D}})\boldsymbol{L}\|_2 \\
&\le \|\widehat{\boldsymbol{D}} - \bar{\boldsymbol{D}}\|_2 \; ,
\end{aligned} \tag{2.10}$$

where in the first inequality we used the triangular inequality and the fact that $\widehat{\boldsymbol{X}} = \boldsymbol{L}\widehat{\boldsymbol{X}}$. In the second inequality we used (2.2) and the fact that $\|\boldsymbol{M} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\|_2 = \min_{\boldsymbol{A}:\text{rank}(\boldsymbol{A})\le d}\|\boldsymbol{M} - \boldsymbol{A}\|_2$, which follows from the definition of $\widehat{\boldsymbol{X}}$. From the definition of $\widehat{\boldsymbol{X}} = \mathsf{MDS}_d(\widehat{\boldsymbol{D}})$, we know that $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T$ is the best rank-$d$ approximation to $\boldsymbol{M}$. Hence, $\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T$ minimizes

$\|\boldsymbol{M} - \boldsymbol{A}\|_2$ for any rank-$d$ matrix $\boldsymbol{A}$. Since the rank of $-(1/2)\boldsymbol{L}\bar{\boldsymbol{D}}\boldsymbol{L}$ is $d$, this implies $\|\boldsymbol{M} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\|_2 \leq \|\boldsymbol{M} + (1/2)\boldsymbol{L}\bar{\boldsymbol{D}}\boldsymbol{L}\|_2$. The inequality (2.10) follows trivially from the observation that $\|\boldsymbol{L}\|_2 = 1$.

Next, to bound $\|\widehat{\boldsymbol{D}} - \bar{\boldsymbol{D}}\|_2$, we use the following key result. The main idea is that, the number of hops scaled by the radio range $R$ provides a good estimate of the correct distance. Let $\widetilde{R} \equiv 2(12\log n/(p_0(n-2)))^{1/d}$.

**Lemma 2.7.** (Bound on the distance estimation) *Under the hypotheses of Theorem 2.1, with probability larger than $1 - 1/n^4$, for any pair of nodes $i \in V$ and $j \in V$, the number of hops between nodes $i$ and $j$ is bounded by*

$$h_{i,j} \;\leq\; \Big(1 + \frac{\widetilde{R}}{R}\Big)\frac{d_{i,j}}{R} + 2\;,$$

*for $R > \max\{7\widetilde{R}, (1/p_0)^{1/d}\widetilde{R}\}$.*

The proof of this lemma is provided in Section 2.5.7. The distance estimate from the first step of MDS-MAP is $\hat{d}_{i,j} = Rh_{i,j}$. The following corollary gives a bound on the error.

**Corollary 2.8.** *Under the hypotheses of Lemma 2.7,*

$$\hat{d}^2_{i,j} - d^2_{i,j} \leq \frac{30\widetilde{R}}{14R}d^2_{i,j} + 8R\;.$$

*Proof.* From Lemma 2.7, we know that

$$(R\,h_{i,j})^2 - d^2_{i,j} \leq \frac{2\widetilde{R}}{R}\Big(1 + \frac{\widetilde{R}}{2R}\Big)d^2_{i,j} + 2R\Big(1 + \frac{\widetilde{R}}{R}\Big)d_{i,j} + 4R^2\;.$$

The corollary follows from the assumption that $7\widetilde{R} < R \leq 1$ and $d \leq 3$. □

Define an error matrix $\boldsymbol{Z} = \widehat{\boldsymbol{D}} - \bar{\boldsymbol{D}}$. Then by Corollary 2.8, $\boldsymbol{Z}$ is element-wise bounded by $0 \leq \boldsymbol{Z}_{ij} \leq (30\widetilde{R}/(14R))\bar{\boldsymbol{D}}_{ij} + 8R$. We can bound the spectral norm of $\boldsymbol{Z}$ as follows. Let $u$ and $v$ be the left and right singular vectors of the non-negative matrix $\boldsymbol{Z}$, respectively. Then by Perron-Frobenius theorem, $u$ and $v$ are also non-negative. It follows that

$$\begin{aligned}\|\widehat{\boldsymbol{D}} - \bar{\boldsymbol{D}}\|_2 &= u^T\boldsymbol{Z}v \\ &\leq (30\widetilde{R}/(14R))u^T\bar{\boldsymbol{D}}v + (\mathbb{1}^T u)(\mathbb{1}^T v)8R \\ &\leq (30\widetilde{R}/(14R))\|\bar{\boldsymbol{D}}\|_2 + 8Rn \\ &\leq (30\widetilde{R}/(14R))dn + 8Rn\;. \end{aligned} \tag{2.11}$$

The first inequality follows from the element-wise bound on $\boldsymbol{Z}$ and the non-negativity of $u$ and $v$, and the second inequality follows form the definition of the spectral norm and the Cauchy-Schwarz inequality. In the last inequality, we used $\|\bar{\boldsymbol{D}}\|_2 \leq dn$, which follows from the fact that $\bar{\boldsymbol{D}}$ is non-negative and element-wise bounded by $d$. Typically we are interested in the regime where $R = o(1)$, and by assumption we know that $R \geq \widetilde{R}$ and $d \leq 3$. Therefore, the first term in (2.11) dominates the error. Substituting this bound on $\|\widehat{\boldsymbol{D}} - \bar{\boldsymbol{D}}\|_2$ in (2.10) proves the theorem.

### 2.5.2 Proof of Theorem 2.3

Using SVD we can write $\boldsymbol{LX}$ as $\boldsymbol{U}_{n\times d}\boldsymbol{\Sigma}_{d\times d}\boldsymbol{V}^T_{d\times d}$ where $\boldsymbol{U}^T\boldsymbol{U} = \mathbb{I}_{d\times d}$, $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{V}\boldsymbol{V}^T = \mathbb{I}_{d\times d}$ and $\boldsymbol{\Sigma}$ is a diagonal matrix. We also denote the inner product of matrices by $\langle \boldsymbol{A}, \boldsymbol{B}\rangle \equiv \sum_{i,j} \boldsymbol{A}_{i,j}\boldsymbol{B}_{i,j}$. It is easy to show that $\langle \boldsymbol{A}, \boldsymbol{B}\rangle = \mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{B}) \leq \|\boldsymbol{A}\|_F\|\boldsymbol{B}\|_F$. Now, for $\boldsymbol{S} = \widehat{\boldsymbol{X}}^T\boldsymbol{LU}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^T$, then

$$\|\boldsymbol{LX} - \boldsymbol{L}\widehat{\boldsymbol{X}}\boldsymbol{S}\|_F = \sup_{\boldsymbol{B}\in\mathbb{R}^{n\times d}, \|\boldsymbol{B}\|_F\leq 1} \langle \boldsymbol{B}, \boldsymbol{LX} - \boldsymbol{L}\widehat{\boldsymbol{X}}\boldsymbol{S}\rangle \tag{2.12}$$

The above equation can be further written as

$$\begin{aligned}
(2.12) &= \sup_{\boldsymbol{B}\in\mathbb{R}^{n\times d}, \|\boldsymbol{B}\|_F\leq 1} \langle \boldsymbol{B}, (\boldsymbol{LXV}\boldsymbol{\Sigma}\boldsymbol{U}^T - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L})\boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^T\rangle \\
&= \sup_{\boldsymbol{B}\in\mathbb{R}^{n\times d}, \|\boldsymbol{B}\|_F\leq 1} \langle \boldsymbol{BV}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T, \boldsymbol{LXX}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\rangle \\
&= \sup_{\boldsymbol{B}\in\mathbb{R}^{n\times d}, \|\boldsymbol{B}\|_F\leq 1} \|\boldsymbol{BV}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\|_F\|\boldsymbol{LXX}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\|_F.
\end{aligned}$$

Using the fact $\|\boldsymbol{A}\|_F = \mathrm{Tr}(\boldsymbol{A}^T\boldsymbol{A})$ and the cyclic property of the trace, i.e., $\mathrm{Tr}(\boldsymbol{ABC}) = \mathrm{Tr}(\boldsymbol{BCA})$, we obtain

$$\|\boldsymbol{BV}\boldsymbol{\Sigma}^{-1}\boldsymbol{U}^T\|_F^2 = \mathrm{Tr}(\boldsymbol{BV}\boldsymbol{\Sigma}^{-2}\boldsymbol{V}^T\boldsymbol{B}^T) \leq \sigma_{\min}^{-2}\|\boldsymbol{B}\|_F^2,$$

where $\sigma_{\min}$ is the smallest singular value of $\boldsymbol{LX}$. It remains to show that $\sigma_{\min} \geq \sqrt{n/6}$ holds with high probability when nodes are placed uniformly at random. To this end we need to consider two facts. First, the singular values (and in particular the smallest singular value) are Lipschitz functions of the entries. Second, we have $\mathbb{E}(\boldsymbol{LXX}^T\boldsymbol{L}) = (n/12)\mathbb{I}_{d\times d}$. By using concentration of measure for Lipschitz functions on bounded independent random variables, the result follows.

### 2.5.3 Proof of Theorem 2.4

In this section we provide the proofs of the theorems 2.4, and proofs of the technical lemmas are provided in the following sections . From Eq. (2.3), we get

$$\begin{aligned}
\|x_i - \hat{x}_i\| &= \|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T b_0^{(i)} - (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T b^{(i)}\| \\
&\leq \|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\|_2\|b_0^{(i)} - b^{(i)}\| \,,
\end{aligned} \tag{2.13}$$

First, to bound $\|b_0^{(i)} - b^{(i)}\|$, we use Corollary 2.8. Since $d_{i,j}^2 \leq d$ for all $i$ and $j$, we have

$$\begin{aligned}
\|b_0^{(i)} - b^{(i)}\| &= \Big(\sum_{k=1}^{m-1}\big(d_{i,k+1}^2 - d_{i,k}^2 - \hat{d}_{i,k+1}^2 + \hat{d}_{i,k}^2\big)^2\Big)^{1/2} \\
&\leq 2\sqrt{m-1}\left(\frac{30\widetilde{R}}{14R}d + 8R\right) \,,
\end{aligned} \tag{2.14}$$

Next, to bound $\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\|_2$, we use the following lemma.

**Lemma 2.9.** *Under the hypothesis of Theorem 2.4, the following is true. Assuming random anchor model in which $m = \Omega(\log n)$ anchors are chosen uniformly at random among $n$ sensors. Then we have*

$$\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\|_2 \leq \sqrt{\frac{3}{m-1}},$$

*with high probability.*

By assumption we know that $R \geq \widetilde{R}$ and $d \leq 3$. By combining (2.13), (2.14) and Lemma 2.9 proves Theorems 2.4.

### 2.5.4 Proof of Theorem 2.5

Similarly to the proof of Theorem 2.4, for an unknown node $i$, and the estimate $\hat{x}_i$ we have

$$\|x_i - \hat{x}_i\| \quad \leq \quad \|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\|_2\|b_0^{(i)} - b^{(i)}\| \ ,$$

We have already bounded the expression $\|b_0^{(i)} - b^{(i)}\|$ in (2.14). To bound $\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\|_2$, we use the following lemma.

**Lemma 2.10.** *Under the hypothesis of Theorem 2.5, the following are true. We assume a deterministic anchor model, where $m = d+1$ anchors are placed on the positions $x_1 = [1, 0, \ldots, 0], \ldots, x_m = [0, 0, \ldots, 0]$. Then, $\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\|_2 \leq \frac{d}{2}$, with high probability.*

This finishes the proof of Theorems 2.5.

### 2.5.5 Proof of Lemmas 2.9 (Random Model)

In order to upper bound $\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}\|_2$ we need to lower bound the smallest singular value of $\boldsymbol{A}$. Let the symmetric matrix $\boldsymbol{B}$ be defined as $\boldsymbol{A}^T\boldsymbol{A}$. The diagonal entries of $\boldsymbol{B}$ are

$$b_{i,i} = 4\sum_{k=1}^{m-1}(x_{k,i} - x_{k+1,i})^2, \tag{2.15}$$

for $1 \leq i \leq d$ and the off-diagonal entries are

$$b_{i,j} = 4\sum_{k=1}^{m-1}(x_{k,i} - x_{k+1,i})(x_{k,j} - x_{k+1,j}), \tag{2.16}$$

for $1 \leq i \neq j \leq d$ where $x_{k,i}$ is the $i$-th element of vector $x_k$. In the following lemmas, we show that with high probability, as $m$ increases, the diagonal entries of $\boldsymbol{B}$ will all be of the order of $m$, i.e., $b_{i,i} = \Theta(m)$, and the off-diagonal entries will be bounded from above by $m^{\frac{1}{2}+\epsilon}$, i.e., $b_{i,j} = o(m)$.

**Lemma 2.11.** *For any $\epsilon > 0$ the diagonal entries of $\boldsymbol{B}$ are bounded as follows.*

$$\mathbb{P}\left(|b_{i,i} - 2(m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \leq 4e^{-m^{2\epsilon}}.$$

We use Hoeffding's inequality. To this end, we need to divide the sum in (2.15) into sums of even and odd terms as follows:

$$b_{i,i} = b_e^i + b_o^i,$$

where

$$\begin{aligned} b_e^i &= 4 \sum_{k \in \text{even}} (x_{k,i} - x_{k+1,i})^2, & (2.17) \\ b_o^i &= 4 \sum_{k \in \text{odd}} (x_{k,i} - x_{k+1,i})^2. & (2.18) \end{aligned}$$

This separation ensures that the random variables in summations (2.17) and (2.18) are independent. Let the random variable $z_k^i$ denote the term $4(x_{k,i} - x_{k+1,i})^2$ in (2.17). Since $z_k^i \in [0,4]$ and all the terms in $b_e^i$ are independent of each other, we can use Hoeffding's inequality to upper bound the probability of the deviation of $b_e^i$ from its expected value:

$$\mathbb{P}\left(|b_e^i - (m-1)/3| > 2m^{\frac{1}{2}+\epsilon}\right) \leq 2e^{-m^{2\epsilon}}, \tag{2.19}$$

for any fixed $\epsilon > 0$. The same bound holds for $b_o$. Namely,

$$\mathbb{P}\left(|b_o^i - (m-1)/3| > 2m^{\frac{1}{2}+\epsilon}\right) \leq 2e^{-m^{2\epsilon}}. \tag{2.20}$$

Hence,

$$\begin{aligned} &\mathbb{P}\left(|b_{i,i} - 2(m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right) \\ &\qquad \leq \mathbb{P}\left(|b_e - (m-1)/3| + |b_o - (m-1)/3| > 4m^{\frac{1}{2}+\epsilon}\right), \end{aligned}$$

where we used triangular inequality. Applying union bound, this is upper bounded by $4e^{-m^{2\epsilon}}$.

**Lemma 2.12.** *For any $\epsilon > 0$ the off-diagonal entries of $\boldsymbol{B}$ are bounded as follows.*

$$\mathbb{P}\left(|b_{i,j}| > 16m^{\frac{1}{2}+\epsilon}\right) \leq 4e^{-m^{2\epsilon}}.$$

Using the Gershgorin circle theorem [HJ85] we can find a lower bound on the minimum eigenvalue of $\boldsymbol{B}$.

$$\lambda_{\min}(\boldsymbol{B}) \geq \min_i (b_{i,i} - R_i), \tag{2.21}$$

where $R_i = \sum_{j \neq i} |b_{i,j}|$. Now, let $\mathbb{B}_{ii}$ denote the event that $\{b_{i,i} < 2(m-1)/3 - 4m^{\frac{1}{2}+\epsilon}\}$ and $\mathbb{B}_{ij}$ (for $i \neq j$) denote the event that $\{b_{i,j} > 16m^{\frac{1}{2}+\epsilon}\}$. Since the matrix $\boldsymbol{B}$ is symmetric, we have only $d(d+1)/2$ degrees of freedom. Lemma 2.11 and 2.12 provide us with a bound on the probability of each event. Therefore, by using the union bound we get

$$\begin{aligned} \mathbb{P}\left(\bigcup_{i \leq j} \overline{\mathbb{B}_{ij}}\right) &\leq 1 - \sum_{i \leq j} \mathbb{P}(\mathbb{B}_{ij}) \\ &= 1 - 3d^2 e^{-m^{2\epsilon}}. \end{aligned}$$

Therefore with probability at least $1 - 3d^2 e^{-m^{2\epsilon}}$ we have

$$b_{i,i} - R_i \geq \frac{2(m-1)}{3} - 16d \cdot m^{\frac{1}{2}+\epsilon}, \tag{2.22}$$

for all $1 \leq i \leq d$. As $m$ grows, the RHS of (2.22) can be lower bounded by $(m-1)/3$. By combining (2.21) and (2.22) we get

$$\mathbb{P}\left(\lambda_{\min}(\boldsymbol{B}) \geq \frac{(m-1)}{3}\right) \geq 1 - 3d^2 e^{-m^{2\epsilon}}. \tag{2.23}$$

As a result, from (2.25) and (2.23) we have

$$\mathbb{P}\left(\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}\|_2 \leq \sqrt{\frac{3}{m-1}}\right) \geq 1 - 3d^2 e^{-m^{2\epsilon}}, \tag{2.24}$$

which finishes the proof.

### 2.5.6 Proof of Lemmas 2.10 (Deterministic Model)

By using the singular value decomposition of a tall $m-1 \times d$ matrix $\boldsymbol{A}$, we know that it can be written as $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ where $\boldsymbol{U}$ is an orthogonal matrix, $\boldsymbol{V}$ is a unitary matrix and $\boldsymbol{\Sigma}$ is a diagonal matrix. Then, $(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}^{-1}\boldsymbol{V}^T$. Hence,

$$\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}\|_2 = \frac{1}{\sigma_{min}(\boldsymbol{A})}, \tag{2.25}$$

where $\sigma_{min}(\boldsymbol{A})$ is the smallest singular value of $\boldsymbol{A}$. This means that in order to upper bound $\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}\|_2$ we need to lower bound the smallest singular value of $\boldsymbol{A}$.

By putting the sensors in the mentioned positions the $d \times d$ matrix $\boldsymbol{A}$ will be Toeplitz and have the following form.

$$\boldsymbol{A} = 2\begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

We can easily find the inverse of matrix $\boldsymbol{A}$.

$$\boldsymbol{A}^{-1} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \end{bmatrix}.$$

Note that the maximum singular value of $\boldsymbol{A}^{-1}$ and the minimum singular value of $\boldsymbol{A}$ are related as $\sigma_{min}(\boldsymbol{A}) = (\sigma_{max}(\boldsymbol{A}^{-1}))^{-1}$. To find the maximum singular value of $\boldsymbol{A}^{-1}$, we need to calculate the maximum eigenvalue of $\boldsymbol{A}^{-1}\left(\boldsymbol{A}^{-1}\right)^T$. Using the Gershgorin circle theorem [HJ85] we can find an upper bound: $\lambda_{\max}\left(\boldsymbol{A}^{-1}\left(\boldsymbol{A}^{-1}\right)^T\right) \leq \frac{d^2}{4}$. Combining this with (2.25), we get $\|(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}\|_2 \leq \frac{d}{2}$.

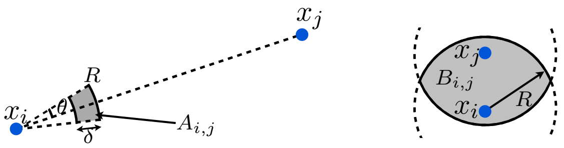


Figure 2.12 – Two dimensional illustration of a bin $A_{i,j}$ when two nodes are far apart (left), and bin $B_{i,j}$ when two nodes are less than $R$ apart (right).

### 2.5.7 Proof of the Bound on the Number of Hops

We start by applying a bin-covering technique in a similar way as in [MP05, OKM10, KO10], but for a more general model where there can be a detection failure. In this section, for simplicity, we assume that the nodes are placed in a 3-dimensional space, but analogous argument is true for $d = 2$ as well.

For each ordered pair of nodes $(i, j)$ we are going to define a corresponding $d$-dimensional space, which we call a 'bin' (see Figure 2.12). When two nodes are far apart, i.e. $d_{i,j} > R$, we define a 'bin' as

$$A_{i,j} = \left\{ x \in [0,1]^3 \;\middle|\; R - \delta \le d(x, x_i) \le R, \measuredangle(x_j - x_i, x - x_i) \le \theta \right\} ,$$

where $\delta$ and $\theta$ are positive parameters which define the size of the bin, and we specify their values later. Here, $\measuredangle(\cdot,\cdot)$ is the angle between two vectors. We say a bin $A_{i,j}$ is occupied if there is a node inside the bin that is detected by node $i$ (i.e., connected to node $i$ in the graph $G$). We want $\delta$ and $\theta$ large enough that all bins are occupied (with high probability), but we want them to be small enough that the furthest distance between two nodes in the same bin is small: this distance bounds the maximum error we make in each bin.

Next, when $(i, j)$ is close by, $d_{i,j} \le R$, we define a bin:

$$B_{i,j} = \left\{ x \in [0,1]^3 \;\middle|\; d(x, x_i) \le R, d(x, x_j) \le R \right\} .$$

We say a bin $B_{i,j}$ is occupied if there is a node inside the bin that is simultaneously detected by nodes $i$ and $j$. When $n$ nodes are deployed in $[0,1]^d$ uniformly at random, we want to ensure that, with high probability, all bins are occupied for appropriate choices of $\delta$ and $\theta$ as functions of $d$, $R$, and $p_0$, where $p_0$ is the minimum probability of detection. First when $d_{i,j} > R$,

$$\mathbb{P}\big(A_{i,j} \text{ occupied}\big) = 1 - \prod_{l \ne i,j} \left(1 - \mathbb{P}(\text{node } l \text{ occupies } A_{i,j})\right) \tag{2.26}$$

The above probability can be lower bounded as follows:

$$\begin{aligned} (2.26) &\ge 1 - \left(1 - \frac{1}{4} \int_0^\theta \int_{R-\delta}^R 2\pi r^2 \sin(\phi) p_0 \mathrm{d}r \mathrm{d}\phi \right)^{n-2} \\ &= 1 - \left(1 - \frac{1}{2} \pi p_0 (1 - \cos(\theta)) \frac{1}{3} (R^3 - (R - \delta)^3) \right)^{n-2} . \end{aligned}$$

The factor $1/4$ in the first line comes from the fact that, in the worst case the intersection of $A_{i,j}$ and the 3-dimensional cube $[0,1]^3$ is at most a quarter of the bin $A_{i,j}$, which happens for instance if $x_i = [0,0,0]$ and $x_j = [1,0,0]$. We choose $\theta$ such that $1-\cos(\theta) = \pi/6$. Then using the facts that $1-z \leq \exp(-z)$ and $1-z^3 \geq (1-z)^3$ for $z \in [0,1]$, we have

$$\mathbb{P}\big(A_{i,j} \text{ is occupied}\big) \geq 1 - \exp\left(-p_0\delta^3(n-2)\right) ,$$

which is larger than $1-1/n^6$ if we set $\delta = (12\log n/(p_0(n-2)))^{1/3}$.

Next we consider the case when nodes $i$ and $j$ are at most $R$ apart. Notice that nodes $i$ and $j$ may not be directly connected in the graph. The probability that they are not directly connected is at least $p_0$, which does not vanish even for large $n$. But we can show that nodes $i$ and $j$ are at most 2 hops apart with overwhelming probability. Then,

$$\begin{aligned}\mathbb{P}\big(B_{i,j} \text{ is occupied}\big) &= 1 - \prod_{l \neq i,j}(1-\mathbb{P}(\text{node } l \text{ is detected by } i \text{ and } j)) \\ &\geq 1-(1-V(B_{i,j})p_0^2)^{n-2} \\ &\geq 1-\exp\big\{-V(B_{i,j})p_0^2(n-2)\big\}, \end{aligned} \tag{2.27}$$

where $V(B_{i,j})$ is the volume of $B_{i,j}$, and we used the fact that the probability of detection is lower bounded by $p_0$. $V(B_{i,j})$ is the smallest when nodes $i$ and $j$ are distance $R$ apart and lie on one of the edges of the cube $[0,1]^3$. In a 3-dimensional space, $V(B_{i,j}) \geq (1/4)(5/12)\pi R^3 \geq (1/4)R^3$. Substituting these bounds in (2.27), we get

$$\mathbb{P}\big(B_{i,j} \text{ is occupied}\big) \geq 1 - \exp\big\{-(1/4)p_0^2R^3(n-2)\big\} , \tag{2.28}$$

which is larger than $1-1/n^6$ for $R \geq \big((24\log n)/((n-2)p_0^2)\big)^{1/3}$.

For each ordered pair $(i,j)$, we are interested in the bin $A_{i,j}$ if $d_{i,j} > R$ and $B_{i,j}$ if $d_{i,j} \leq R$. Using the bounds in (2.27) and (2.28) and applying union bound over $n(n-1)$ ordered pairs, all bins are occupied with a probability larger than $1-1/n^4$.

Now assuming all bins are occupied, we first show that the number of hops between two nodes $i$ and $j$ is bounded by a function $F(d_{i,j})$ that only depends on the distance between the two nodes. The function $F : \mathbb{R}^+ \to \mathbb{R}^+$ is defined as

$$F(z) = \begin{cases} 2 & \text{if } z \leq R , \\ k+2 & \text{if } z \in \mathcal{L}_k \;\; \text{for } k \in \{1,2,\ldots\} , \end{cases}$$

where $\mathcal{L}_k$ denotes the interval $(k(R-\sqrt{3}\delta)+\sqrt{3}\delta, k(R-\sqrt{3}\delta)+R]$. Our strategy is to use induction to show that for all pairs,

$$h_{i,j} \leq F(d_{i,j}) . \tag{2.29}$$

First, assume nodes $i$ and $j$ are at most $R$ apart. Then, by the assumption that $B_{i,j}$ is occupied there exists a node connected to both $i$ and $j$. Then the number of hops $h_{i,j}$ is at most two. Next, assume that (2.29) is true for all $(l,m)$ with

$$d_{l,m} \leq \sqrt{3}\delta + k(R-\sqrt{3}\delta).$$

For two nodes $i$ and $j$ at distance $d_{i,j} \in \mathcal{L}_k$, consider a line segment $\ell_{i,j}$ in the 3-dimensional space with one end at $x_i$ and the other at $x_j$. Let $y \in \mathbb{R}^3$ be the point in the line segment $\ell_{i,j}$ that is at distance $R$ from $x_i$. We want to show that there exists a node that is close to $y$ and is connected to node $i$. By definition, $y$ is inside the bin $A_{i,j}$. We know that the

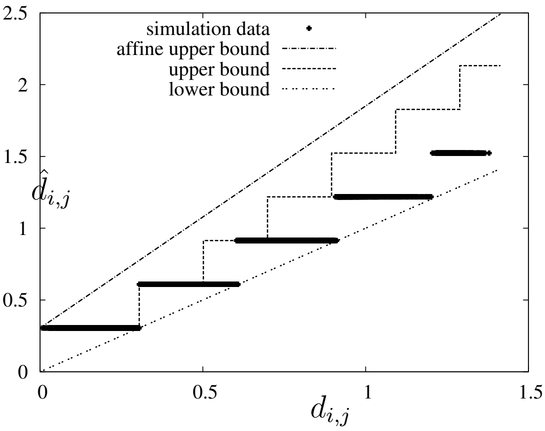


Figure 2.13 – Comparison of upper and lower bound of shortest paths $\{\hat{d}_{i,j}\}$ with respect to the correct distance $\{d_{i,j}\}$ under connectivity-based model.

bin $A_{i,j}$ is occupied by at least one node that is connected to node $i$. Let us denote one of these nodes by $l$. Then $d(y, x_l) \leq \sqrt{3}\delta$ because

$$\sup_{z \in A_{i,j}} d(z, y) = \sqrt{\delta^2 + 2R(R-\delta)(1-\cos(\theta))} \leq \sqrt{3}\delta.$$

We use the following triangular inequality: $h_{i,j} \leq h_{i,l} + h_{l,j}$. Since $l$ is connected to $i$ we have $h_{i,l} = 1$. By triangular inequality, we also have $d_{l,j} \leq d(y, x_j) + d(y, x_l)$. It follows from $d(y, x_j) = d_{i,j} - R$ and $d(y, x_l) \leq \sqrt{3}\delta$ that

$$d_{l,j} \leq d_{i,j} - R + \sqrt{3}\delta.$$

Recall that we assumed $d_{i,j} \leq R + k(R - \sqrt{3}\delta)$. Since we assumed that (2.29) holds for $d_{l,j} \leq \sqrt{3}\delta + k(R - \sqrt{3}\delta)$, we have $h_{i,j} \leq k + 2$, for all nodes $i$ and $j$ such that $d_{i,j} \leq R + k(R - \sqrt{3}\delta)$. By induction, this proves that the bound in (2.29) holds for all pairs $(i, j)$.

We can upper bound $F(z)$ with a simple affine function:

$$\begin{aligned} F(z) &\leq 2 + \frac{1}{R - \sqrt{3}\delta} z \\ &\leq 2 + \left(1 + \frac{2\delta}{R}\right)\frac{z}{R}, \end{aligned}$$

where the last inequality is true for $R \geq 2\sqrt{3}\delta/(2 - \sqrt{3})$. Together with (2.29) this finishes the proof of the lemma.

Figure 2.13 illustrates the comparison of the upper bounds $F(d_{i,j})$ and $F_a(d_{i,j})$, and the trivial lower bound $\hat{d}_{i,j} \geq d_{i,j}$ in a simulation with $d = 2$, $n = 6000$ and $R = \sqrt{64 \log n/n}$. The simulation confirms that the shortest paths lie between the analytical upper and lower bounds. Although the gap between the upper and lower bound is seemingly large, in the regime where $R = C\sqrt{\log n/n}$ with a constant $C$, the vertical gap $R$ vanishes as $n$ grows large and the slope of the affine upper bound can be made arbitrarily small by taking large enough $C$.

“−·−− · ··· ·−·−·− − ···· · −·−− ·− ·−· ·
··−· ·− −·− · ······ − ···· · ·−· · ·− ·−·· −−− −· · ···
− ·−· ·· · −·· − −−− −·− ·· ·−·· ·−·· −− · ······”

·−−− · −· −· ·· ··−· · ·−· ·−−− · ·−· −−· · −· ···

# Calibration Through Matrix Completion

# 3

In most applications that involve sensing, finding the correct positions of the sensors is of crucial importance for obtaining reliable results. This is particularly true in the case of inverse problems that can be very sensitive to incorrect sensor placement. This requirement can be satisfied in two ways: We can put the effort in the construction of the instruments and try to place the sensors exactly in the desired positions, or we can use a method to find the exact positions after the construction of the device. In this chapter [1], we will consider this method for obtaining the sensor positions, called *calibration*. Note that even in the former case, due to the precision of the construction instruments, a calibration is needed afterwards to determine the exact sensor positions. Although in rare cases a single calibration might be enough throughout the lifetime of the measurement system, it is very useful to have a calibration procedure that can be repeated easily and with low cost.

In this chapter, our focus is on the calibration problem in circular sensing devices, in particular, those manufactured and deployed in [DLP$^+$07, JSV09]. These devices consist of a circular ring that surrounds an object and scans horizontal planes. Ultrasound sensors are placed on the interior boundary of the ring and act both as transmitters and receivers.

The calibration problem we address here is the following: In circular tomography devices, sensors are not placed exactly on a perfect circle. This uncertainty in the positions of the sensors acts as a source of error in the reconstruction algorithms used to obtain the characteristics of the enclosed object. We develop a simple method for calibrating the system with correct sensor positions at low cost and without using any extra calibrating instruments.

In order to find the correct sensor positions, we incorporate the time-of-flight (ToF) of ultrasound signals between pairs of sensors, which is the time taken by an ultrasound wavefront to travel from a transmitter to a receiver. If we have all the ToF measurements between all pairs of sensors when the enclosed medium is homogeneous, then we can construct a ToF matrix where each entry corresponds to the ToF between each pair of sensors. We can infer the positions of the sensors by using this ToF matrix.

---

[1] This chapter is the result of a collaboration with R. Parhizkar, S. Oh and M. Vetterli.

To obtain reliable ToF entries appropriate for our purpose, we assume that no object is placed inside the ring during the calibration phase nor prior to actual measurements. There are a number of challenges we encounter:

- the ToF matrices obtained in a practical setup have missing entries,
- the measured entries of the ToF matrices are corrupted by noise,
- there is an unknown time delay added to the measurements.

If we had the complete and noiseless ToF matrix without time delay, the task of finding the exact positions would be very simple. As we saw in the previous chapter, this problem is addressed in the literature as multi-dimensional scaling (MDS) [DJMI$^+$06]. Unfortunately, the ToF matrix in practical setups is never complete and many of the time-of-flight values are missing. The missing entries can be divided into two categories: *structured missing entries* caused by the inability of sensors to compute the mutual time of flights with their close-by neighbors; and *random missing entries* due to the malfunctioning of sensors or the ToF estimation algorithm during the measurement procedure.

A good estimation of the positions of the sensors can be obtained, if we have a good estimation of the missing entries of the ToF matrix. In general, it is a difficult task to infer missing entries of a matrix. However, it has recently been established that if the matrix is low-rank, a small random subset of its entries permits an exact reconstruction [CR08]. Because a modified version of the ToF matrix (when the entries are the squared ToF measurements) is low-rank, its missing entries can be accurately estimated, using matrix completion algorithms. To this end, we use OptSpace, a robust matrix completion algorithm developed by Keshavan et al. [KMO09].

On top of the missing entries, we also need to deal with an unknown time delay. This delay it due to the fact that, in practice, the impulse response of the piezoelectric and the time origin in the measurement procedure are not known and this causes an unknown time delay, which is added to the measurements. To infer this time delay simultaneously with the positions of the sensors, we propose a heuristic algorithm based on OptSpace.

In circular setups, the sensors are not necessarily on a circle and deviate from the circumference, which in fact necessitates the calibration process. We therefore need to assume that they are in the proximity of a circle (the precise statement is given later) and we are required to find the exact positions. Our approach is to estimate the local and random missing pairwise distances from which we can then infer the positions. As we have already mentioned, we show that a modified version of the ToF matrix has rank at most four; and by using this property, we propose our calibration procedure. The block diagram shown in Figure 3.1 summarizes the overall procedure.

## 3.1 Related Work

Calibration for circular tomography devices is a variant of sensor localization, the problem that we discussed in the previous chapter. As we have seen, in sensor localization, given the *local connectivity*, the objective is to devise an algorithm that can infer the global position of the sensors. We have also mentioned several methods that are deployed as a means to obtain the local information: the Signal Strength, the Angle of Arrival (AOA), and the Time Difference of Arrival (TDoA). Our problem here is naturally related to sensor localization when estimated TDoAs are used to measure the pairwise distances between nearby nodes. As we have noted, due to energy constraints, each node has a small communication range, compared to the field size they are installed. Consequently,

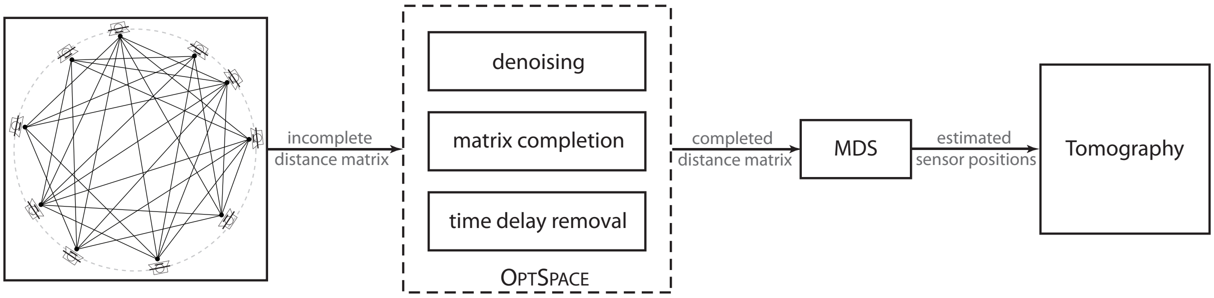

incomplete distance matrix
denoising
matrix completion
time delay removal
OptSpace
completed distance matrix
MDS
estimated sensor positions
Tomography

Figure 3.1 – Block diagram for the calibration procedure prior to ultrasound tomography. The incomplete distance matrix is passed through the OptSpace algorithm that denoises it, estimates the missing entries and removes the unknown time delay. The calibration is finished then by applying the MDS algorithm on the completed matrix that estimates the actual sensor positions.

only nodes within the communication range of each other can communicate hence estimate their pairwise TDoA's. This situation is depicted in Figure 3.2.

In the calibration problem, however, the local connectivity is precisely the kind of information that is missing. In fact, the beam width of transducers and the transient behaviour of ultrasound sensors prevent us from having reliable ToFs for nearby sensors (see Section 3.2). For this reason, in practice, the ToFs for close-by sensors are discarded and no information regarding their pairwise distances can be deduced. Consequently, in our scenario we are faced with a different setting from that of sensor localization:

– the pairwise distances of neighboring sensors are missing,
– only the pairwise distances of faraway sensors can be figured out from their ToFs.

This situation is demonstrated in Figure 3.2. By comparing these two scenarios in Figure 3.2, one can think of the calibration problem for ultrasound sensors as the dual problem of sensor localization. As a result, all sensor localization algorithms that rely on local information/connectivity are doomed to fail in our scenario. To confirm this fact, in Section 3.6, through numerical simulations we compare the performance of our proposed method with the state-of-the-art algorithms for sensor localization applied to our setting.

The first sensor localization algorithm we consider is MDS-MAP [SRZF03]. Recall from the previous chapter that this algorithm has two phases. First, the Euclidean distance of far-off sensors are approximated by the shortest path between them. Then, to estimate the relative positions of sensors, multidimensional scaling is applied to the approximated distance matrix. We can easily see, however, that given faraway sensors' distances, the shortest path is a very coarse estimate of the distance between the close-by sensors. This makes MDS-MAP perform very poorly in our setting.

One of the most prominent algorithms for centralized sensor localization is based on semi-definite programming (SDP). The method was first introduced by Biswas et al. in [BY04, BLT$^+$06] and solves the sensor localization problem using convex relaxation. From a practical point of view, the major problem of SDP-based methods is their heavy computations. According to [BLT$^+$06], the sensor localization for more than 200 sensors is computationally prohibitive. Theoretical guarantees of such methods were provided recently by Javanmard et al. [JM11]. As their results suggest, in the case of sensor localization, once the number of sensors grows, we cannot reduce the error of semidefinite programming below a threshold unless we increase the communication range, hence the power consumption of sensors. We will show, however, using matrix completion, the error

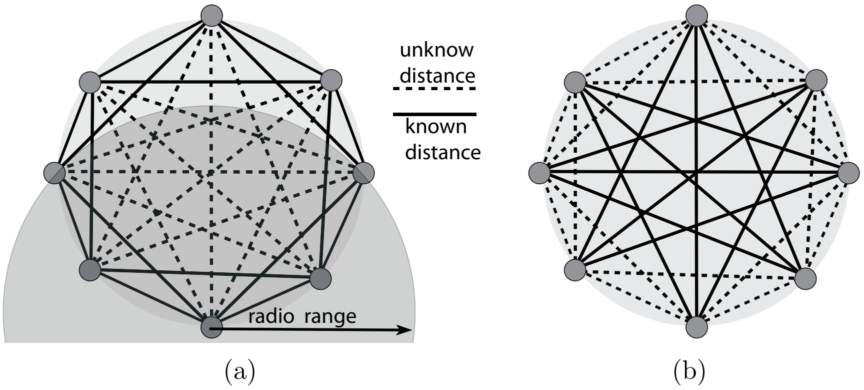


Figure 3.2 – In sensor localization (left figure) the local connectivity information is available and faraway ones are missing whereas in calibration (right figure) the opposite is true.

decreases as the number of transmitter/receivers grows.

In the core of our proposed method is matrix completion, an algorithm that recovers a low-rank matrix from its randomly known entries. It is easy to show that a matrix formed by pairwise distances is low-rank (see Lemma 3.1). Based on this property, Drineas et al. suggested using matrix completion for inferring the unknown distances [DJMI$^{+}$06]. In the previous chapter, we have mentioned that their analysis relied on the assumption that even for faraway nodes, there was a non-zero probability of communication. Thinking back to duality between sensor localization and our problem, this assumption suggests that, in our case, the pairwise distances of nearby transmitters/receivers can be obtained with a non-zero probability, an assumption that does not hold.

Recently, we have witnessed many improvements on the matrix completion algorithms. Candès et al. showed that a small random fraction of the entries suffices to reconstruct a low-rank matrix *exactly.* In a series of papers [KMO10, KMO09, KO09], Keshavan et al. studied an efficient implementation of a matrix completion algorithm, called OptSpace, and showed its optimality. Furthermore, they proved that their algorithm is robust against noise [KMO09]. In view of this progress, we show in this chapter that OptSpace is also capable of finding the missing nearby distances in our scenario hence providing us with their corresponding ToFs. To the best of our knowledge, all the above work, as well as the recent matrix completion algorithms [Rec09, RXH11], deal only with *random* missing entries. However, in our case, we encounter *structured* missing entries, in addition to the random ones (see Section 3.2), which is an aspect that was absent from the previous work. Therefore, one of our contributions is to provide analytic bounds on the error of OptSpace in the presence of structured missing entries.

## 3.2 Circular Time of Flight Tomography

The focus of this chapter is ultrasound tomography with circular apertures. In this setup, $n$ ultrasound transmitters and receivers are installed on the interior edge of a circular ring and an object with unknown acoustic characteristics is placed inside the ring. At

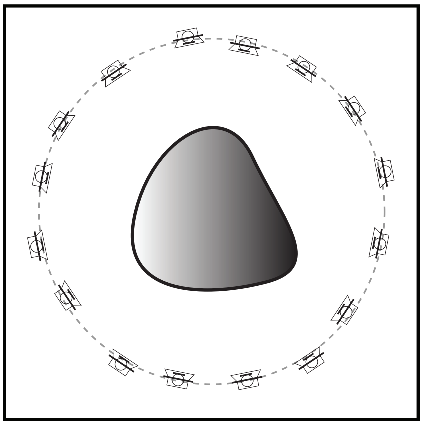


Figure 3.3 – Circular setup for ultrasound tomography considered in this work. Ultrasound transducers are distributed on the edge of a circular ring and the object with unknown characteristics is put inside. Transmitters and receivers are collocated. In practice, the positions deviate from an ideal circle.

each time instance a transmitter is fired, sending ultrasound signals with frequencies ranging from hundreds to thousands of kHz, while the rest of the sensors record the received signals. The same process is repeated for all the transmitters. Each one of $n$ sensors on the ring is capable of transmitting and receiving ultrasound signals. The aim of tomography in general is to use the recorded signals in order to reconstruct the characteristics of the enclosed object (e.g., sound speed, sound attenuation, etc.) The general configuration for such a tomography device is depicted in Figure 3.3. By employing these measurements, we can construct an inverse problem whose solution provides the acoustic characteristics of the enclosed object.

There are two common methods for solving the inverse problem. The solutions are either based on the wave equation [NW01] or the bent-ray theory [Jov08]. Both techniques consist of forward modeling the problem and comparing the simulation results with the measured data. For the details see [NW01] and [Jov08]. In both cases, in order to simulate the forward model and rely on the recorded data, a very precise estimate of the sensor positions is needed. In most applications (e.g., [JHSV07, SHD07, Nat08]), however, it is assumed that the sensors are positioned equidistantly apart on a circle and no later calibration is performed to find the exact sensor positions.

### Homogeneous Medium and Dimensionality Reduction

In order to estimate sensor positions, we utilize the ToF measurements for a homogeneous medium (e.g., water in the context of breast cancer detection). Let us assume that the mutual ToFs are stored in a matrix $\boldsymbol{T}$. In a homogeneous medium, the entries of $\boldsymbol{T}$ represent the travel time of sound in a straight line between each pair of a transmitter and receiver.

Knowing the temperature and the characteristics of the medium inside the ring, we can accurately estimate the constant sound speed $c_0$. Thus, it is reasonable to assume that $c_0$ is fixed and known. Having the ToFs for a homogeneous medium where no object is placed inside the ring, we can construct a distance matrix $\boldsymbol{D}$ consisting of the mutual distances between the sensors as

$$\boldsymbol{D} = [d_{i,j}] = c_0\boldsymbol{T}\,, \qquad \boldsymbol{T} = [t_{i,j}]\,, \quad i,j \in \{1,\cdots,n\} \tag{3.1}$$

where $t_{i,j}$ is the ToF between sensors $i$ and $j$ and $n$ is the total number of sensors around the circular ring. Notice that the only difference between the ToF matrix $\boldsymbol{T}$ and distance matrix $\boldsymbol{D}$, is the constant $c_0$. This is why in the rest of this chapter, our focus will be mainly on the distance matrix rather than the actual measured matrix $\boldsymbol{T}$.

As the enclosed medium is homogeneous, the matrix $\boldsymbol{T}$ is a symmetric matrix with zeros on the diagonal, and so is the matrix $\boldsymbol{D}$. Even though, the distance matrix $\boldsymbol{D}$ is full rank in general, a simple point-wise transform of its entries will lead to a low-rank matrix.

**Lemma 3.1.** *If one constructs the squared distance matrix* $\bar{\boldsymbol{D}}$ *as*

$$\bar{\boldsymbol{D}} = \left[d_{i,j}^2\right]\,,$$

*then the matrix* $\bar{\boldsymbol{D}}$ *has rank at most 4 [DJMI*$^+$*06] and if the sensors are placed on a circle, the rank is exactly 3.*

In practice, as we will explain in the next section, many of the the entries of the ToF matrix (or equivalently the distance matrix) are missing and there is an unknown time delay added to all the measurements.

### Time-of-Flight Estimation

Several methods for ToF estimation have been proposed in the signal processing community [Jov08, LHD$^+$09]. These methods are also known as time-delay estimation in acoustics [CHB04]. In these methods, the received signal is compared to a reference signal (ideally the sent signal), and the relative delay between the two signals is estimated. As the sent signal is not available in most cases, the received signal (that passed through the enclosed object) is compared to the received signal when the underlying medium is homogeneous. Unfortunately, we cannot use these methods in our scenario because in the calibration process we have only access to signals passed though the homogeneous medium.

Due to this limitation, we are forced to estimate the *absolute* ToFs. For this purpose, we use the first arrival method. This method probes the received signal and defines the time of flight as the time instant at which the received signal power exceeds a predefined threshold.

In practical screening systems, to record measurements for one fired transmitter, all the sensors are turned on simultaneously and after some unknown transition time (which is caused by the system structure, different sensor responses, etc.), the transmitter is fed with the electrical signal and the receivers start recording the signal. This unknown time can change for each pair of transmitters and receivers. We will see later that accurately estimating this unknown time delay is a critical part of the calibration process.

The beam width of the transducers and the transition behaviour of the ultrasonic sensors prevent the sensors from having a reliable ToF measurement for close-by neighbours. This results in incorrect ToF values for the sensors positioned close to each other. Therefore, numbering the sensors on the ring from 1 to $n$, in the ToF matrix $\boldsymbol{T}$, we will not have measurements on a certain band around the main diagonal and on the lower left

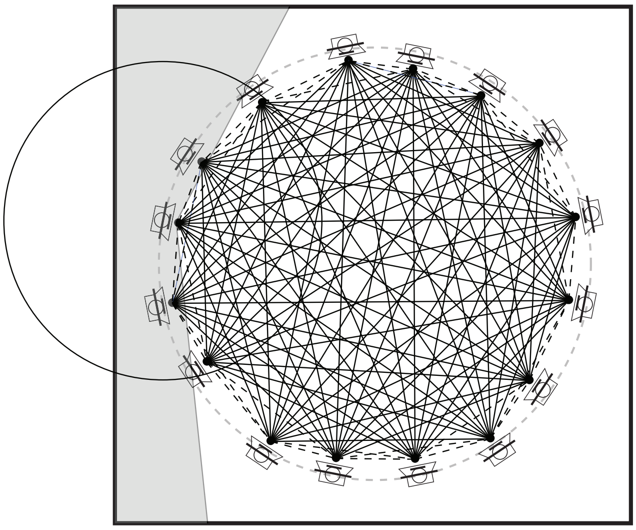


Figure 3.4 – The beam width of the transmitter causes the neighbouring sensors to not have reliable ToF measurements. This is shown by dashed lines in the figure. The area shown in gray corresponds to the part not covered by the transmitter's wave beam. This results in the structured missing entries.

$\boldsymbol{T} =$

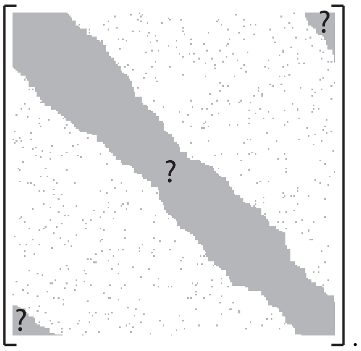


Figure 3.5 – A sample incomplete ToF matrix with structured and random missing entries.

and upper right parts as well. We call these missing entries *structured missing entries*. This is illustrated in Figure 3.4. The links shown by dashed lines do not contribute to the ToF measurements, because the beam for the transmitter does not cover the gray part.

During the measurement procedure, it can also happen that some sensors do not act properly and give outliers. Consequently, we perform a post processing on the measurements, in which a smoothness criterion is defined and the measurements not satisfying this criterion are removed from the ToF matrix. We call these entries *random missing entries*. An instance of the ToF matrix with the structured and random effects is shown in Figure 3.5, where $\boldsymbol{T}_{\text{inc}}$ denotes the incomplete ToF matrix and the gray entries correspond to the missing entries. Furthermore, in practice, the measurements are corrupted by noise.

The above mentioned problems result in an incomplete and noisy matrix $\boldsymbol{T}$ that cannot be used for position reconstruction unless the time delay effect is removed, the unknown entries are estimated, and the noise is smoothed.

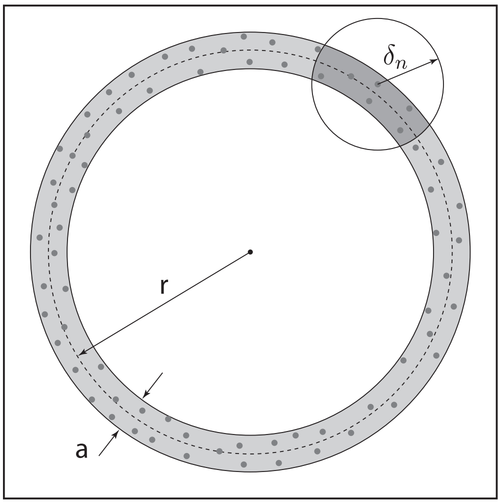


Figure 3.6 – Sensors are distributed around a circle of radius $r$ with small deviations from the circumference.

## 3.3 Problem Statement

We have observed that the distance matrix, when the aperture is in homogeneous medium, is calculated as in (3.1). We have also seen in the previous section that the measurements for the ToF matrix $\boldsymbol{T}$ have three major problems : they are noisy, some of them are missing, and the measurements contain some unknown time delay. For simplicity, we will assume that this time delay is constant for all the transmitters, i.e., all the transmitters send the electrical signal after some fixed but unknown delay $t_0$. Hence, we can rewrite the ToF matrix as follows

$$\widetilde{\boldsymbol{T}} = \boldsymbol{T} + t_0\boldsymbol{A} + \boldsymbol{Z}_0\,,$$

where $\boldsymbol{T}$ consists of ideal measurements for ToF, $\boldsymbol{Z}_0$ is the noise matrix and $\boldsymbol{A}$ is defined as

$$\boldsymbol{A} = [a_{i,j}]\,, \quad a_{i,j} = \begin{cases} 1 & \text{if } i \neq j\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

With the above considerations, the distance matrix can also be written as

$$\widetilde{\boldsymbol{D}} = \boldsymbol{D} + d_0\boldsymbol{A} + \boldsymbol{Z}\,, \tag{3.2}$$

where $\boldsymbol{D} = c_0\boldsymbol{T}$, $d_0 = c_0 t_0$, and $\boldsymbol{Z} = c_0\boldsymbol{Z}_0$.

In our model, we do not assume that the sensors are placed exactly on the ring. In practice, the sensor positions deviate from the circumference and our ultimate goal is to estimate these deviations or equivalently the correct positions (see Figure 3.6). The general positions taken by sensors are denoted by the set of vectors $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$.

To incorporate the structured missing entries, we assume that any measurements between sensors of distance less than $\delta_n$ are missing (see Figure 3.6). Hence, the number of structured missing entries depends on $\delta_n^2$. We are interested in the regime where we have a small number of structured missing entries per row in the large systems limit.

Accordingly, a typical range of interest for $\delta_n$ is $\delta_n = \Theta(r\sqrt{\log n/n})$. A random set of structured missing indices $S \subseteq [n] \times [n]$ is defined from $\{\boldsymbol{x}_i\}$ and $\delta_n$, by

$$S = \{(i,j) : \; d_{i,j} \le \delta_n \text{ and } i \neq j\} \; ,$$

where $d_{i,j} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$. Then, the structured missing entries are denoted by a matrix

$$\boldsymbol{D}^s_{i,j} = \begin{cases} \boldsymbol{D}_{i,j} & \text{if } (i,j) \in S \; , \\ 0 & \text{otherwise} \; . \end{cases}$$

Note that the matrix $\boldsymbol{D}^{\bar{s}} = \boldsymbol{D} - \boldsymbol{D}^s$ captures the noiseless distance measurements that are not affected by structured missing entries. This way, we can interpret the matrix $\boldsymbol{D}^s$ as additive noise in our model. Likewise, for the constant additive time delay we can define

$$\boldsymbol{A}^{\bar{s}}_{i,j} = \begin{cases} \boldsymbol{A}_{i,j} & \text{if } (i,j) \in S^\perp \; , \\ 0 & \text{otherwise} \; , \end{cases}$$

where $S^\perp$ denotes the complementary set of $S$. Next, to model the noise we add a random noise matrix $\boldsymbol{Z}^{\bar{s}}$.

$$\boldsymbol{Z}^{\bar{s}}_{i,j} = \begin{cases} \boldsymbol{Z}_{i,j} & \text{if } (i,j) \in S^\perp \; , \\ 0 & \text{otherwise} \; . \end{cases}$$

We do not assume a prior distribution on $\boldsymbol{Z}$, and the main theorem is stated for any general noise matrix $\boldsymbol{Z}$, deterministic or random. One practical example of $\boldsymbol{Z}$ is an i.i.d. Gaussian model.

Finally, to model the random missing entries, we assume that each entry of $\boldsymbol{D}^{\bar{s}} + t_0 c_0 \boldsymbol{A}^{\bar{s}} + \boldsymbol{Z}^{\bar{s}}$ is sampled with probability $p_n$. In the calibration data, we typically see a small number of random missing entries. Hence, in order to model it we assume that $p_n = \Theta(1)$. Let $E \subseteq [n] \times [n]$ denote the subset of indices which are not erased by random missing entries. Then a projection $\mathcal{P}_E : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is defined as

$$\mathcal{P}_E(\boldsymbol{M})_{i,j} = \begin{cases} \boldsymbol{M}_{i,j} & \text{if } (i,j) \in E \; , \\ 0 & \text{otherwise} \; . \end{cases}$$

We denote the observed measurement matrix by

$$\boldsymbol{N}^E = \mathcal{P}_E(\boldsymbol{D}^{\bar{s}} + d_0 \boldsymbol{A}^{\bar{s}} + \boldsymbol{Z}^{\bar{s}}), \tag{3.3}$$

where $d_0 = t_0 c_0$ is a constant. Notice that the matrix $\boldsymbol{N}^E$ has the same shape as $\boldsymbol{T}_{\text{inc}}$ shown already schematically in Figure 3.5.

Now we can state the goal of our calibration problem: Given the observed matrix $\boldsymbol{N}^E$ and the missing indices $S \cup E^\perp$, we want to estimate a low-rank matrix $\widehat{\boldsymbol{D}}$ close (in Frobenius norm) to the correct squared distance matrix $\widehat{\boldsymbol{D}}$. This issue is investigated under the name of matrix completion, which is the topic of the next section. Having estimated the missing entries of $\boldsymbol{N}^E$, the sensor positions can be then retrieved by applying MDS on $\widehat{\boldsymbol{D}}$ (we talked extensively about MDS in Section 2.3.1).

## 3.4 Matrix Completion

OptSpace, introduced in [KMO10], is an algorithm for recovering a low-rank matrix from noisy data with missing entries. The steps are shown in Algorithm 3.1. Let $\boldsymbol{M}$ be a rank-$q$ matrix of dimensions $n \times n$, $\boldsymbol{Z}$ the measurement noise, and $E$ the set of indices of the measured entries. Then, the measured noisy and incomplete matrix is $\boldsymbol{M}^E = \mathcal{P}_E(\boldsymbol{M}+\boldsymbol{Z})$.

**Algorithm 3.1** OptSpace [KMO10]

```
Input: Observed matrix $M^E = \mathcal{P}_E(M + Z)$.
Output: Estimate $M$.
1: Trimming: remove over-represented columns/rows;
2: Rank-$q$ projection on the space of rank-$q$ matrices according to (3.4);
3: Gradient descent: Minimize a cost function $F(\cdot)$ defined in (3.5);
```

In the trimming step, a row or a column is over-represented if it contains more samples than twice the average number of samples per row or column. These rows or columns can dominate the spectral characteristics of the observed matrix $\boldsymbol{M}^E$. Thus, some of their entries are removed uniformly at random from the observed matrix. Let $\widetilde{\boldsymbol{M}}^E$ be the resulting matrix of this trimming step. This trimming step is presented here for completeness, but in the case when $p_n$ is larger than some fixed constant (like in our case where $p_n = \Theta(p)$), $\boldsymbol{M}^E = \widetilde{\boldsymbol{M}}^E$ with high probability and the trimming step can be omitted.

In the second step, we first compute the singular value decomposition (SVD) of $\widetilde{\boldsymbol{M}}^E$.

$$\widetilde{\boldsymbol{M}}^E = \sum_{i=1}^{n} \sigma_i(\widetilde{\boldsymbol{M}}^E) u_i v_i^T \,,$$

where $\sigma_i(\cdot)$ denotes the $i$-th singular value of a matrix. Then, the rank-$q$ projection returns the matrix

$$\mathcal{P}_q(\widetilde{\boldsymbol{M}}^E) = (1/p_n) \sum_{i=1}^{q} \sigma_i(\widetilde{\boldsymbol{M}}^E) u_i v_i^T, \tag{3.4}$$

obtained by setting to 0 all but the $q$ largest singular values.

Starting from the initial guess provided by the rank-$q$ projection $\mathcal{P}_q(\widetilde{\boldsymbol{M}}^E)$, the final step solves a minimization problem stated as the following [KMO10]:
Given $\boldsymbol{X} \in \mathbb{R}^{n\times q}, \boldsymbol{Y} \in \mathbb{R}^{n\times q}$ with $\boldsymbol{X}^T\boldsymbol{X} = \mathbb{I}_{q\times q}$ and $\boldsymbol{Y}^T\boldsymbol{Y} = \mathbb{I}_{q\times q}$, define

$$F(\boldsymbol{X},\boldsymbol{Y}) = \min_{\boldsymbol{S}\in\mathbb{R}^{q\times q}} \mathcal{F}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{S})\,, \tag{3.5}$$

$$\mathcal{F}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{S}) = \frac{1}{2} \sum_{(i,j)\in E} (\boldsymbol{M}_{i,j} - (\boldsymbol{X}\boldsymbol{S}\boldsymbol{Y}^T)_{i,j})^2\,. \tag{3.6}$$

Values for $\boldsymbol{X}$ and $\boldsymbol{Y}$ are computed by minimizing $F(\boldsymbol{X},\boldsymbol{Y})$. This consists of writing $\mathcal{P}_q(\widetilde{\boldsymbol{M}}^E) = \boldsymbol{X}_0\boldsymbol{S}_0 Y_0^T$ and minimizing $F(\boldsymbol{X},\boldsymbol{Y})$ locally with initial condition $\boldsymbol{X} = \boldsymbol{X}_0$ and $\boldsymbol{Y} = \boldsymbol{Y}_0$. This last step is an attempt to get us as close as possible to the correct low-rank matrix $\boldsymbol{M}$.

Table 3.1 – Summary of Notation.

| | | | |
|---|---|---|---|
| $n$ | number of sensors | $\boldsymbol{D}$ | complete noiseless distance matrix |
| $r_0$ | radius of the circle from which the sensors deviate | $\bar{\boldsymbol{D}}$ | squared distance matrix |
| $a/2$ | maximum radial deviation from the circle | $\widetilde{\boldsymbol{D}}$ | noisy distance matrix |
| $\mathcal{P}_E$ | projection onto matrices with entries on index set $E$ | $\widehat{\boldsymbol{D}}$ | estimated squared distance matrix |
| $t_0$ | unknown time delay added to the ToF measurements | $\boldsymbol{Z}$ | noise matrix |
| $d_0$ | distance mismatch caused by the unknown time delay | $\boldsymbol{N}^E$ | observed matrix |
| $p_n$ | probability of having random missing entries | $\boldsymbol{X}$ | positions matrix |
| $\delta_n$ | radius of the circle defining structured missing entries | $\widehat{\boldsymbol{X}}$ | estimated positions matrix |
| $\boldsymbol{D}^s$ | distance matrix with observed entries on index set $S$ | $\boldsymbol{T}$ | ToF matrix |

## 3.5 Main Results

We have seen that the OptSpace algorithm is not directly applicable to the squared distance matrix because of the unknown delay. Since $\boldsymbol{A}$ in (3.2) is a full rank matrix, the matrix $\widetilde{\boldsymbol{D}} \odot \widetilde{\boldsymbol{D}} = [\tilde{d}^2_{i,j}]$ no longer has rank four. Moreover, because the measurements are noisy, we cannot hope to estimate the exact value of $d_0$. Therefore, in the following, we will provide error bounds on the reconstruction of the positions assuming that the time delay (equivalently $d_0$) is known. Afterwards, a heuristic method is proposed to estimate the value of $d_0$.

**Theorem 3.2.** *Assume $n$ sensors are distributed independently and uniformly at random on an annulus of width $a$ with central radius $r_0$ as in Figure 3.4. Let $\delta_n = \delta\, r_0 \sqrt{\log n/n}$. The resulting distance matrix $\boldsymbol{D}$ is corrupted by structured missing entries $\boldsymbol{D}^s$ and measurement noise $\boldsymbol{Z}^{\bar{s}}$. Further, the entries are missing randomly with probability $p_n = p$. Then, there exist constants $C_1$ and $C_2$, such that the output of* OptSpace $\widehat{\boldsymbol{D}}$ *achieves*

$$\frac{1}{n}\|\bar{\boldsymbol{D}} - \widehat{\boldsymbol{D}}\|_F \leq C_1 \left(\sqrt{\frac{\log n}{n}}\right)^3 + C_2 \frac{\|\boldsymbol{P}_E(\boldsymbol{Y}^{\bar{s}})\|_2}{p\, n}\,, \tag{3.7}$$

*with probability larger than $1-n^{-3}$, provided that the right-hand side is less than $\sigma_4(\bar{\boldsymbol{D}})/n$. In (3.7), $\boldsymbol{Y}^{\bar{s}}_{i,j} = {\boldsymbol{Z}^{\bar{s}}}^2_{i,j} + 2\boldsymbol{Z}^{\bar{s}}_{i,j}\boldsymbol{D}^{\bar{s}}_{i,j}$.*

The above theorem holds in general for any noise matrix $\boldsymbol{Z}$, deterministic or random. The above guarantees only hold up to numerical constants. To see how good OptSpace is in practice, we need to run numerical experiments. For more results supporting the robustness of OptSpace, we refer to [KO09].

Theorem 3.2 provides us with a good estimate of $\bar{\boldsymbol{D}}$. To find the positions of the sensors we can now apply MDS. More precisely, the estimated position matrix will be $\widehat{\boldsymbol{X}} = \mathsf{MDS}_2(\widehat{\boldsymbol{D}})$. Recall the distance we defined in the previous chapter between $\boldsymbol{X}$ and $\widehat{\boldsymbol{X}}$,

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) = \frac{1}{n}\left\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\right\|_F .$$

Then, the following theorem bounds the overall error of the process.

**Algorithm 3.2** Finding $d_0$.

```
Input: Matrix $\boldsymbol{N}^E$;
Output: Estimate $d_0$;
Construct the candidate set $\mathcal{C}_d = \{d_0^{(1)}, \ldots, d_0^{(M)}\}$ containing discrete values for $d_0$.
for $k = 1$ to $M$ do
    Set $\boldsymbol{N}^E_{(k)} = \boldsymbol{N}^E - d_0^{(k)} A^E$;
    Set $\bar{\boldsymbol{N}}^E_{(k)} = \boldsymbol{N}^E_{(k)} \odot \boldsymbol{N}^E_{(k)}$;
    Apply OptSpaceon $\bar{\boldsymbol{N}}^E_{(k)}$ and call the output $\hat{\boldsymbol{N}}^{(k)}$;
    Apply MDS and let $\boldsymbol{X}^{(k)} = \mathsf{MDS}_2(\hat{\boldsymbol{N}}^{(k)})$;
    Find $c^{(k)}$
        $c^{(k)} = \sum_{(i,j) \in E \cap S^{\perp}} \left(d_0^{(k)} + \|\boldsymbol{X}_i^{(k)} - \boldsymbol{X}_j^{(k)}\| - \boldsymbol{N}^E_{i,j}\right)^2$;
end for
Find $d_0$ satisfying
    $d_0 = d_0^{(l)}, \quad l = \arg\min_k c^{(k)}$;
```

**Theorem 3.3.** *Applying multidimensional scaling algorithm on $\widehat{\boldsymbol{D}}$, the error on the resulting coordinates will be bounded as follows*

$$d_{\text{inv}}(\boldsymbol{X}, \widehat{\boldsymbol{X}}) \leq C_1 \left(\sqrt{\frac{\log n}{n}}\right)^3 + C_2 \frac{\|\boldsymbol{P}_E(\boldsymbol{Y}^{\bar{s}})\|_2}{pn}\,, \tag{3.8}$$

*with probability larger than* $1 - 1/n^3$.

In Algorithm 3.2, we propose a heuristic method for estimating the value of $d_0$ along with completion of the squared distance matrix. In fact, this algorithm guarantees that after removing the effect of the time delay, we will find the best rank-4 approximation of the distance squared matrix. In other words, if we remove exactly the mismatch $d_0$, we will have an incomplete version of a rank-4 matrix and after reconstruction, the measured values will be close to those reconstructed.

## 3.6 Numerical Evaluation

As we have discussed in Section 3.1, we can treat the calibration problem as a special case of the sensor localization problem. However, there is a duality between the calibration problem and the traditional sensor localization problems. We mentioned in Section 3.2 that in the calibration problem the local distance/connectivity information is not available, whereas most of the state-of-the-art algorithms for sensor localization are based on the local information. In order to compare the performance of these algorithms with our proposed method, a set of simulations are performed. Figure 3.7 demonstrates the position reconstruction error of our method versus MDS-MAP [SRZF03], SDP-based [BLT$^+$06] and Svd-Reconstruct [DJMI$^+$06] algorithms, as the number of sensors grows.

For the simulations, we set the value of $a$ to 1 cm, $\delta$ to 1, $r_0$ to 10 cm, $t_0$ to zero, and the percentage of the random missing entries to 5. The distance measurements are corrupted by a white Gaussian noise with mean zero and standard deviation 0.6 mm. For each algorithm and each $n$, the experiment is performed 10 times for different positions and different noises, and the average error is taken. For the SDP-based method, we use the

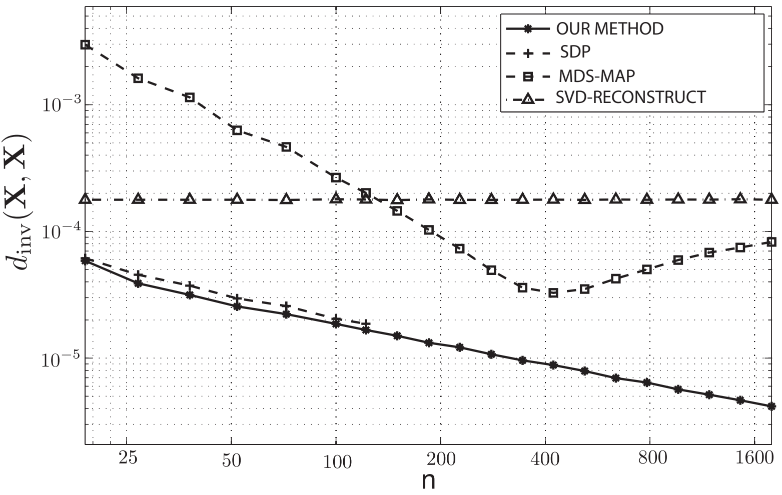


Figure 3.7 – Error in position estimation versus the number of sensors for different methods.

algorithm presented in [BLT$^+$06] and the code published by the same authors. For MDS-MAP, we estimate the shortest paths using Johnson's algorithm [Joh77]. Finally, for Svd-Reconstruct, we use the algorithm in [DJMI$^+$06]. In order to adapt the measurements with the assumptions of the method, we assume that $p_{ij} = 1 - 0.05 = 0.95$ for the measured points (note that 0.05 is on average the probability of having a random missing entry) and $\gamma_{ij} = 0$.

As the results in Figure 3.7 suggest, MDS-MAP and Svd-Reconstruct methods perform very badly compared to the other two methods. The poor performance of MDS-MAP is due to the fact that it highly relies on the presence of local distance information, whereas in our case these measurements are in fact missing. Recall that this method is based on estimating the missing distance measurements by the shortest path between the two sensors. However, we can easily see that given faraway sensor distances, the shortest path is a very coarse approximation of the distance between close-by sensors. Also note that, as the simulation results show, there is no guarantee that the estimation error will decrease as $n$ grows.

For Svd-Reconstruct, the unrealistic assumption that all the sensors have a non-zero probability of being connected causes the bad performance of the method. In our case, the probability that the close-by sensors are connected is zero because of the structured missing entries. In fact, since $p_{ij}$ is high, we can see this method as simply applying the classic MDS on the incomplete distance matrix. The surprising observation about the performance of this method is that the estimation error does not change much with $n$.

In contrast to the two aforementioned algorithms, the SDP-based method performs very well for estimating the sensor positions and the reconstruction error is very close to our proposed method. This is due to the fact that this method does not directly rely on the local distance information. In fact, the distance measurements are fed to the algorithm as the constraints of a convex optimization problem. However, as the number of sensors becomes large, the number of measurements also increases as does the number of constraints for the semi-definite program. This causes the method to fail for $n$ larger than 150. The same limitation is also reported by the authors of [BLT$^+$06].

To show the importance of calibration in an ultrasound scanning device, we perform another simulation as follows. Note that if the ToF measurements correspond to the correct positions of sensors (without time delay $t_0$), reconstruction of water will lead to a homogeneous region with values equal to the water sound speed. On the contrary, an inaccurate estimate of sensor positions and $t_0$ causes the inverse method to give incorrect values for the sound speed to compensate the effect of position mismatch and time delay.

In our experiment, we simulate the reconstruction of the water sound speed ($c_0 = 1500$) using the ToF measurements. In this setup, 200 sensors are distributed around a circle with radius $r_0 = 10$ cm where they deviate at most 5 mm from the circumference. Moreover, the time delay $t_0 = 4\mu s$ is added to ToF measurements.

In order to complete the distance matrix and find the time delay at the same time, we use Algorithm 3.2 in which we force the rank of $\bar{\boldsymbol{D}}$ to 4. The estimated value of $t_0$ is $4\mu s$, which is exactly as set in the simulation. The outputs of OptSpace algorithm together with those of the MDS method are used to estimate the positions. These positions are then fed to a tomography algorithm to reconstruct the water sound speed. The results of the overall process are shown in Figure 3.8.

In Figure 3.8a, the ToF matrix is not complete, it contains the time delay $t_0$, and the positions are not calibrated. The dark gray ring is caused by the non-zero time delay in the ToF measurements. In Figure 3.8b, the time mismatch is resolved by using Algorithm 3.2, but the sensor positions are not calibrated and the ToF matrix is still not complete. This figure shows clearly that finding the unknown time delay improves significantly the reconstruction image. Figure 3.8c shows the reconstructed medium when the ToF matrix is completed and time mismatch is removed, but the sensor positions are not yet calibrated. From this figure, it is confirmed that accurate time of flights are necessary but not sufficient to have a good reconstruction of the inclosed object. Finally, Figure 3.8d shows the reconstruction when the positions are also calibrated. Notice the change in the dynamic range for the last case.

# 3.7 Analysis

## 3.7.1 Proof of Theorem 3.2

To prove our main theorem, we apply Theorem 1.2 of [KMO09] to the rank-4 matrix $\bar{\boldsymbol{D}}$ and the observed matrix $\boldsymbol{N}^E = \mathcal{P}_E(\bar{\boldsymbol{D}} - \bar{\boldsymbol{D}}^s + \boldsymbol{Z}^{\bar{s}})$.

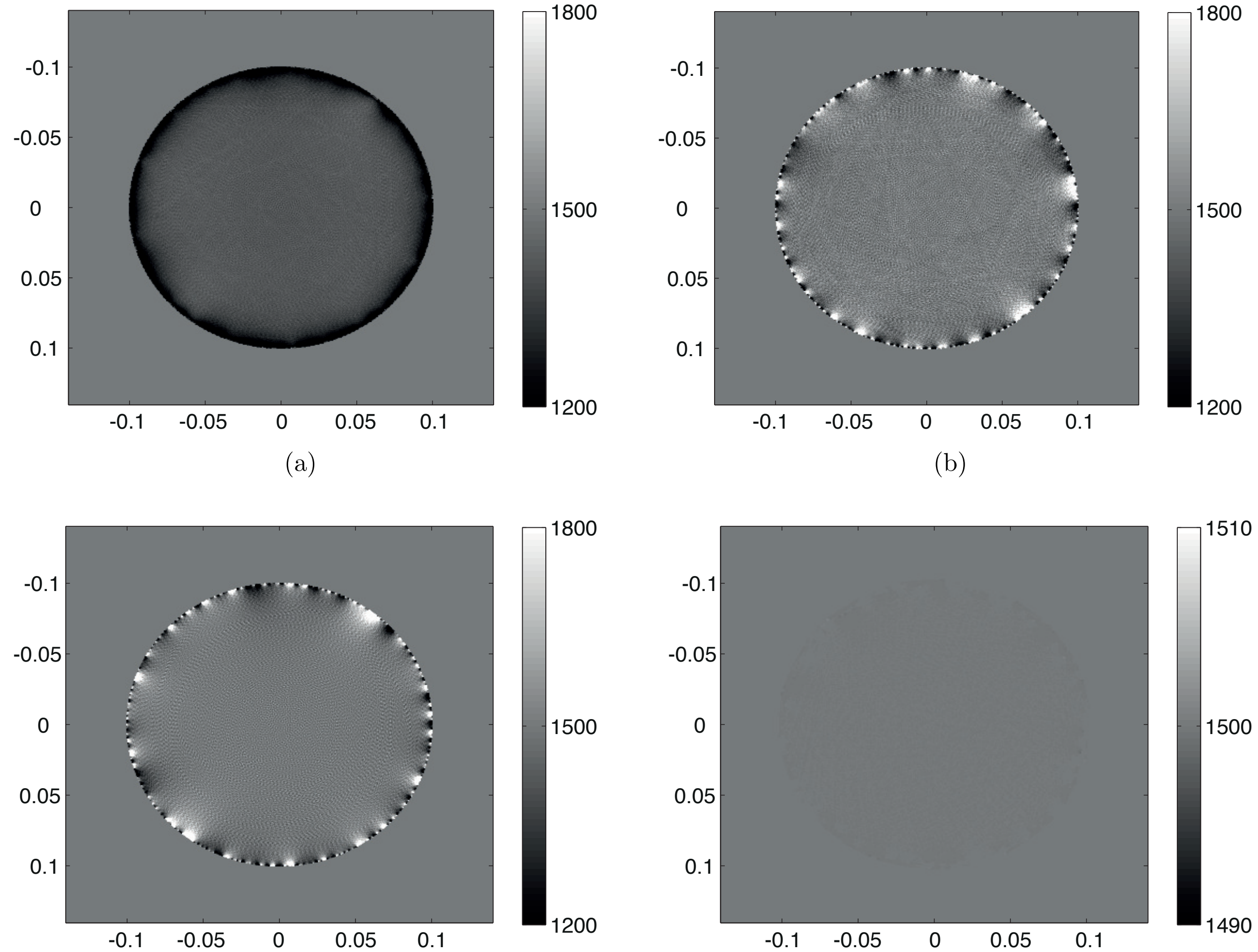
First, let us recall the *incoherence* property of $\bar{\boldsymbol{D}}$ from [KMO09]. A rank-4 symmetric matrix $\bar{\boldsymbol{D}} \in \mathbb{R}^{n\times n}$ is said to be *$\mu$-incoherent* if the following conditions hold. Let $\boldsymbol{U\Sigma U}^T$ be the singular value decomposition of $\bar{\boldsymbol{D}}$.

**A0.** For all $i \in [n]$, we have $\sum_{k=1}^{4} \boldsymbol{U}_{i,k}^2 \le 4\mu/n$.

**A1.** For all $i \in [n]$, $j \in [n]$, we have $\left|\bar{\boldsymbol{D}}_{i,j}/\sigma_1(\bar{\boldsymbol{D}})\right| \le \sqrt{4}\mu/n$.

The extra $1/n$ terms in the right hand side are due to the fact that, in this chapter, we assume that the singular vectors are normalized to unit norm, whereas in [KMO09] the singular vectors are normalized to have norm $\sqrt{n}$.

Theorem 1.2 of [KMO09] states that if a rank-4 matrix $\bar{\boldsymbol{D}}$ is *$\mu$-incoherent*, then the following is true with probability at least $1 - 1/n^3$. Let $\sigma_i(\bar{\boldsymbol{D}})$ be the $i$-th singular value of $\bar{\boldsymbol{D}}$ and $\kappa(\bar{\boldsymbol{D}}) = \sigma_1(\bar{\boldsymbol{D}})/\sigma_4(\bar{\boldsymbol{D}})$ be the condition number of $\bar{\boldsymbol{D}}$. Also, let $\widehat{\boldsymbol{D}}$ denote the estimation returned by OptSpace with input $N^E = \mathcal{P}_E(\bar{\boldsymbol{D}} - \bar{\boldsymbol{D}}^s + \boldsymbol{Y}^{\bar{s}})$. Then, there



(a) (b)

(c) (d)

Figure 3.8 – Results of the inversion procedure for finding the sound speed inside the ring filled with only water. (a) Reconstruction of homogeneous water when no calibration is performed. (b) Reconstruction after $t_0$ is removed from the ToF matrix, but the matrix is still incomplete and the positions are not calibrated. (c) Reconstruction when the matrix is also completed, but the positions are not yet calibrated. (d) Reconstruction with completed ToF matrix and calibrated positions.

exists numerical constants $C_1$ and $C_2$ such that

$$\frac{1}{n}||\bar{\boldsymbol{D}} - \widehat{\bar{\boldsymbol{D}}}||_F \leq C_1 \, \frac{\|\mathcal{P}_E(\bar{\boldsymbol{D}}^s)\|_2 + \|\mathcal{P}_E(\boldsymbol{Y}^{\bar{s}})\|_2}{p\,n} \; , \tag{3.9}$$

provided that

$$np \geq C_2 \mu^2 \kappa(\bar{\boldsymbol{D}})^6 \log n \; , \tag{3.10}$$

and

$$C_1 \, \frac{\|\mathcal{P}_E(\bar{\boldsymbol{D}}^s)\|_2 + \|\mathcal{P}_E(\boldsymbol{Y}^{\bar{s}})\|_2}{p\,n} \leq \frac{\sigma_4(\bar{\boldsymbol{D}})}{n} \; . \tag{3.11}$$

First, using Lemma 3.4, we show that the bound in (3.9) gives the desired bound in the theorem. Then, it is enough to show that there exists a numerical constant $N$ such that the conditions in (3.10) and (3.11) are satisfied with high probability for $n \geq N$.

**Lemma 3.4.** *In the model defined in the previous section, $n$ sensors are distributed independently and uniformly at random on a circular ring of width $a$ with central radius $r_0$. Then, with probability larger than $1 - n^{-3}$, there exists a constant $c$ such that*

$$\|\mathcal{P}_E(\bar{\boldsymbol{D}}^s)\|_2 \leq c\delta^3 (r_0 + a)^2 \Big(\sqrt{\frac{\log n}{n}}\Big)^3 p\, n\,. \tag{3.12}$$

The proof of this lemma can be found in [PKV11].

Now, to show that (3.10) holds with high probability for $n \geq C \log n/p$ for some constant $C$, we show that $\kappa \leq f_\kappa(r_0, a)$ and $\mu \leq f_\mu(r_0, a)$ with high probability, where $f_\kappa$ and $f_\mu$ are independent of $n$. Recall that $\kappa(\bar{\boldsymbol{D}}) = \sigma_1(\bar{\boldsymbol{D}})/\sigma_4(\bar{\boldsymbol{D}})$. We have

$$\begin{aligned}
\bar{\boldsymbol{D}}_{i,j} &= \|\boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_j\|^2 - 2\boldsymbol{x}_i^T \boldsymbol{x}_j \\
&= (r_0 + \rho_i)^2 + (r_0 + \rho_j)^2 - 2\boldsymbol{x}_i^T \boldsymbol{x}_j \\
&= 2r_0^2 + (2r_0\rho_i + \rho_i^2) + (2r_0\rho_j + \rho_j^2) - 2\boldsymbol{x}_i^T \boldsymbol{x}_j\,,
\end{aligned}$$

where $\rho_i$ is distributed in such a way that we have uniform distribution over the circular band. Thus, one can show that

$$\bar{\boldsymbol{D}} = \boldsymbol{A}\boldsymbol{S}\boldsymbol{A}^T\,,$$

where

$$\begin{aligned}
\boldsymbol{A} &= \begin{bmatrix} r_0 & x_{1,1} & x_{1,2} & 2r_0\rho_1 + \rho_1^2 \\ \vdots & \vdots & \vdots & \vdots \\ r_0 & x_{n,2} & x_{n,2} & 2r_0\rho_n + \rho_n^2 \end{bmatrix}, \\
\boldsymbol{S} &= \begin{bmatrix} 2 & 0 & 0 & \frac{1}{r_0} \\ 0 & -2 & 0 & 0 \\ 0 & 0 & -2 & 0 \\ \frac{1}{r_0} & 0 & 0 & 0 \end{bmatrix}.
\end{aligned}$$

One can write $\boldsymbol{S}$ as

$$\begin{aligned}
\boldsymbol{S} &= \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{-1}, \\
\boldsymbol{\Lambda} &= \mathrm{diag}\left(-2, -2, \frac{r_0 + \sqrt{1 + r_0^2}}{r_0}, \frac{r_0 - \sqrt{1 + r_0^2}}{r_0}\right),
\end{aligned}$$

It follows that

$$\sigma_1(\bar{\boldsymbol{D}}) \leq \frac{r_0 + \sqrt{1 + r_0^2}}{r_0} \sigma_1(\boldsymbol{A}\boldsymbol{A}^T)$$

and

$$\sigma_4(\bar{\boldsymbol{D}}) \geq \min\Big(2, \frac{\sqrt{1 + r_0^2} - r_0}{r_0}\Big) \sigma_4(\boldsymbol{A}\boldsymbol{A}^T).$$

We can compute the expectation of this matrix over the distribution of node positions. With a uniform distribution of the sensors over the circular ring, we have for the probability distribution of $\rho$:

$$p_\rho(\rho) = \frac{r_0 + \rho}{r_0 a}, \text{ for } -\frac{a}{2} \leq \rho \leq \frac{a}{2}\,.$$

Thus, the expectation of the matrix $\boldsymbol{A}^T\boldsymbol{A}$ is easily computed as

$$\mathbb{E}[\boldsymbol{A}^T\boldsymbol{A}] = \begin{bmatrix} nr_0^2 & 0 & 0 & nr_0\frac{a^2}{4} \\ 0 & \frac{n}{2}(r_0^2+\frac{a^2}{4}) & 0 & 0 \\ 0 & 0 & \frac{n}{2}(r_0^2+\frac{a^2}{4}) & 0 \\ nr_0\frac{a^2}{4} & 0 & 0 & n(\frac{a^2}{16}+\frac{r_0^2a^2}{3}) \end{bmatrix} .$$

Let the largest and smallest singular values of $\mathbb{E}[\boldsymbol{A}^T\boldsymbol{A}]$ be $n\sigma_{\max}(r_0,a)$ and $n\sigma_{\min}(r_0,a)$. Using the fact that $\sigma_i(\cdot)$ is a Lipschitz continuous function of its arguments, together with the Chernoff bound for large deviation of sums of i.i.d. random variables, we get

$$\mathbb{P}(\sigma_1(\boldsymbol{A}\boldsymbol{A}^T) > 2n\sigma_{\max}(r_0,a)) \le e^{-Cn} ,$$

$$\mathbb{P}(\sigma_1(\boldsymbol{A}\boldsymbol{A}^T) < (1/2)n\sigma_{\max}(r_0,a)) \le e^{-Cn} , \tag{3.13}$$

$$\mathbb{P}(\sigma_4(\boldsymbol{A}\boldsymbol{A}^T) < (1/2)n\sigma_{\min}(r_0,a)) \le e^{-Cn} , \tag{3.14}$$

for some constant $C$. Hence, with high probability,

$$\kappa(\bar{\boldsymbol{D}}) \le \frac{4\sigma_{\max}(r_0,a)}{\sigma_{\min}(r_0,a)} = f_\kappa(r_0,a).$$

Now, to bound $\mu$, note that with probability 1 the columns of $\boldsymbol{A}$ are linearly independent. Therefore, there exists a matrix $\boldsymbol{B} \in \mathbb{R}^{r\times r}$ such that $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{B}^T$ with $\boldsymbol{V}^T\boldsymbol{V} = \mathbb{I}$. The SVD of $\bar{\boldsymbol{D}}$ then reads $\bar{\boldsymbol{D}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T$ with $\boldsymbol{\Sigma} = \boldsymbol{Q}^T\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{Q}$ and $\boldsymbol{U} = \boldsymbol{V}\boldsymbol{Q}$ for some orthogonal matrix $\boldsymbol{Q}$. To show incoherence property **A0**, we need to show that, for all $i \in [n]$,

$$\|\boldsymbol{V}_i\|^2 \le \frac{4\mu}{n} .$$

Since $\boldsymbol{V}_i = \boldsymbol{B}^{-1}\boldsymbol{A}_i$, we have

$$\|\boldsymbol{V}_i\|^2 \le \sigma_4(\boldsymbol{B})^{-2}\|\boldsymbol{A}_i\|^2 \le \sigma_4(\boldsymbol{A})^{-2}\|\boldsymbol{A}_i\|^2.$$

Combined with

$$\|\boldsymbol{A}_i\|^2 = r_0^2 + (r_0+\rho_i)^2 + (2r_0\rho_i+\rho_i^2)^2 \le r_0^2 + (r_0+a)^2 + (2r_0a+a^2)^2$$

and (3.14), we have

$$\|\boldsymbol{U}_i\|^2 \le \frac{f_\mu(r_0,a)}{n} , \tag{3.15}$$

with high probability, where $f_\mu(r_0,a) = 2(r_0^2 + (r_0+a)^2 + (2r_0a+a^2)^2)$.

To show incoherence property **A1**, we use

$$|\bar{\boldsymbol{D}}_{ij}| \le (2r_0+a)^2$$

and

$$\sigma_1(\bar{\boldsymbol{D}}) \ge \frac{1}{4}n\,\sigma_{\min}(r_0,a)\min\left(2, \frac{\sqrt{1+r_0^2}-r_0}{r_0}\right)$$

from (3.13). Then,

$$\frac{|\bar{\boldsymbol{D}}_{ij}|}{\sigma_1(\bar{\boldsymbol{D}})} \leq \frac{g(r_0, a)}{n}\,, \tag{3.16}$$

with high probability, where

$$g(r_0, a) = \max\left(2, \frac{4r_0}{\sqrt{1+r_0^2}-r_0}\right)(2r_0+a)^2/\sigma_{\min}(r_0, a).$$

Combining (3.15) and (3.16), we see that the incoherence property is satisfied, with high probability.

Further, (3.11) holds, with high probability, if the right-hand side of (3.7) is less than $C_3\frac{\sqrt{1+r_0^2}+r_0}{r_0}\sigma_{\max}(r_0, a)$, since

$$\sigma_4(\bar{\boldsymbol{D}}) \leq \frac{1}{2}n\frac{\sqrt{1+r_0^2}+r_0}{r_0}\sigma_{\max}(r_0, a).$$

This completes the proof of Theorem 3.2.

### 3.7.2 Proof of Lemma 3.1

The proof for the general case where the sensors are not on a circle is provided in [DJMI$^+$06]. In the circular case however, we have $\bar{\boldsymbol{D}}_{i,j} = \|\boldsymbol{x}_i\|^2 + \|\boldsymbol{x}_j\|^2 - 2\boldsymbol{x}_i^T\boldsymbol{x}_j = 2r^2 - 2\boldsymbol{x}_i^T\boldsymbol{x}_j$, where $r$ is the circle radius. Thus, the squared distance matrix is decomposable to

$$\bar{\boldsymbol{D}} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T\,,$$

where

$$\begin{aligned}
\boldsymbol{V} &= \begin{bmatrix} r & x_{1,1} & x_{1,2} \\ \vdots & \vdots & \vdots \\ r & x_{n,1} & x_{n,2} \end{bmatrix}, \\
\boldsymbol{\Sigma} &= \begin{bmatrix} 2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix}.
\end{aligned}$$

This finishes the proof.

### 3.7.3 Proof of Theorem 3.3

The proof follows the same steps as in the proof of Theorem 2.1. In particular, note that $(\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L})$ has rank at most 4. Therefore,

$$\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\|_F \leq 2\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\|_2,$$

where we used the fact that for any matrix $\boldsymbol{A}$ of rank $k$ we have $\|A\|_F \leq \sqrt{k}\|\boldsymbol{A}\|_2$. Furthermore, the spectral norm can be bounded in terms of $\bar{\boldsymbol{D}}$ and $\widehat{\boldsymbol{D}}$ as follows.

$$\begin{aligned}
\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\|_2 &\overset{(a)}{\leq} \|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \frac{1}{2}\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L}\|_2 + \|\frac{1}{2}\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\|_2 \\
&\overset{(b)}{\leq} \frac{1}{2}\|\boldsymbol{L}(\bar{\boldsymbol{D}} - \widehat{\boldsymbol{D}})\boldsymbol{L}\|_2 + \frac{1}{2}\|\boldsymbol{L}(-\bar{\boldsymbol{D}} + \widehat{\boldsymbol{D}})\boldsymbol{L}\|_2,
\end{aligned} \tag{3.17}$$

where in $(a)$, we used the triangle inequality and $\boldsymbol{L}\widehat{\boldsymbol{X}} = \widehat{\boldsymbol{X}}$. In $(b)$, we used the fact that for any matrix $\boldsymbol{A}$ of rank 2, $\|\frac{1}{2}\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L} - \widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\|_2 \leq \|\frac{1}{2}\boldsymbol{L}\widehat{\boldsymbol{D}}\boldsymbol{L} - \boldsymbol{A}\|_2$. In particular, by setting $\boldsymbol{A} = \frac{1}{2}\boldsymbol{L}\bar{\boldsymbol{D}}\boldsymbol{L}$ the second term in (3.17) follows. Since $\boldsymbol{L}$ is a projection matrix we have $\|\boldsymbol{L}\|_2 = 1$. Hence, from (3.17) we can conclude that

$$\|\boldsymbol{L}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{L} - \boldsymbol{L}\widehat{\boldsymbol{X}}\widehat{\boldsymbol{X}}^T\boldsymbol{L}\|_2 \leq \|\widehat{\bar{\boldsymbol{D}}} - \bar{\boldsymbol{D}}\|_2.$$

This immediately leads to Theorem 3.3.

# Part II

# Where Is It?

“– ···· ·   –– ·· –·· –·· ·–·· ·   ––– ··–·
· ···– · ·–· –·––   ··· ··– –·–· –·–· · ··· ··· ··–· ··– ·–··
·––· ·–· ––– ·––– · –·–· –   ·–·· ––– ––– –·– ···
·–·· ·· –·– ·   ·–   –·· ·· ··· ·– ··· – · ·–· ······ ”

·–· ––– ··· ·– –··· · – ····   –– ––– ··· ···
–·–· ·– –· – ––– ·–·

# 4

# Graph-Constrained Group Testing

In this chapter [1] we introduce the graph-constrained group testing problem motivated by applications in network tomography, sensor networks and infection propagation.

Group testing involves identifying at most $d$ defective items out of a set of $n$ items. In non-adaptive group testing, which is the subject of this chapter, we are given an $m \times n$ binary matrix, $\boldsymbol{M}$, usually referred to as a test or measurement matrix. Ones on the $j$th row of $\boldsymbol{M}$ indicate which subset of the $n$ items belongs to the $j$th pool. A test is conducted on each pool; a positive outcome indicating that at least one defective item is part of the pool; and a negative test indicating that no defective items are part of the pool. The conventional group testing problem is to design a matrix $\boldsymbol{M}$ with minimum number of rows $m$ that guarantees error free identification of the defective items. While the best known (probabilistic) pooling design requires a test matrix with $m = O(d^2 \log(n/d))$ rows, and an almost-matching lower bound of $m = \Omega(d^2(\log n)/(\log d))$ is known on the number of pools (cf. [DH99, Chapter 7]), the size of the optimal test still remains open.

Note that in the standard group testing problem the test matrix $\boldsymbol{M}$ can be designed arbitrarily. In this chapter we consider a generalization of the group testing problem to the case where the matrix $\boldsymbol{M}$ must conform to constraints imposed by a graph $G = (V, E)$. In general, as we will describe shortly, such problems naturally arise in several applications such as network tomography [Duf06, NT07], sensor networks [NT06], and infection propagation [CHKV09].

In our graph group testing scenario the $n$ items are either vertices or links (edges) of the graph; at most $d$ of them are defective. The task is to identify the defective vertices or edges. The test matrix $\boldsymbol{M}$ is constrained as follows: for items associated with vertices each row must correspond to a subset of vertices that are connected by a path on the graph; similarly, for items associated with links each row must correspond to links that form a path on $G$. The task is to design an $m \times n$ binary test matrix with minimum number of rows $m$ that guarantees error free identification of the defective items.

We will next describe several applications, which illustrate the graph constrained group testing problem.

---

[1] This chapter is the result of a collaboration with M. Cheraghchi, S. Mohajer, and V. Saligrama.

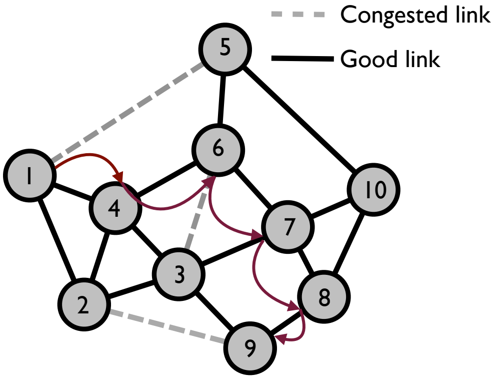


Figure 4.1 – In our setup, graph constrained group testing, the route 1 → 4 → 6 → 7 → 8 → 9 is valid while 2 → 6 → 5 is not.

**Network Tomography:** For a given network, identification of congested links from end-to-end path measurements is one of the key problems in network tomography [NT07], [Duf06]. In many settings of today's IP networks, there is one or a few links along the path which cause packet losses along the path. Finding the locations of such congested links is sufficient for most of the practical applications.

This problem can be phrased as a graph-constrained group testing problem as follows. We model the network as a graph $G = (V, E)$ where the set $V$ denotes the network routers/hosts and the set $E$ denotes the communication links (see Fig. 4.1). Suppose, we have a monitoring system that consists of one or more end hosts (so called *vantage* points) that can send and receive packets. Each vantage point sends packets through the network by assigning the routes and the end hosts.

All measurement results (i.e., whether each packet has reached its destination) will be reported to a central server whose responsibility is to identify the congested links. Since the network is given, not any route is a valid one. A vantage point can only assign those routes which form a path in the graph $G$. The question of interest is to determine the number of measurements that is needed in order to identify the congested links in a given network.

**Sensor Networks:** The network tomography problem is also present in wireless sensor networks (WSN). As described in [NT06] the routing topology in WSN is constantly changing due to the inherent ad-hoc nature of the communication protocols. The sensor network is static with a given graph topology such as a geometric random graph. Sensor networks can be monitored passively or actively. In passive monitoring, at any instant, sensor nodes form a tree to route packets to the sink. The routing tree constantly changes unpredictably but must be consistent with the underlying network connectivity. A test is considered positive if the arrival time is significantly large, which indicates that there is at least one defective sensor node or a congested link. The goal is to identify defective links or sensor nodes based on packet arrival times at the sink. In active monitoring network nodes continuously calculate some high level, summarized information such as the average or maximum energy level among all nodes in the network. When the high level information indicates congested links, a low level and more energy consuming procedure is used to

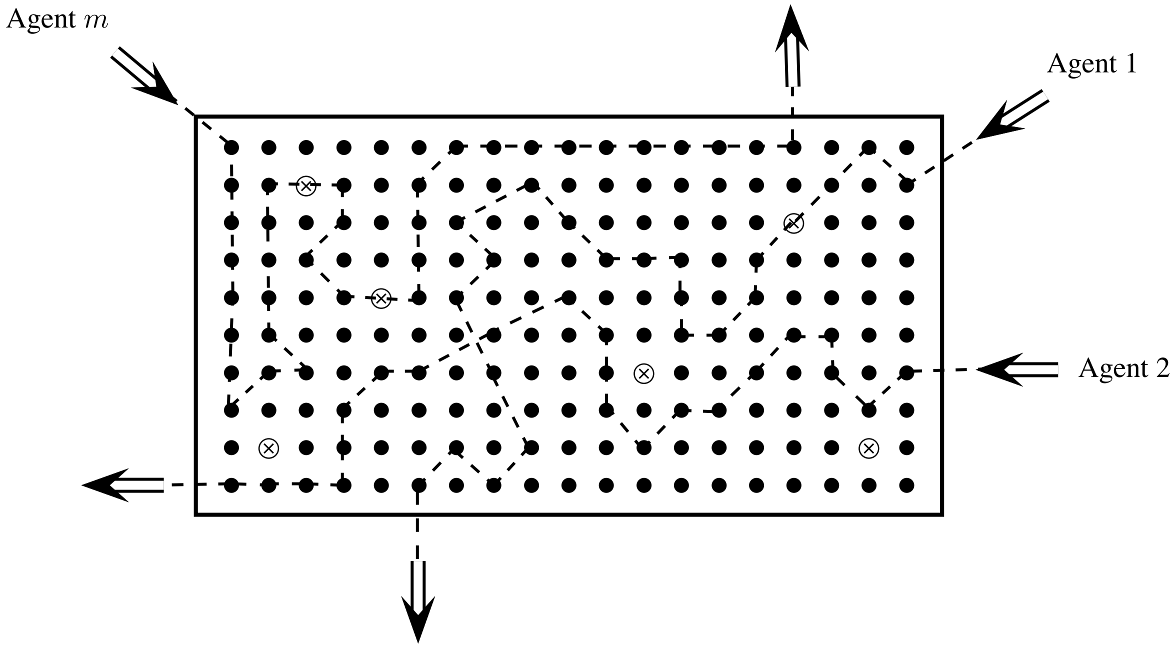


Figure 4.2 – Collective sampling using agents. The ⊗ symbols represent infected people whereas the healthy population is indicated by • symbols. The dashed lines show the group of people contacted by each agent [CHKV09].

accurately locate the trouble spots.

**Infection Propagation:** Suppose that we have a large population where only a small number of people are infected by a certain viral sickness (e.g., a flu epidemic). The task is to identify the set of infected individuals by sending agents among them. Each agent contacts a pre-determined or randomly chosen set of people. Once an agent has made contact with an infected person, there is a chance that he gets infected, too. By the end of the testing procedure, all agents are gathered and tested for the disease. While this problem has been described in [CHKV09], the analysis ignores the inherent graph constraints that need to be further imposed. It is realistic to assume that, once an agent has contacted a person, the next contact will be with someone in close proximity of that person. Therefore, in this model we are given a random geometric graph that indicates which set of contacts can be made by an agent (see Fig. 4.2). Now, the question is to determine the number of agents that is needed in order to identify the set of infected people.

These applications present different cases where graph constrained group testing can arise. However, there are important distinctions. In the wired network tomography scenario the links are the items and each row of the matrix $\boldsymbol{M}$ is associated with a route between any two vantage points. A test is positive if a path is congested, namely, if it contains at least one congested link. Note that in this case since the routing table is assumed to be static, the route between any two vantage points is fixed. Consequently, the matrix $\boldsymbol{M}$ is deterministic and the problem reduces to determining whether or not the matrix $\boldsymbol{M}$ satisfies identifiability.

Our problem is closer in spirit to the wireless sensor network scenario. In the passive case the links are the items and each row of the matrix $\boldsymbol{M}$ is associated with a route between a sensor node and the sink. A test is positive if a path is congested, namely, if it contains at least one congested link. Note that in this case since the routing table is constantly changing, the route between a sensor node and the sink is constantly changing as well. Nevertheless the set of possible routes must be drawn from the underlying connectivity graph. Consequently, the matrix $\boldsymbol{M}$ can be assumed to be random and the problem is to determine how many different tests are required to identify the congested links. Note that, in contrast to the wired scenario, tests conducted between the same sensor node and sink yields new information here. A similar situation arises in the active monitoring case as well. Here one could randomly query along different routes to determine whether or not a path is congested. These tests can be collated to identify congested links. Note that in the active case the test matrix $\boldsymbol{M}$ is amenable to design in that one could selectively choose certain paths over others by considering weighted graphs.

Motivated by the WSN scenario we describe pool designs based on random walks on graphs. As is well known a random walk is the state evolution on a finite reversible Markov chain. Each row of the binary test matrix is derived from the evolution of the random walk, namely, the ones on the $j$th row of $\boldsymbol{M}$ correspond to the vertices visited by the $j$th walk. This is close to the WSN scenario because as in the WSN scenario the path between two given nodes changes randomly. We develop several results in this context.

First, we consider random walks that start either at a random node or an arbitrary node but terminate after some appropriately chosen number of steps $t$. By optimizing the length of the walk we arrive at an interesting result for important classes of graphs. Specifically we show that the number of tests required to identify $d$ defective items is essentially similar to that required in conventional group testing problems, except the fact that an extra term appears which captures the topology of the underlying graph. The best known result for the number of tests required when no graph constraints are imposed scales as $O(d^2 \log(n/d))$. For the graph constrained case we show that with $m = O(d^2 T^2(n) \log(n/d))$ non-adaptive tests one can identify the defective items, where $T(n)$ corresponds to the mixing time of the underlying graph $G$. Consequently, for the Erdős-Rényi random graph $G(n,p)$ with $p = \Omega((\log^2 n)/n)$, as well as expander graphs with constant spectral gap, it follows that $m = O(d^2 \log^3 n)$ non-adaptive tests are sufficient to identify $d$ defective items. In particular, for a complete graph where no pooling constraint is imposed, we have $T(n) = 1$, and therefore, our result subsumes the well-known result for the conventional group testing problem.

Next we consider unbounded-length random walks that originate at a source node and terminate at a sink node. Both the source node and the sink node can either be arbitrary or be chosen uniformly at random. This directly corresponds to the network tomography problem that arises in the WSN context. This is because the source nodes can be viewed as sensor nodes, while the sink node maybe viewed as the fusion center, where data is aggregated. At any instant, we can assume that a random tree originating at the sensor nodes and terminating at the sink is realized. While this random tree does not have cycles, there exist close connections between random walks and randomly generated trees. Indeed, it is well known that the so called loop-erased random walks, obtained by systematically erasing loops in random walks, to obtain spanning trees, is a method for sampling spanning trees from a uniform distribution [Wil96]. In this scenario, we show that $m = O(d^3 \log^3 n)$ non-adaptive tests are sufficient to identify $d$ defective items. By considering complete graphs we also establish that the cubic dependence on $d$ in this result cannot be improved.

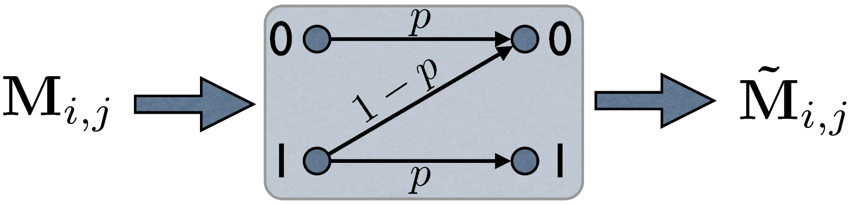


Figure 4.3 – In the dilution model, each element of matrix $\boldsymbol{M}_{ij}$ passes through a $Z$-channel. The zeros of the matrix remain zeros while the ones are converted to zeros with probability $1-p$.

We will also consider noisy counterparts of the graph constrained group testing problem, where the outcome of each measurement may be independently corrupted (flipped) with probability[2] $0 \leq q < 1/2$. We develop parallel results for these cases. In addition to a setting with noisy measurement outcomes, these results can be used in a so called *dilution model* (as observed in [AS09, CHKV09]). In this model, each item can be *diluted* in each test with some a priori known probability (see Figure 4.3). In a network setting, this would correspond to the case where a test on a path with a congested link can turn out to be negative with some probability. We show that similar scaling results hold for this case as well.

## 4.1 Related Work

Since its emergence decades ago, group testing has found a large number of applications that are too numerous to be extensively treated here. We particularly refer to applications in industrial quality assurance [SG59], DNA library screening [PL94], software testing [BG02], and multi-access communications [Wol85], among others.

Several variations of classical group testing have been studied in the literature that possess a graph theoretic nature. A notable example is the problem of learning hidden sparse subgraphs (or more generally, hypergraphs), defined as follows (cf. [Aig88]): Assume that, for a given graph, a small number of edges are marked as defective. The problem is to use a small number of measurements of the following type to identify the set of defective edges: Each measurement specifies a subset of vertices, and the outcome would be positive iff the graph induced on the subset contains a defective edge. Another variation concerns group testing with constraints defined by a rooted tree. Namely, the set of items corresponds to the leaves of a given rooted tree, and each test is restricted to pool all the leaves that descend from a specified node in the tree (see [DH99, Chapter 12]).

We should also point out the relationship between our setup and the compressed sensing framework [Don06, CRT06]. We primarily deal with Boolean operations on binary valued variables in this chapter, namely, link states are binary valued and the measurements are boolean operations on the link states. Nevertheless, the techniques described

---

[2] It is clear that if $q > 1/2$, one can first flip all the outcomes, and then reduce the problem to the $q < 1/2$ regime. For $q = 1/2$, since we only observe purely random noise, there is no hope to recover from the errors.

here can be extended to include non-boolean operations and non-binary variables as well. Specifically, suppose there is a sparse set of links that take on non-zero values. These non-zero values could correspond to packet delays, and packet loss probabilities along each link. Measurements along each path provides aggregate delay or aggregate loss along the path. The set of paths generated by $m$ random walks forms a $m \times |E|$ routing matrix $\boldsymbol{M}$. For an appropriate choice of $m$ and graphs studied in this chapter, it turns out (see [XMT11]) that such routing matrices belongs to the class of so called expander matrices. These expander type properties in turn obey a suitable type of restricted-isometry-property (1-RIP) [BGI$^+$08]. Such properties in turn are sufficient for recovering sparse vectors using $\ell_1$ optimization techniques. Consequently, the results of this chapter have implications for compressed sensing on graphs.

## 4.2 Definitions and Notation

In this section we introduce some tools, definition and notations which are used throughout the chapter.

### Disjunct Matrix

For two given boolean vectors $S$ and $T$ of the same length we denote their element-wise logical or by $S \vee T$. More generally, we will use $\bigvee_{i=1}^{d} S_i$ to denote the element-wise or of $d$ boolean vectors $S_1, \ldots, S_d$. The logical subtraction of two boolean vectors $S = (s_1, \ldots, s_n)$ and $T = (t_1, \ldots, t_n)$, denoted by $S \setminus T$, is defined as a boolean vector which has a 1 at position $i$ if and only if $s_i = 1$ and $t_i = 0$. We also use $|S|$ to show the number of 1's in (i.e., the Hamming weight of) a vector $S$.

We often find it convenient to think of boolean vectors as characteristic vectors of sets. That is, $x \in \{0,1\}^n$ would correspond to a set $X \subseteq [n]$ (where $[n] := \{1, \ldots, n\}$) such that $i \in X$ iff the entry at the $i$th position of $x$ is 1. In this sense, the above definition extends the set-theoretic notions of union, subtraction, and cardinality to boolean vectors.

Matrices that are suitable for the purpose of group testing are known as *disjunct* matrices. The formal definition is as follows.

An $m \times n$ boolean matrix $\boldsymbol{M}$ is called $d$-disjunct, if, for every column $S_0$ and every choice of $d$ columns $S_1, \ldots, S_d$ of $\boldsymbol{M}$ (different from $S_0$), there is at least one row at which the entry corresponding to $S_0$ is 1 and those corresponding to $S_1, \ldots, S_d$ are all zeros. More generally, for an integer $e \geq 0$, the matrix is called $(d, e)$-disjunct if for every choice of the columns $S_i$ as above, they satisfy

$$|S_0 \setminus \bigvee_{i=1}^{d} S_i| > e.$$

A $(d, 0)$-disjunct matrix is said to be $d$-disjunct.

A classical observation in group testing theory states that disjunct matrices can be used in non-adaptive group testing schemes to distinguish sparse boolean vectors (cf. [DH99]). More precisely, suppose that a $d$-disjunct matrix $\boldsymbol{M}$ with $n$ columns is used as the measurement matrix; i.e., we assume that the rows of $\boldsymbol{M}$ are the characteristic vectors of the pools defined by the scheme. Then, the test outcomes obtained by applying the scheme on two distinct $d$-sparse vectors of length $n$ must differ in at least one position. More generally, if $\boldsymbol{M}$ is taken to be $(d, e)$-disjunct, the test outcomes must differ in at least $e + 1$

positions. Thus, the more general notion of $(d, e)$-disjunct matrices is useful for various "noisy" settings, where we are allowed to have a few false outcomes (in particular, up to $\lfloor (e-1)/2 \rfloor$ incorrect measurement outcomes can be tolerated by $(d, e)$-disjunct matrices without causing any confusion).

### Graph Consistent Disjunct Matrix

For our application, sparse vectors (that are to be distinguished) correspond to boolean vectors encoding the set of defective vertices (or edges) in a given undirected graph. The encoding is such that the coordinate positions are indexed by the set of vertices (edges) of the graph and a position contains 1 iff it corresponds to a defective vertex (edge). Moreover, we aim to construct disjunct matrices that are also constrained to be *consistent* with the underlying graph.

Let $G = (V, E)$ be an undirected graph, and $\boldsymbol{A}$ and $\boldsymbol{B}$ be boolean matrices with $|V|$ and $|E|$ columns, respectively. The columns of $\boldsymbol{A}$ are indexed by the elements of $V$ and the columns of $\boldsymbol{B}$ are indexed by the elements of $E$. Then,

- The matrix $\boldsymbol{A}$ is said to be *vertex-consistent* with $G$ if each row of $\boldsymbol{A}$, seen as the characteristic vector of a subset of $V$, exactly represents the set of vertices visited by some walk on $G$.
- The matrix $\boldsymbol{B}$ is said to be *edge-consistent* with $G$ if each row of $\boldsymbol{B}$, seen as the characteristic vector of a subset of $E$, exactly corresponds to the set of edges traversed by a walk on $G$.

Note that the choice of the walk corresponding to each row of $\boldsymbol{A}$ or $\boldsymbol{B}$ need not be unique. Moreover, a walk may visit a vertex (or edge) more than once.

### Mixing Time and Conductance

An undirected graph $G = (V, E)$ is called $(D, c)$-uniform, for some $c \geq 1$, if the degree of each vertex $v \in V$ (denoted by $\deg(v)$) is between $D$ and $cD$. Throughout this chapter, the constraint graphs are considered to be $(D, c)$-uniform, for an appropriate choice of $D$ and some (typically constant) parameter $c$. When $c = 1$, the graph is $D$-regular.

Before being able to define the mixing time, we need to define the distance between two probability distributions. In particular, the *point-wise distance* of two probability distributions $\mu, \mu'$ on a finite space $\Omega$ is defined as

$$\|\mu - \mu'\|_\infty := \max_{i \in \Omega} |\mu(i) - \mu'(i)|,$$

where $\mu(i)$ (resp., $\mu'(i)$) denotes the probability assigned by $\mu$ (resp., $\mu'$) to the outcome $i \in \Omega$. We say that the two distributions are $\delta$-close if their point-wise distance is at most $\delta$.

For notions such as random walks, stationary distribution and mixing time we refer to standard books on probability theory, Markov chains, and randomized algorithms. In particular for an accessible treatment of the basic notions, see [MR95, Chapter 6] or [MU05, Chapter 7]. The particular variation of the mixing time that we will use in this work is defined with respect to the point-wise distance as follows.

**Definition 4.1.** *Let $G = (V, E)$ with $|V| = n$ be a $(D, c)$-uniform graph and denote by $\mu$ its stationary distribution. For $v \in V$ and an integer $\tau$, denote by $\mu_v^\tau$ the distribution that a random walk of length $\tau$ starting at $v$ ends up at. Then, the $\delta$-*mixing time* of $G$ (with* *respect to the $\ell_\infty$ norm*[3]*) is the smallest integer $t$ such that $\|\mu_v^\tau - \mu\|_\infty \leq \delta$, for $\forall \tau \geq t$ and $\forall v \in V$. For concreteness, we define the quantity $T(n)$ as the $\delta$-mixing time of $G$ for $\delta := (1/2cn)^2$.*

For a graph to have a small mixing time, a random walk starting from any vertex must quickly induce a uniform distribution on the vertex set of the graph. Intuitively this happens if the graph has no "bottle necks" at which the walk can be "trapped", or in other words, if the graph is "highly connected". The standard notion of *conductance*, as defined below, quantifies the connectivity of a graph.

**Definition 4.2.** *Let $G = (V, E)$ be a graph on $n$ vertices. For every $S \subseteq V$, define $\Delta(S) := \sum_{v \in S} \deg(v)$, $\bar{S} := V \setminus S$, and denote by $E(S, \bar{S})$ the number of edges crossing the cut defined by $S$ and its complement. Then the* conductance *of $G$ is defined by the quantity*

$$\Phi(G) := \min_{S \subseteq V \colon \Delta(S) \leq |E|} \frac{E(S, \bar{S})}{\Delta(S)}.$$

### Random and Expander Graphs

We also formally define two important classes of graphs, for which we will specialize our results.

**Definition 4.3.** *Take a complete graph on $n$ vertices, and remove edges independently with probability $1 - p$. The resulting graph is called the Erdős-Rényi random graph, and denoted by $G(n, p)$.*

**Definition 4.4.** *For a graph $G = (V, E)$ with $|V| = n$, the (edge) expansion of $G$ is defined as*

$$h(G) = \min_{S \subseteq V \colon 0 < |S| \leq \frac{n}{2}} \frac{E(S, \bar{S})}{|S|}.$$

*A family $\mathcal{G}$ of $D$-regular graphs is called an (edge) expander family if there exists a constant $\sigma > 0$ such that $h(G) \geq \sigma$ for each $G \in \mathcal{G}$. In particular each $G \in \mathcal{G}$ is called an expander graph.*

For a general study of Erdős-Rényi random graphs and their properties we refer to the fascinating book of Bollobás [BC88]. For the terminology on expander graphs, we refer the reader to the excellent survey by Hoory, Linial and Wigderson [HLW06].

### Random Walk

Consider a particular random walk $W := (v_0, v_1, \ldots, v_t)$ of length $t$ on a graph $G = (V, E)$, where the random variables $v_i \in V$ denote the vertices visited by the walk, and form a Markov chain.

**Definition 4.5.** *We distinguish the following quantities related to the walk $W$:*

- *For a vertex $v \in V$ (resp., edge $e \in E$), denote by $\pi_v$ (resp., $\pi_e$) the probability that $W$ passes $v$ (resp., $e$).*

---

[3] Note that the mixing time highly depends on the underlying distance by which the distance between two distributions is quantified. In particular, we are slightly deviating from the more standard definition which is with respect to the variation ($\ell_1$) distance (see, e.g., [MU05, Definition 11.2]).

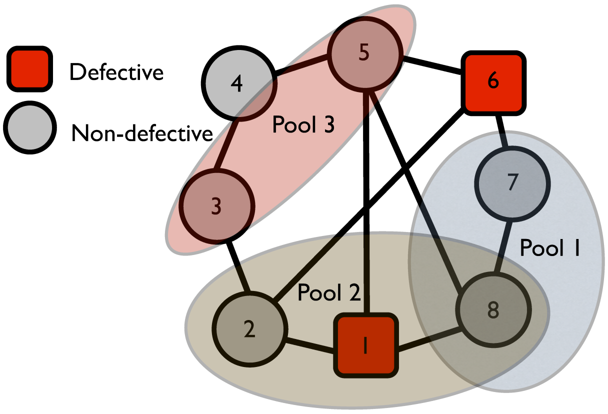


Figure 4.4 – The result of pool 1 is negative because it does not contains a defective item, whereas the result of pool 2 is positive because it contains a defective item. Pool 3 is not consistent with the graph and thus not allowed because the items are not connected by a path.

- *For a vertex* $v \in V$ *(resp., edge* $e \in E$*) and subset* $A \subseteq V$, $v \notin A$ *(resp.,* $B \subseteq E$, $e \notin B$*), denote by* $\pi_{v,A}$ *(resp.,* $\pi_{e,B}$*) the probability that* $W$ *passes* $v$ *but none of the vertices in* $A$ *(resp., passes* $e$ *but none of the edges in* $B$*).*

Note that these quantities are determined by not only $v, e, A, B$ (indicated as subscripts) but they also depend on the choice of the underlying graph, the distribution of the initial vertex $v_0$ and length of the walk $t$. However, we find it convenient to keep the latter parameters implicit when their choice is clear from the context.

In the previous definition, the length of the random walk was taken as a fixed parameter $t$. Another type of random walks that we consider in this work have their end points as a parameter and do not have an a priori fixed length. In the following, we define similar probabilities related to the latter type of random walks.

Consider a particular random walk $W := (v_0, v_1, \ldots, u)$ on a graph $G = (V, E)$ that continues until it reaches a fixed vertex $u \in V$.

**Definition 4.6.** *We distinguish the following quantities related to* $W$*: For a vertex* $v \in V$ *(resp., edge* $e \in E$*) and subset* $A \subseteq V$, $v \notin A$ *(resp.,* $B \subseteq E$, $e \notin B$*), denote by* $\pi^{(u)}_{v,A}$ *(resp.,* $\pi^{(u)}_{e,B}$*) the probability that* $W$ *passes* $v$ *but none of the vertices in* $A$ *(resp., passes* $e$ *but none of the edges in* $B$*).*

Again these quantities depend on the choice of $G$ and the distribution of $v_0$ that we will keep implicit.

## 4.3 Problem Statement

Consider a given graph $G = (V, E)$ in which at most $d$ vertices (resp., edges) are defective. The goal is to characterize the set of defective items using a number of measurements that is as small as possible, where each measurement determines whether the set of vertices (resp., edges) observed along a path on the graph has a non-empty intersection with the defective set. We call the problem of finding defective vertices *vertex group testing* and that of finding defective edges *edge group testing*.

As mentioned earlier, not all sets of vertices can be grouped together, and only those that share a path on the underlying graph $G$ can participate in a pool (see Fig. 4.4).

## 4.4 Main Results

In the following, we introduce four random constructions (designs) for both problems. The proposed designs follow the natural idea of determining pools by taking random walks on the graph.

**Design 1.**

*Given:* a constraint graph $G = (V, E)$ with $r \geq 0$ designated vertices $s_1, \dots, s_r \in V$, and integer parameters $m$ and $t$.

*Output:* an $m \times |V|$ boolean matrix $\boldsymbol{M}$.

*Construction:* Construct each row of $\boldsymbol{M}$ independently as follows: Let $v \in V$ be any of the designated vertices $s_i$, or otherwise a vertex chosen uniformly at random from $V$. Perform a random walk of length $t$ starting from $v$, and let the corresponding row of $\boldsymbol{M}$ be the characteristic vector of the set of vertices visited by the walk.

**Design 2.**

*Given:* a constraint graph $G = (V, E)$ and integer parameters $m$ and $t$.

*Output:* an $m \times |E|$ boolean matrix $\boldsymbol{M}$.

*Construction:* Construct each row of $\boldsymbol{M}$ independently as follows: Let $v \in V$ be any arbitrary vertex of $G$. Perform a random walk of length $t$ starting from $v$, and let the corresponding row of $\boldsymbol{M}$ be the characteristic vector of the set of edges visited by the walk.

**Design 3.**

*Given:* a constraint graph $G = (V, E)$ with $r \geq 0$ designated vertices $s_1, \dots, s_r \in V$, a *sink node* $u \in V$, and integer parameter $m$.

*Output:* an $m \times |V|$ boolean matrix $\boldsymbol{M}$.

*Constructions:* Construct each row of $\boldsymbol{M}$ independently as follows: Let $v \in V$ be any of the designated vertices $s_i$, or otherwise a vertex chosen uniformly at random from $V$. Perform a random walk starting from $v$ until we reach $u$, and let the corresponding row of $\boldsymbol{M}$ be the characteristic vector of the set of vertices visited by the walk.

**Design 4.**

*Given:* a constraint graph $G = (V, E)$, a *sink node* $u \in V$, and integer parameter $m$.

*Output:* an $m \times |E|$ boolean matrix $\boldsymbol{M}$.

*Construction:* Construct each row of $\boldsymbol{M}$ independently as follows: Let $v \in V$ be any arbitrary vertex of $G$. Perform a random walk, starting from $v$ until we reach $u$, and let the corresponding row of $\boldsymbol{M}$ be the characteristic vector of the set of edges visited by the walk.

By construction, Designs 1 and 3 (resp., Designs 2 and 4) output boolean matrices that are vertex- (resp., edge-) consistent with the graph $G$. Our main goal is to show that, when the number of rows $m$ is sufficiently large, the output matrices become $d$-disjunct (for a given parameter $d$) with overwhelming probability.

Designs 1 and 3 in particular provide two choices for constructing the measurement matrix $\boldsymbol{M}$. Namely, the start vertices can be chosen within a fixed set of designated vertices, or, chosen randomly among all vertices of the graph. As we will see later, in theory there is no significant difference between the two schemes. However, for some applications it might be the case that only a small subset of vertices are accessible as the starting points (e.g., in network tomography such a subset can be determined by the vantage points), and this can be modeled by an appropriate choice of the designated vertices in Designs 1 and 3.

The following theorem states the main result of this work, showing that our proposed designs indeed produce disjunct matrices that can be used for the purpose of graph-constrained group testing. We will state both noiseless results (corresponding to $d$-disjunct matrices), and noisy ones (corresponding to $(d, e)$-disjunct ones, where the noise tolerance $e$ depends on a fixed "noise parameter" $\eta \in [0, 1)$). The proof of the following theorem is given in Section 4.5.1.

**Theorem 4.7.** *Let $\eta \geq 0$ be a fixed parameter, and suppose that $G = (V, E)$ is a $(D, c)$-uniform graph on $n$ vertices with $\delta$-mixing time $T(n)$ (where $\delta := (1/2cn)^2$). Then there exist parameters with asymptotic values given in Table 4.1 such that, provided that $D \geq D_0$,*

1. *Design 1 with the path length $t := t_1$ and the number of measurements $m := m_1$ outputs a matrix $\boldsymbol{M}$ that is vertex-consistent with $G$. Moreover, once the columns of $\boldsymbol{M}$ corresponding to the designated vertices $s_1, \ldots, s_r$ are removed, the matrix becomes $d$-disjunct with probability $1 - o(1)$. More generally, for $m := m_1'$ the matrix becomes $(d, e_1)$-disjunct with probability $1 - o(1)$.*
2. *Design 2 with path length $t := t_2$ and $m := m_2$ measurements outputs a matrix $\boldsymbol{M}$ that is edge-consistent with $G$ and is $d$-disjunct with probability $1 - o(1)$. More generally, for $m := m_2'$ the matrix becomes $(d, e_2)$-disjunct with probability $1 - o(1)$.*
3. *Design 3 with the number of measurements $m := m_3$ outputs a matrix $\boldsymbol{M}$ that is vertex-consistent with $G$. Moreover, once the columns of $\boldsymbol{M}$ corresponding to the designated vertices $s_1, \ldots, s_r$ and the sink node $u$ are removed, the matrix becomes $d$-disjunct with probability $1 - o(1)$. More generally, for $m := m_3'$ the matrix becomes $(d, e_3)$-disjunct with probability $1 - o(1)$.*
4. *Design 4 with the number of measurements $m := m_4$ outputs a matrix $\boldsymbol{M}$ that is edge-consistent with $G$ and is $d$-disjunct with probability $1 - o(1)$. More generally, for $m := m_4'$ the matrix becomes $(d, e_4)$-disjunct with probability $1 - o(1)$.*

In Designs 1 and 3, we need to assume that the designated vertices (if any) are not defective, and hence, their corresponding columns can be removed from the matrix $\boldsymbol{M}$. By doing so, we will be able to ensure that the resulting matrix is disjunct. Obviously, such a restriction cannot be avoided since, for example, $\boldsymbol{M}$ might be forced to contain an all-ones column corresponding to one of the designated vertices and thus, fail to be even 1-disjunct.

By applying Theorem 4.7 on the complete graph (using Design 1), we get $O(d^2 \log(n/d))$ measurements, since in this case, the mixing time is $T(n) = 1$ and also $c = 1$. Thereby, we

Table 4.1 – The asymptotic values of various parameters in Theorem 4.7.

| Parameter | Value |
|---|---|
| $D_0$ | $O(c^2 d T^2(n))$ |
| $m_1, m_2$ | $O(c^4 d^2 T^2(n) \log(n/d))$ |
| $m_3$ | $O(c^8 d^3 T^4(n) \log(n/d))$ |
| $m_4$ | $O(c^9 d^3 D T^4(n) \log(n/d))$ |
| $t_1$ | $O(n/(c^3 d T(n)))$ |
| $t_2$ | $O(nD/(c^3 d T(n)))$ |
| $e_1, e_2, e_3, e_4$ | $\Omega(\eta d \log(n/d)/(1-\eta)^2)$ |
| $m'_i, i \in [4]$ | $O(m_i/(1-\eta)^2)$ |

recover the trade-off obtained by the probabilistic construction in classical group testing (note that classical group testing corresponds to graph-constrained group testing on the vertices of the complete graph).

We will show in Section 4.5.2 that, for our specific choice of $\delta := (1/2cn)^2$, the $\delta$-mixing time of an Erdős-Rényi random graph $G(n,p)$ is (with overwhelming probability) $T(n) = O(\log n)$. This bound more generally holds for any graph with conductance $\Omega(1)$, and in particular, expander graphs with constant spectral gap. Thus we have the following result (with a summary of the achieved parameters given in Table 4.2).

**Theorem 4.8.** *There is an integer $D_0 = \Omega(d \log^2 n)$ such that for every $D \geq D_0$ the following holds: Suppose that the graph $G$ is either*

1. *A $D$-regular expander graph with normalized second largest eigenvalue (in absolute value) $\lambda$ that is bounded away from 1; i.e., $\lambda = 1 - \Omega(1)$, or,*
2. *An Erdős-Rényi random graph $G(n, D/n)$.*

*Then for every $\eta \in [0,1)$, with probability $1 - o(1)$ Designs 1, 2, 3, and 4 output $(d,e)$-disjunct matrices (not considering the columns corresponding to the designated vertices and the sink in Designs 1 and 3), for some $e = \Omega(\eta d \log n)$, using respectively $m_1, m_2, m_3, m_4$ measurements, where*

$$\begin{aligned} m_1 &= O(d^2(\log^3 n)/(1-\eta)^2), \\ m_2 &= O(d^2(\log^3 n)/(1-\eta)^2), \\ m_3 &= O(d^3(\log^5 n)/(1-\eta)^2), \\ m_4 &= O(d^3 D(\log^5 n)/(1-\eta)^2). \end{aligned}$$

**The fixed-input case.** Recall that, as Theorem 4.7 shows, our proposed designs almost surely produce disjunct matrices using a number of measurements summarized in Table 4.1. Thus, with overwhelming probability, once we fix the resulting matrix, it has the combinatorial property of distinguishing between *any* two $d$-sparse boolean vectors (each corresponding to a set of up to $d$ defective vertices, not including designated ones, for Designs 1 and 3, or up to $d$ defective edges for Designs 2 and 4) in the *worst case*. However, the randomized nature of our designs can be used to our benefit to show that, practically, one can get similar results with a number of measurements that is almost by

Table 4.2 – The asymptotic values of the bounds achieved by Theorem 4.8.

| Parameter | Value |
|---|---|
| $D_0$ | $O(d \log^2 n)$ |
| $m_1, m_2$ | $O(d^2 (\log^3 n)/(1-p)^2)$ |
| $m_3$ | $O(d^3 (\log^5 n)/(1-p)^2)$ |
| $m_4$ | $O(d^3 D(\log^5 n)/(1-p)^2)$ |
| $e$ | $\Omega(\eta d \log n)$ |

a factor $d$ smaller than what required by Theorem 4.7. Of course, assuming a substantially lower number of measurements, we should not expect to obtain disjunct matrices, or equivalently, to be able to distinguish between any two sparse vectors in the worst case. However, it can be shown that, for every *fixed* $d$-sparse vector $x$, the resulting matrix with overwhelming probability will be able to distinguish between $x$ and any other $d$-sparse vector using a lower number of measurements. In particular, with overwhelming probability (over the choice of the measurements), from the measurement outcomes obtained from $x$, it will be possible to uniquely reconstruct $x$. More precisely, it is possible to show the following theorem, as proved in Section 4.5.1.

**Theorem 4.9.** *Consider the assumptions of Theorem 4.7, and let* $\gamma := (\log n)/(d \log(n/d))$. *Consider any fixed set of up to d vertices* $S \subseteq V$ *such that* $|S| \leq d$ *and* $S \cap \{s_1, \ldots, s_r\} = \emptyset$ *and any fixed set of up to d edges* $T \subseteq E$, $|T| \leq d$. *Then with probability* $1 - o(1)$ *over the randomness of the designs the following holds.*

*Let* $M_1, \ldots M_4$ *respectively denote the measurement matrices produced by Designs* $1, \ldots, 4$ *with the number of rows set to* $O(\gamma m'_i)$. *Then for every* $S' \subseteq V$ *and every* $T' \subseteq E$ *such that* $S' \neq S$, $T' \neq T$ *and* $|S'| \leq d$, $S' \cap \{s_1, \ldots, s_r\} = \emptyset$, $|T'| \leq d$, *we have that*

1. *The measurement outcomes of* $M_1$ *on* $S$ *and* $S'$ *(resp.,* $M_3$ *on* $S$ *and* $S'$*) differ at more than* $\Omega(\gamma e_1)$ *(resp.,* $\Omega(\gamma e_3)$*) positions.*
2. *The measurement outcomes of* $M_2$ *on* $T$ *and* $T'$ *(resp.,* $M_4$ *on* $T$ *and* $T'$*) differ at more than* $\Omega(\gamma e_2)$ *(resp.,* $\Omega(\gamma e_4)$*) positions.*

A direct implication of this result is that (with overwhelming probability), once we fix the matrices obtained from our randomized designs with the lowered number of measurements (namely, having $O(\gamma m'_i) \approx O(m'_i/d)$ rows), the fixed matrices will be able to distinguish between *almost all* pairs of $d$-sparse vectors (and in particular, uniquely identify randomly drawn $d$-sparse vectors, with probability $1 - o(1)$ over their distribution).

**Example in Network Tomography.** Here we illustrate a simple concrete example that demonstrates how our constructions can be used for network tomography in a simplified model. Suppose that a network (with known topology) is modeled by a graph with nodes representing routers and edges representing links that connect them, and it is suspected that at most $d$ links in the network are congested (and thus, packets routed through them are dropped). Assume that, at a particular "source node" $s$, we wish to identify the set of congested links by distributing packets that originate from $s$ in the network.

First, $s$ generates a packet containing a time stamp $t$ and sends it to a randomly chosen neighbor, who in turn, decrements the time stamp and forwards the packet to a randomly chosen neighbor, etc. The process continues until the time stamp reaches zero, at which point the packet is sent back to $s$ along the same path it has traversed. This can be achieved by storing the route to be followed (which is randomly chosen at $s$) in the packet. Alternatively, for practical purposes, instead of storing the whole route in the packet, $s$ can generate and store a random seed for a pseudorandom generator as a header in the packet. Then each intermediate router can use the specified seed to determine one of its neighbors to which the packet has to be forwarded.

Using the procedure sketched above, the source node generates a number of independent packets, which are distributed in the network. Each packet is either returned back to $s$ in a timely manner, or, eventually does not reach $s$ due to the presence of a congested link within the route. By choosing an appropriate timeout, $s$ can determine the packets that are routed through the congested links.

The particular scheme sketched above implements our Design 2, and thus Theorem 4.7 implies that, by choosing the number of hops $t$ appropriately, after generating a sufficient number of packets (that can be substantially smaller than the size of the network), $s$ can determine the exact set of congested links. This result holds even if a number of the measurements produce false outcomes (e.g., a congested link may nevertheless manage to forward a packet, or a packet may be dropped for reasons other than congestion), in which case by estimating an appropriate value for the *noise parameter* $p$ in Theorem 4.7 and increasing the number of measurements accordingly, the source can still correctly distinguish the congested links. Of course one can consider different schemes for routing the test packets. For example, it may be more desirable to forward the packets until they reach a pre-determined "sink node", an approach that is modeled by our Designs 3 and 4 above.

## 4.5 Analysis

This section contains the proofs of our main theorems.

### 4.5.1 Proof of Theorems 4.7 and 4.9

Before discussing Theorem 4.7 and its proof, we introduce some basic propositions that are later used in the proof. The omitted proofs will be presented in the appendix. Throughout this section, we consider an underlying graph $G = (V, E)$ that is $(D, c)$-uniform, with mixing time $T(n)$.

**Proposition 4.10.** *Let $A, B_1, B_2, \ldots, B_n$ be events on a finite probability space, define $B := \cup_{i=1}^{n} B_i$, and suppose that:*

1. *For every $i \in [n]$, $\Pr[A \mid B_i] \leq \epsilon$.*
2. *For every set $S \subseteq [n]$ with $|S| > k$, $\cap_{i \in S} B_i = \emptyset$.*

*Then, $\Pr[A \mid B] \leq \epsilon k$.*

The proof of this proposition may be found in Section 4.5.5. The following proposition is a corollary of a well-known result for the stationary distribution of irregular graphs [MU05, Theorem 7.13]. A formal proof of this proposition is given in Section 4.5.6.

**Proposition 4.11.** *Let $G = (V, E)$ be a $(D, c)$-uniform graph, and denote by $\mu$ the stationary distribution of $G$ (assuming that $G$ is not bipartite). Then for each $v \in V$, $1/cn \leq \mu(v) \leq c/n$.*

**Proposition 4.12.** *For the quantities $\pi_v$ and $\pi_e$ in Definition 4.5, we have*

$$\pi_v = \Omega\left(\frac{t}{cnT(n)}\right), \quad \pi_e = \Omega\left(\frac{t}{cDnT(n)}\right).$$

The proof of this Proposition 4.12 is presented in Section 4.5.7. In fact, a stronger statement than this proposition can be obtained, that with noticeable probability, every fixed vertex (or edge) is hit by the walk at least once but not too many times nor too "early". This is made more precise in the following two propositions, which are proved in Sections 4.5.8 and 4.5.9, respectively.

**Proposition 4.13.** *Consider any walk $W$ in Design 1 (resp., Design 2). There is a $k = O(c^2T(n))$ such that, for every $v \in V$ and every $e \in E$, the probability that $W$ passes $v$ (resp., $e$) more than $k$ times is at most $\pi_v/4$ (resp., $\pi_e/4$).*

**Proposition 4.14.** *For any random walk $W$ in Design 1, let $v \in V$ be any vertex that is not among the designated vertices $s_1, \ldots, s_r$. Then the probability that $W$ visits $v$ within the first $k$ steps is at most $k/D$.*

The following proposition shows that the distributions of two vertices on a random walk that are far apart by a sufficiently large number of steps are almost independent. The proof of this proposition may be found in Section 4.5.10.

**Proposition 4.15.** *Consider a random walk $w := (v_0, v_1, \ldots, v_t)$ on $G$ starting from an arbitrary vertex, and suppose that $j \geq i + T(n)$. Let $\mathcal{E}$ denote any event that only depends on the first $i$ vertices visited by the walk. Then for every $u, v \in V$,*

$$|\Pr[v_i = u | v_j = v, \mathcal{E}] - \Pr[v_i = u | \mathcal{E}]| \leq 2/(3cn).$$

The following lemmas, which form the technical core of this work, lower bound the quantities $\pi_{v,A}$, $\pi_{e,B}$, $\pi_{v,A}^{(u)}$, $\pi_{e,B}^{(u)}$ as defined by Definitions 4.5 and 4.6.

**Lemma 4.16.** *There is a $D_0 = O(c^2dT^2(n))$ and $t_1 = O(n/(c^3dT(n)))$ such that whenever $D \geq D_0$, by setting the path lengths $t := t_1$ in Design 1 the following holds. Let $v \in V$, and $A \subseteq V$ be a set of at most $d$ vertices in $G$ such that $v \notin A$ and $A \cup \{v\}$ does not include any of the designated vertices $s_1, \ldots s_r$. Then*

$$\pi_{v,A} = \Omega\left(\frac{1}{c^4dT^2(n)}\right). \tag{4.1}$$

*Proof.* Denote by $\mu$ the stationary distribution of $G$. We know from Proposition 4.11 that for each $u \in V$, $1/cn \leq \mu_u \leq c/n$.

Let $k = O(c^2T(n))$ be the quantity given by Proposition 4.13, $\mathcal{B}$ denote the *bad event* that $W$ hits some vertex in $A$. Moreover, let $\mathcal{G}$ denote the *good event* that $W$ hits $v$ no more than $k$ times in total and never within the first $2T(n)$ steps. The probability of $\mathcal{G}$ is, by Propositions 4.13 and 4.14, at least

$$\Pr(\mathcal{G}) \geq 1 - 2T(n)/D - O(t/cnT(n)),$$

which can be made arbitrarily close to 1 (say larger than 0.99) by choosing $D$ sufficiently large and $t$ sufficiently small (as required by the statement). Now,

$$\begin{aligned}\pi_{v,A} &= \Pr[\neg\mathcal{B}, v \in W] \\ &\geq \Pr[\neg\mathcal{B}, v \in W, \mathcal{G}] \\ &= \Pr[v \in W, \mathcal{G}](1 - \Pr[\mathcal{B} \mid v \in W, \mathcal{G}]). \end{aligned} \tag{4.2}$$

By taking $D$ large enough, and in particular, $D = \Omega(c^2 d T^2(n))$, we can ensure that

$$2T(n)/D \leq \pi_v/4.$$

Combined with Proposition 4.13, we have $\Pr[v \in W, \mathcal{G}] \geq \pi_v/2$, since

$$\begin{aligned}\Pr[v \in W, \mathcal{G}] &= \Pr[v \in W] + \Pr[\mathcal{G}] - \Pr[(v \in W) \cup \mathcal{G}] \\ &\geq \pi_v + (1 - \pi_v/2) - 1 = \pi_v/2.\end{aligned}$$

Thus, (4.2) gives

$$\pi_{v,A} \geq \pi_v(1 - \Pr[\mathcal{B} \mid v \in W, \mathcal{G}])/2. \tag{4.3}$$

Now we need to upperbound $\pi := \Pr[\mathcal{B} \mid v \in W, \mathcal{G}]$. Before doing so, fix some $i > 2T(n)$, and assume that $v_i = v$. Moreover, fix some vertex $u \notin A$ and assume that $v_0 = u$. We first try to upperbound $\Pr[\mathcal{B} \mid v_i = v, v_0 = u]$.

Let $\ell := i - T(n)$ and $\rho := i + T(n)$, and for the moment, assume that $T(n) + 1 < \ell < \rho < t$ (a "degenerate" situation occurs when this is not the case). Partition $W$ into four parts:

$$\begin{aligned} W_1 &:= (v_0, v_1, \ldots, v_{T(n)}), \\ W_2 &:= (v_{T(n)+1}, v_{T(n)+2}, \ldots, v_{\ell-1}), \\ W_3 &:= (v_\ell, v_{\ell+1}, \ldots, v_\rho), \\ W_4 &:= (v_{\rho+1}, v_{\rho+2}, \ldots, v_t). \end{aligned}$$

For $j = 1, 2, 3, 4$, define

$$\pi_j := \Pr[W_j \text{ enters } A \mid v_i = v, v_0 = u].$$

Now we upperbound each of the $\pi_j$. In a degenerate situation, some of the $W_i$ may be empty, and the corresponding $\pi_j$ will be zero.

Each of the sub-walks $W_2$ and $W_4$ are "oblivious" to the conditioning on $v_i$ and $v_0$ (because they are sufficiently far from both and Proposition 4.15 applies). In particular, the distribution of each vertex on $W_4$ is point-wise close to $\mu$. Therefore, under our conditioning the probability that each such vertex belongs to $A$ is at most $|A|(c/n+\delta) < 2dc/n$. The argument on $W_2$ is similar, but more care is needed. Without the conditioning on $v_i$, each vertex on $W_2$ has an almost-stationary distribution. Moreover, by Proposition 4.15, the conditioning on $v_2$ changes this distribution by up to $\delta' := 2/(3cn) < 1/n$ at each point. Altogether, for each $j \in \{T(n)+1, \ldots, \ell - 1\}$, we have

$$\begin{aligned}\Pr[v_j \in A \mid v_i = v, v_0 = u] &\leq |A|(c/n + \delta + \delta') \\ &\leq 2dc/n.\end{aligned}$$

Using a union bound on the number of steps, we conclude that $\pi_2 + \pi_4 \leq 2dct/n$.

In order to bound $\pi_3$, we observe that of all $D$ or more neighbors of $v_i$, at most $d$ can lie on $A$. Therefore,

$$\Pr[v_{i+1} \in A \mid v_i = v, v_0 = u] \leq d/D.$$

Similarly,

$$\Pr[v_{i+2} \in A \mid v_i = v, v_0 = u, v_{i+1}] \leq d/D,$$

regardless of $v_{i+1}$ which means

$$\Pr[v_{i+2} \in A \mid v_i = v, v_0 = u] \leq d/D,$$

and in general,

$$(\forall j = i+1, \ldots, \rho), \quad \Pr[v_j \in A \mid v_i = v, v_0 = u] \leq d/D. \tag{4.4}$$

Similarly we have,

$$\Pr[v_{i-1} \in A \mid v_i = v] \leq d/D,$$

and by Proposition 4.15 (and time-reversibility), conditioning on $v_0$ changes this probability by at most $d\delta'$. Therefore,

$$\Pr[v_{i-1} \in A \mid v_i = v, v_0 = u] \leq d/D + d\delta',$$

and in general,

$$(\forall j = \ell, \ldots, i-1), \qquad \Pr[v_j \in A \mid v_i = v, v_0 = u] \leq d/D + d\delta'. \tag{4.5}$$

Altogether, using a union bound and by combining (4.4) and (4.5), we get that

$$\pi_3 \leq 2dT(n)/D + dT(n)/n \leq 3dT(n)/D.$$

Using the same reasoning, $\pi_1$ can be bounded as

$$\pi_1 \leq dT(n)/D + dT(n)/n \leq 2dT(n)/D.$$

Finally, we obtain

$$\begin{aligned} \Pr[\mathcal{B} \mid v_i = v, v_0 = u] &\leq \pi_1 + \pi_2 + \pi_3 + \pi_4 \\ &\leq \frac{5dT(n)}{D} + \frac{2dct}{n}. \end{aligned} \tag{4.6}$$

Our next step is to relax the conditioning on the starting point of the walk. The probability that the initial vertex is in $A$ is at most $d/n$ (as this happens only when the initial vertex is taken randomly), and by Proposition 4.15, conditioning on $v_i$ changes this probability by at most $d\delta' < d/n$. Now we write

$$\begin{aligned} \Pr[\mathcal{B} \mid v_i = v] &\leq \Pr[v_0 \in A] + \Pr[\mathcal{B} \mid v_i = v, v_0 \notin A] \\ &\leq \Pr[v_0 \in A] + \pi_1 + \pi_2 + \pi_3 + \pi_4 \\ &\leq \frac{5dT(n)}{D} + \frac{4dct}{n}, \end{aligned}$$

where we have used the chain rule in the first inequality, and Proposition 4.10 with $k = 1$ for the second one. Now, since $\Pr[\mathcal{G}]$ is very close to 1, conditioning on this event does not increase probabilities by much (say no more than a factor 1.1). Therefore,

$$\Pr[\mathcal{B} \mid v_i = v, \mathcal{G}] \leq 1.1 \left( \frac{5dT(n)}{D} + \frac{4dct}{n} \right).$$

Now in the probability space conditioned on $\mathcal{G}$, define events $\mathcal{G}_i$, $i = 2T(n)+1, \dots t$, where $\mathcal{G}_i$ is the event that $v_i = v$. Note that the intersection of more than $k$ of the $\mathcal{G}_i$ is empty (as conditioning on $\mathcal{G}$ implies that the walk never passes $v$ more than $k$ times), and moreover, the union of these is the event that the walk passes $v$. Now we apply Proposition 4.10 to conclude that

$$\begin{aligned} \Pr[\mathcal{B} \mid v \in W, \mathcal{G}] &\leq 1.1k \left( \frac{5dT(n)}{D} + \frac{4dct}{n} \right) \\ &= O\left( c^2 T(n) \left( \frac{5dT(n)}{D} + \frac{4dct}{n} \right) \right). \end{aligned}$$

By taking $D = \Omega(c^2 d T^2(n))$ and $t = O(n/c^3 d T(n))$ we can make the right hand side arbitrarily small (say at most 1/2). Now we get back to (4.3) to conclude, using Proposition 4.12, that

$$\pi_{v,A} \geq \pi_v/4 = \Omega\left( \frac{t}{cnT(n)} \right) = \Omega\left( \frac{1}{c^4 d T^2(n)} \right).$$

□

Similarly, we can bound the edge-related probability $\pi_{e,B}$ as in the following lemma. The proof of the lemma is very similar to that of Lemma 4.16, and is therefore skipped for brevity.

**Lemma 4.17.** *There is a* $D_0 = O(c^2 d T^2(n))$ *and* $t_2 = O(nD/c^3 d T(n))$ *such that whenever* $D \geq D_0$*, by setting the path lengths* $t := t_2$ *in Design 2 the following holds. Let* $B \subseteq E$ *be a set of at most* $d$ *edges in* $G$*, and* $e \in E$*,* $e \notin B$*. Then*

$$\pi_{e,B} = \Omega\left( \frac{1}{c^4 d T^2(n)} \right). \tag{4.7}$$

In Designs 3 and 4, the quantities $\pi^{(u)}_{v,A}$ and $\pi^{(u)}_{e,B}$ defined in Definition 4.6 play a similar role as $\pi_{v,A}$ and $\pi_{e,B}$. In order to prove disjunctness of the matrices obtained in Designs 3 and 4, we will need lower bounds on $\pi^{(u)}_{v,A}$ and $\pi^{(u)}_{e,B}$ as well. In the following we show the desired lower bounds.

**Lemma 4.18.** *There is a* $D_0 = O(c^2 d T^2(n))$ *such that whenever* $D \geq D_0$*, in Design 3 the following holds. Let* $v \in V$*, and* $A \subseteq V$ *be a set of at most* $d$ *vertices in* $G$ *such that* $v \notin A$ *and* $A \cup \{v\}$ *is disjoint from* $\{s_1, \dots s_r, u\}$*. Then*

$$\pi^{(u)}_{v,A} = \Omega\left( \frac{1}{c^8 d^2 T^4(n)} \right). \tag{4.8}$$

*Proof.* Let $D_0$ and $t_1$ be quantities given by Lemma[4] 4.16. Let $w_0$ denote the start vertex of a walk performed in Design 3, and consider an infinite walk $W = (v_0, v_1, v_2, \ldots)$ that starts from a vertex identically distributed with $w_0$. Let the random variables $i, j, k$ respectively denote the times that $W$ visits $v, u$, and any of the vertices in $A$ for the first time. Therefore, $v_i = v$, $v_j = u$, and $v_k \in A$, $v_t \neq v$ for every $t < i$ and so on. Then the quantity $\pi^{(u)}_{v,A}$ that we wish to bound corresponds to the probability that $i < j < k$, that is, probability of the event that in $W$, the first visit of $v$ occurs before the walk reaches the sink node $u$ for the first time, and moreover, the walk never hits $A$ before reaching $u$. Observe that this event in particular contains the sub-event that $i \leq t_1$, $t_1 < j \leq 2t_1$, and $k > 2t_1$, where $t_1$ is picked as in Lemma 4.16. Denote by $\mathcal{W} \subseteq V^{t_1+1}$ the set of all sequences of $t_1 + 1$ vertices of $G$ (i.e., walks of length $t_1$) that include $v$ but not any of the vertices in $A \cup \{u\}$. Now, we can write

$$\begin{aligned}
\pi^{(u)}_{v,A} &= \Pr[i < j < k] & (4.9)\\
&\geq \Pr[i \leq t_1 < j \leq 2t_1 < k] \\
&= \Pr[(i \leq t_1) \wedge (j > t_1) \wedge (k > t_1)] \cdot \\
&\quad \Pr[t_1 < j \leq 2t_1 < k \mid \\
&\qquad (i \leq t_1) \wedge (j > t_1) \wedge (k > t_1)] \\
&= \Pr[(v_0, \ldots, v_{t_1}) \in \mathcal{W}] \cdot \\
&\quad \Pr[t_1 < j \leq 2t_1 < k \mid (v_0, \ldots, v_{t_1}) \in \mathcal{W}] & (4.10)
\end{aligned}$$

The probability $\Pr[(v_0, \ldots, v_{t_1}) \in \mathcal{W}]$ is exactly $\pi_{v,A\cup\{u\}}$ with respect to the start vertex $w_0$. Therefore, Lemma 4.16 gives the lower bound

$$\Pr[(v_0, \ldots, v_{t_1}) \in \mathcal{W}] = \Omega\left(\frac{1}{c^4 d T^2(n)}\right).$$

Furthermore observe that, regardless of the outcome $(v_0, \ldots, v_{t_1}) \in \mathcal{W}$, we have

$$\Pr[t_1 < j \leq 2t_1 < k \mid v_0, \ldots, v_{t_1}] = \pi_{u,A}$$

where $\pi_{u,A}$ is taken with respect to the start vertex $v_{t_1}$. Therefore, since $v_{t_1} \notin A \cup \{u\}$, again we can use Lemma 4.16 to conclude that

$$\Pr[t_1 < j \leq 2t_1 < k \mid (v_0, \ldots, v_{t_1}) \in \mathcal{W}] = \Omega\left(\frac{1}{c^4 d T^2(n)}\right).$$

By plugging the bounds in (4.10) the claim follows. □

A similar result can be obtained for Design 4 on the edges. Since the arguments are very similar, we only sketch a proof.

**Lemma 4.19.** *There is a $D_0 = O(c^2 d T^2(n))$ such that whenever $D \geq D_0$, in Design 4 the following holds. Let $B \subseteq E$ be a set of at most $d$ edges in $G$, and $e \in E$, $e \notin B$. Then*

$$\pi^{(u)}_{e,B} = \Omega\left(\frac{1}{c^9 d^2 D T^4(n)}\right). \tag{4.11}$$

---

[4] In fact, as will be clear by the end of the proof, Lemma 4.16 should be applied with the sparsity parameter $d + 1$ instead of $d$. However, this will only affect constant factors that we ignore.

*Proof. (sketch)* Similar to the proof of Lemma 4.18, we consider an infinite continuation $W = (v_0, v_1, \ldots)$ of a walk performed in Design 4 and focus on its first $t_1 + t_2$ steps, where $t_1$ and $t_2$ are respectively the time parameters given by Lemmas 4.16 and 4.17. Let

$$W_1 := (v_0, \ldots, v_{t_1}),$$
$$W_2 := (v_{t_1+1}, \ldots, v_{t_1+t_2}).$$

Again following the argument of Lemma 4.18, we lower bound $\pi^{(u)}_{e,B}$ by the probability of a sub-event consisting the intersection of the following two events:

1. The event $\mathcal{E}_1$ that $W_1$ visits $e$ but neither the sink node $u$ nor any of the edges in $B$, and
2. The event $\mathcal{E}_2$ that $W_2$ visits the sink node $u$ but none of the edges in $B$.

Consider the set $A \subseteq V$ consisting of the endpoints of the edges in $B$ and denote by $v \in V$ any of the endpoints of $e$. Let $p := \pi_{v,A}$ (with respect to the start vertex $v_0$). Now, $\Pr[\mathcal{E}_1] \geq p/(cD)$ since upon visiting $v$, there is a $1/\deg(v)$ chance that the next edge taken by the walk turns out to be $e$. The quantity $p$ in turn, can be lower bounded using Lemma 4.16. Moreover, regardless of the outcome of $W_1$, the probability that $W_2$ visits $u$ but not $B$ (and subsequently, the conditional probability $\Pr[\mathcal{E}_2 \mid \mathcal{E}_1]$) is at least the probability $\pi_{e',B}$ (with respect to the start vertex $v_{t_1}$), where $e' \in E$ can be taken as any edge incident to the sink node $u$. This latter quantity can be lower bounded using Lemma 4.17. Altogether, we obtain the desired lower bound on $\pi^{(u)}_{e,B}$. ☐

It is natural to ask whether the exponent of $d^2$ in the denominator of the lower bound in Lemma 4.18 can be improved. We argue that this is not the case in general, by considering the basic example where the underlying graph is the complete graph $K_n$ and each walk is performed starting from a random node. Consider an infinite walk $W$ starting at a random vertex and moreover, the set of $d + 2$ vertices $A' := A \cup \{u, v\}$. Due to the symmetry of the complete graph, we expect that the order at which $W$ visits the vertices of $A'$ for the first time is uniformly distributed among the $(d+2)!$ possible orderings of the elements of $A'$. However, in the event corresponding to $\pi^{(u)}_{v,A}$, we are interested in seeing $v$ first, then $u$, and finally the elements of $A$ in some order. Therefore, for the case of complete graph we know that $\pi^{(u)}_{v,A} = O(1/d^2)$, and thus, the quadratic dependence on $d$ is necessary even for very simple examples.

Another question concerns the dependence of the lower bound in Lemma 4.19 on the degree parameter $D$. Likewise above, an argument for the case of complete graph suggests that in general this dependence cannot be eliminated. For edge group testing on the complete graph, we expect to see a uniform distribution on the ordering at which we visit a particular set of edges in the graph. Now the set of edges of our interest consists of the union of the set $B \cup \{e\}$ and all the $n - 1$ edges incident to the sink node $u$, and is thus of size $n + d$. The orderings that contribute to $\pi^{(u)}_{e,B}$ must have $e$ as the first edge and an edge incident to $u$ as the second edge. Therefore we get that, for the case of complete graph,

$$\pi^{(u)}_{e,B} = O(1/n) = O(1/D),$$

which exhibits a dependence on the degree in the denominator. Now, we are ready to prove our main theorem.

*Proof of Theorem 4.7.* We prove the first part of the theorem. Proofs of the other parts follow the same reasoning. The high-level argument is similar to the well known probabilistic argument in classical group testing, but we will have to use the tools that we have developed so far for working out the details. By construction, the output matrix $\boldsymbol{M}$ is vertex-consistent with $G$. Now, take a vertex $v \in V$ and $A \subseteq V$ such that $v \notin A$, $|A| \leq d$, and $(\{v\} \cup A) \cap \{s_1, \ldots, s_r\} = \emptyset$. For each $i = 1, \ldots m_1$, define a random variable $X_i \in \{0, 1\}$ such that $X_i = 1$ iff the $i$th row of $\boldsymbol{M}$ has a 1 entry at the column corresponding to $v$ and all-zeros at those corresponding to the elements of $A$. Let $X := \sum_{i=1}^{m_1} X_i$. Note that the columns corresponding to $v$ and $A$ violate the disjunctness property of $\boldsymbol{M}$ iff $X = 0$, and that the $X_i$ are independent Bernoulli random variables. Moreover,

$$\mathbb{E}[X_i] = \Pr[X_i = 1] = \pi_{v,A},$$

since $X_i = 1$ happens exactly when the $i$th random walk passes vertex $v$ but never hits any vertex in $A$. Now by using Lemma 4.16 we can ensure that, for an appropriate choice of $D_0$ and $t_1$ (as in the statement of the lemma), we have $\pi_{v,A} = \Omega(1/(c^4 d T^2(n)))$.

Denote by $p_f$ the *failure probability*, namely that the resulting matrix $\boldsymbol{M}$ is not $d$-disjunct. By a union bound we get

$$\begin{aligned} p_f &\leq \sum_{v,A} (1 - \pi_{v,A})^{m_1} \\ &\leq \exp\left(d \log \frac{n}{d}\right) \cdot \left(1 - \Omega\left(\frac{1}{c^4 d T^2(n)}\right)\right)^{m_1}. \end{aligned}$$

Thus by choosing

$$m_1 = O\left(d^2 c^4 T^2(n) \log \frac{n}{d}\right)$$

we can ensure that $p_f = o(1)$, and hence, $\boldsymbol{M}$ is $d$-disjunct with overwhelming probability.

For the claim on $(d, e_1)$-disjunctness, note that a failure occurs if, for some choice of the columns (i.e., some choice of $v, A$), we have $X \leq e_1$. Set

$$\eta' := \eta \pi_{v,A} = \Omega\left(\frac{\eta}{c^4 d T^2(n)}\right),$$

and $e_1 := \eta' m_1'$. Note that $\mathbb{E}[X] = \pi_{v,A} m_1'$. Now by a Chernoff bound, we get

$$\begin{aligned} \Pr[X \leq \eta' m_1'] &\leq \exp\left(-\frac{(\mathbb{E}[X] - \eta' m_1')^2}{2\mathbb{E}[X]}\right) \\ &= \exp(-\mathbb{E}[X](1 - \eta)^2/2). \end{aligned}$$

So now, by a union bound, the failure probability $p_f$ becomes

$$p_f \leq \exp\left(d \log \frac{n}{d} - m_1'(1 - \eta)^2 \tilde{\pi}/2\right), \tag{4.12}$$

where $\tilde{\pi}$ is the lower bound $\Omega(1/(c^4 d T^2(n)))$ on $\pi_{v,A}$. Thus we will have $p_f = o(1)$ by choosing

$$m_1' = O\left(d^2 \log \frac{n}{d} c^4 T^2(n)/(1 - \eta)^2\right).$$

□

*Proof of Theorem 4.9.* The proof follows line-by-line the same arguments as in the proof of Theorem 4.7, except that for the last union bound it would suffice to enumerate a substantially lower number of choices of $v$ and $A$. In particular, consider Design 1 as in the proof of Theorem 4.7 (the argument for the other designs is essentially the same). Then the only part of the proof that needs to be changed is the union bound from which (4.12) follows. Contrary to the proof of Theorem 4.7, in the case we consider here, only up to $n$ choices of the tuple $(v, A)$, in particular the following set, need to be enumerated:

$$\mathcal{B} := \{(v, A)\colon v \in V \setminus \{s_1, \ldots, s_r\}, A = S \setminus \{v\}\}.$$

Now assume that the resulting matrix "satisfies" all the choices of the tuples $(v, A) \in \mathcal{B}$, in that it has enough rows at which the entry corresponding to $v$ is 1 while those corresponding to $A$ are all zeros (this is guaranteed to hold, with overwhelming probability, by the union bound).

Consider the case where $S' \not\subseteq S$ and take any $v \in S' \setminus S$. Since $(v, S) \in \mathcal{B}$, we can be sure that the measurement outcome corresponding to $S'$ would be positive at more than $\Omega(\gamma e_1)$ of the positions while at those positions, the outcome of $S$ must be zero. A similar argument is true for the case $S' \subseteq S$, in which case it would suffice to take any $v \in S \setminus S'$ and observe that $(v, S \setminus \{v\}) \in \mathcal{B}$.

Altogether, from the above observations, the estimate (4.12) can be improved to

$$p_f \leq \exp\left(\log n - \tilde{m}(1 - \eta)^2 \tilde{\pi}/2\right),$$

where $\tilde{m}$ is the number of measurements. Therefore, we can ensure that $p_f = o(1)$ by taking $\tilde{m} = O(\gamma m'_1)$, i.e., a factor $\gamma$ less than what needed by Theorem 4.7. $\square$

## 4.5.2 Proof of Theorem 4.8

In Theorem 4.8 we consider two important instantiations of the result given by Theorem 4.7, namely when $G$ is taken as an expander graph with constant spectral gap, and when it is taken as an Erdős-Rényi random graph $G(n, p)$. In the following we show that in both cases (and provided that $p$ is not too small), the mixing time is $O(\log n)$ (with probability $1 - o(1)$). Then Theorem 4.7 will lead to the proof.

Before we proceed, we need to bound the distance between the stationary distribution and the distribution obtained after $t$ random steps on a graph. The following theorem, which is a direct corollary of a result in [SJ89], is the main tool that we will need. We skip the proof of this theorem here and refer to the main article for interested readers.

**Theorem 4.20** ([SJ89]). *Let $G$ be an undirected graph with stationary distribution $\mu$, and denote by $d_{\min}$ and $d_{\max}$ the minimum and maximum degrees of its vertices, respectively. Let $\mu_v^t$ be the distribution obtained by any random walk on $G$ in $t$ steps starting at node $v$. Then for all $v \in V$*

$$\|\mu_v^t - \mu\|_\infty \leq (1 - \Phi(G)^2/2)^t d_{\max}/d_{\min},$$

*where $\Phi(G)$ denotes the conductance of $G$ (see Definition 4.2).*

### 4.5.3 The Erdős-Rényi Random Graph

First, we present some tools for the case of random graphs. Consider a random graph $G(n, p)$ which is formed by removing each edge of the complete graph on $n$ vertices independently with probability $1 - p$. Our focus will be on the case where $np \gg \ln n$. In this case, the resulting graph becomes (almost surely) connected and the degrees are highly concentrated around their expectations. In particular, we can show the following fact, which we believe to be folklore. The proof of this is presented in Section 4.5.11.

**Proposition 4.21.** *For every $\epsilon > 0$, with probability $1 - o(1)$, the random graph $G(n, p)$ with $np \geq (2/\epsilon^2) \ln n$ is $(np(1 - \epsilon), (1 + \epsilon)/(1 - \epsilon))$-uniform.*

In light of Theorem 4.20, all we need to show is a lower bound on the conductance of a random graph. This is done in the following.

**Lemma 4.22.** *For every $\varphi < 1/2$, there is an $\alpha > 0$ such that a random graph $G = G(n, p)$ with $p \geq \alpha \ln n/n$ has conductance $\Phi(G) \geq \varphi$ with probability $1 - o(1)$.*

*Proof.* First, note that by Proposition 4.21 we can choose $\alpha$ large enough so that with probability $1 - o(1)$, the degree of each vertex in $G$ is between $D(1 - \epsilon)$ and $D(1 + \epsilon)$, for an arbitrarily small $\epsilon > 0$ and $D := np$. We will suitably choose $\epsilon$ later.

Fix a set $S \subseteq V$ of size $i$. We wish to upper bound the probability that $S$ makes the conductance of $G$ undesirably low, i.e., the probability that $E(S, \bar{S}) < \varphi\Delta(S)$. Denote this probability by $p_S$. By the definition of conductance and $(D, \epsilon)$-uniformity of $G$, we only need to consider subsets of size at most $\eta n$, for $\eta := (1 + \epsilon)/2(1 - \epsilon)$.

There are $i(n - i)$ "potential" edges between $S$ and its complement in $G$, where each edge is taken independently at random with probability $p$. Therefore, the expected size of $E(S, \bar{S})$ is

$$\nu := Di(1 - i/n) \geq Di(1 - \eta).$$

Now note that the event $E(S, \bar{S}) < \varphi\Delta(S)$ implies that

$$E(S, \bar{S}) < \varphi Di(1 + \epsilon) < \varphi'\nu,$$

where $\varphi' := \varphi(1+\epsilon)/(1-\eta)$. So it suffices the upper bound the probability that $E(S, \bar{S}) < \varphi'\nu$. Note that, since $\varphi < 1/2$, we can choose $\epsilon$ small enough to ensure that $\varphi' < 1$. Now, by a Chernoff bound,

$$\begin{aligned} p_S &\leq \Pr[E(S, \bar{S}) < \varphi'\nu] \\ &\leq \exp(-(1 - \varphi')^2\nu) \\ &\leq n^{-i\alpha(1-\varphi')^2(1-\eta)}. \end{aligned}$$

Set $\alpha$ large enough (i.e., $\alpha \geq 2/(1 - \varphi')^2(1 - \eta)$) so that the right hand side becomes at most $n^{-2i}$. Therefore, with high probability, for our particular choice of $S$ we have $E(S, \bar{S})/\varphi(S) \geq \varphi$.

Now we take a union bound on all possible choices of $S$ to upper bound the probability of conductance becoming small as follows.

$$\begin{aligned} \Pr[\Phi(G) < \varphi] &\leq \sum_{i=1}^{\eta n} \binom{n}{i} n^{-2i} \\ &\leq \sum_{i=1}^{\eta n} n^{-i} = o(1). \end{aligned}$$

Thus with probability $1 - o(1)$, we have $\Phi(G) \geq \varphi$.

□

By combining Lemma 4.22 and Theorem 4.20 we get the following corollary, which is formally proved in Section 4.5.12.

**Corollary 4.23.** *There is an $\alpha > 0$ such that a random graph $G = G(n, p)$ with $p \geq \alpha \ln n/n$ has $\delta$-mixing time bounded by $O(\log(1/\delta))$ with probability $1 - o(1)$.*

In particular, for our specific choice of $\delta := (1/2cn)^2$, the $\delta$-mixing time of $G(n, p)$ would be $T(n) = O(\log n)$.

### 4.5.4 Expander Graphs with Constant Spectral Gap

Similar to Corollary 4.23, we need to show that the mixing time of an expander graph with second largest eigenvalue that is bounded away from 1 is bounded by $O(\log n)$.

**Lemma 4.24.** *If $G = (V, E)$ is an expander graph with a (normalized) second largest eigenvalue that is bounded away from 1 by a constant, then $T(n) = O(\log n)$.*

*Proof.* We first recall a well known result in graph theory (cf. [HLW06]), which states that any regular graph with a normalized adjacency matrix whose second largest eigenvalue (in absolute value) is bounded away from 1 must have good expansion (i.e., $\sigma = \Omega(1)$).

Moreover, note that for regular graphs we have $\Delta(S) = D|S|$, and therefore the two notions of conductance (see Definition 4.2) and expansion (Definition 4.4) coincide (except a multiplicative constant).

Finally, we can applying Theorem 4.20 to find the smallest $t$ which satisfies

$$(1 - \Phi(G)^2/2)^t \leq \frac{1}{(2n)^2},$$

which implies $T(n) = O(\log n)$. □

We now have all the tools required for proving Theorem 4.8.

*Proof of Theorem 4.8.* Follows immediately by combining Theorem 4.7, Proposition 4.21, Corollary 4.23, and Lemma 4.24. □

### 4.5.5 Proof of Proposition 4.10

We can write

$$\begin{aligned}\Pr[A \mid B] &= \frac{\sum_{i=1}^{n} \Pr[A|B_i]\Pr[B_i]}{\Pr[B]} \\ &\leq \epsilon \cdot \frac{\sum_{i=1}^{n} \Pr[B_i]}{\Pr[B]} \\ &\leq \epsilon k.\end{aligned}$$

The last inequality is due to the fact that each element of the sample space can belong to at most $k$ of the $B_i$, and thus, the summation $\sum_{i=1}^{n} \Pr[B_i]$ counts the probability of each element in $B$ at most $k$ times.

### 4.5.6 Proof of Proposition 4.11

We start with a well-known result [MU05, Theorem 7.13], that a random walk on any graph graph $G$ that is not bipartite converges to a stationary distribution $\mu$, where

$$\mu(v) = \frac{d(v)}{2|E|}.$$

Since $G$ is a $(D, c)$-uniform graph we know that $D \leq d(v) \leq cD$ and that $nD \leq 2|E| \leq ncD$.

### 4.5.7 Proof of Proposition 4.12

Let $t' := \lfloor t/T(n) \rfloor$, and for each $i \in \{0, \dots, t'\}$, $w_i := v_{iT(n)}$. Denote by $W' := \{w_0, \dots, w_{t'}\}$ a subset of $t' + 1$ vertices visited by $W$. Obviously, $\pi_v$ is at least the probability that $v \in W'$. Thus it suffices to lower bound the latter probability.

By the definition of mixing time, regardless of the choice of $w_0$, the distribution of $w_1$ is $\delta$-close to the stationary distribution $\mu$, which assigns a probability between $1/cn$ and $c/n$ to $v$ (by Proposition 4.11). Therefore, $\Pr[w_1 \neq v \mid w_0] \leq 1 - 1/cn + \delta$. Similarly, $\Pr[w_2 \neq v \mid w_0, w_1] \leq 1 - 1/2cn$, and so on. Altogether, this means that

$$\begin{aligned} \Pr[w_0 \neq v, w_1 \neq v, \dots, w_{t'} \neq v] &\leq (1 - 1/cn + \delta)^{t/T(n)} \\ &\leq (1 - 1/2cn)^{t/T(n)} \\ &\leq \exp(-t/(2cnT(n))) \\ &\leq 1 - \Omega(t/(cnT(n))). \end{aligned}$$

In the last equality we used the fact that $\exp(-x) \leq 1 - x/2$ for $0 \leq x \leq 1$. Thus the complement probability is lower bounded by $\Omega(t/(cnT(n))$. The calculation for $\pi_e$ is similar.

### 4.5.8 Proof of Proposition 4.13

For every $i = 0, \dots, t$, define a boolean random variable $X_i \in \{0, 1\}$ such that $X_i = 1$ iff $v_i = v$. Let $X := \sum_{i=0}^{t} X_i$ be the number of times that the walk visits $v$. For every $i \geq T(n)$, we have

$$\begin{aligned} \mathbb{E}[X_i] &= \Pr[v_i = v] \\ &\leq c/n + \delta \\ &\leq 2c/n, \end{aligned}$$

where the first inequality is due to the assumption that $i \geq T(n)$ and after the mixing time, the distribution induced on each vertex is within $\delta$ of the stationary distribution, and the second inequality is by the particular choice of the proximity parameter $\delta$. Define $X' := \sum_{i=T(n)}^{t} X_i$. By linearity of expectation, $\mathbb{E}[X'] < 2ct/n$, and by Markov's inequality,

$$\Pr[X' \geq \alpha c^2 T(n)] < \frac{2t}{\alpha c n T(n)}.$$

By taking $\alpha$ a large constant (depending on the constant hidden in the asymptotic estime of $\pi_v$ given by Lemma 4.12), and using Proposition 4.12, we can ensure that the bound on the probability is at most $\pi_v/4$. Thus the probability that $X \geq k$ for $k := (1 + \alpha c^2)T(n)$ is at most $\pi_v/4$. Proof for the edge case is similar.

### 4.5.9 Proof of Proposition 4.14

By the choice of $v$ (that is not a designated vertex), the walk $W$ has a chance of visiting $v$ as the initial vertex $v_0$ only if it starts at a vertex chosen uniformly at random. Thus the probability of visiting $v$ at the initial step is $1/n \leq 1/D$.

Now, regardless of the outcome of the initial vertex $v_0$, the probability of visiting $v$ as the second vertex $v_1$ is at most $1/D$, as $v_0$ has at least $D$ neighbors and one is chosen uniformly at random. Thus, $\Pr[v_1 = v] \leq 1/D$, and similarly, for each $i$, $\Pr[v_i = v] \leq 1/D$. A union bound gives the claim.

### 4.5.10 Proof of Proposition 4.15

We can write

$$\Pr[v_i = u \mid v_j = v, \mathcal{E}] = \Pr[v_j = v \mid v_i = u, \mathcal{E}] \cdot \frac{\Pr[v_i = u \mid \mathcal{E}]}{\Pr[v_j = v \mid \mathcal{E}]}.$$

Now, from the definition of mixing time, we know that

$$|\Pr[v_j = v \mid v_i = u, \mathcal{E}] - \Pr[v_j = v \mid \mathcal{E}]| \leq 2\delta,$$

because regardless of the knowledge of $v_i = u$, the distribution of $v_j$ must be $\delta$-close to the stationary distribution. Therefore,

$$\begin{aligned} |\Pr[v_i = u | v_j = v, \mathcal{E}] - \Pr[v_i = u | \mathcal{E}]| &\leq 2\delta / \Pr[v_j = v \mid \mathcal{E}] \\ &\leq 2\delta/(1/cn - \delta) \\ &\leq 8\delta cn/3. \end{aligned}$$

### 4.5.11 Proof of Proposition 4.21

Let $\alpha := 6/\epsilon^2$ so that $np \geq \alpha \ln n$. Take any vertex $v$ of the graph. The expected degree of $v$ is $np$. As the edges are chosen independently, by a Chernoff bound, the deviation probability of $\deg(v)$ can be bounded as

$$\begin{aligned} \Pr[|\deg(v) - np| > \epsilon np] &\leq 2e^{-\epsilon^2 np/3} \\ &\leq 2n^{-\epsilon^2\alpha/3} = 2/n^2. \end{aligned}$$

This upper bounds the probability by $2n^{-2}$. Now we can use a union bound on the vertices of the graph to conclude that with probability at least $1 - 2/n$, the degree of each vertex in the graph is between $np(1-\epsilon)$ and $np(1+\epsilon)$.

### 4.5.12 Proof of Corollary 4.23

Choose $\alpha$ large enough so that, by Proposition 4.21 the graph becomes $(np(1-\epsilon), (1+\epsilon)/(1-\epsilon))$-uniform, for a sufficiently small $\epsilon$ and so that Lemma 4.22 can be applied to obtain $\Phi(G) = \Omega(1)$. Let $\mu'$ be the distribution obtained by any random walk on $G$ in $t$ steps and denote by $\mu$ the stationary distribution of $G$. Now Theorem 4.20 implies that,

$$\|\mu' - \mu\|_\infty \leq (1 - \Phi(G)^2/2)^t (1+\epsilon)/(1-\epsilon),$$

and thus, it suffices to choose $t = O(\log(1/\delta))$ to have $\|\mu' - \mu\|_\infty \leq \delta$.

“· ···– · ·–· –·–– ·–– ···· · ·–· ·   ·· ···
·–– ·· – ···· ·· –·   ·–– ·– ·–·· –·– ·· –· ––·
–·· ·· ··· – ·– –· –·–· ·   ·· ··–·   –·–– ––– ··–
···· ·– ···– ·   – ···· ·   – ·· –– · ······”

··· – · ···– · –·   ·–– ·–· ·· ––· ···· –

# 5

# Navigability with a Bias

In the first two chapters, we studied efficient algorithms for inferring the position of the nodes for a given network. We then looked at network tomography or how we can identify congested links/nodes from end-to-end measurements. In this chapter [1], we investigate another important feature of networks, namely, navigability or how fast nodes can deliver a message from one end to another by the means of local operations.

When each node has a full view or global map of the network, finding the shortest route and hence routing information is only a matter of computation. However, in many networks, including social, peer-to-peer and neural networks, nodes can potentially communicate efficiently with one another even without such global knowledge. This phenomenon, known as the *small-world* or *six degrees of separation*, dates back to the famous experiment conducted by Milgram in the 1960's [Mil67]. Milgram's experiment was subsequently confirmed by others under many different settings in recent years [NNK$^+$05, DMW03]. These studies revealed that in such networks, the diameter is exponentially smaller than the size. In other words, short routes exist in abundance. Many models have been proposed in the literature to capture this property [WS98, Wat99, BC88]. However, as Kleinberg observed in his seminal work [Kle00], the existence of such short paths does not imply that a decentralized algorithm can find them. By revisiting Milgram's experiment, Kleinberg [Kle00] proposed a mathematical model similar to that of [WS98] to capture this aspect of small-worlds.

In Kleinberg's model, nodes of the network are embedded in a 2-dimensional grid which are then augmented with additional shortcut edges as follows (see Figure 5.1). From each node $x$ in the network, a directed edge is added to some other node $y$ according to a probability distribution that depends on the distance between $x$ and $y$ (in contrast to [WS98] where it is assumed that this probability is uniform). More specifically, node $x$ is connected to node $y$ on the grid with probability proportional to $d^{-\alpha}(x, y)$. Kleinberg showed that for the critical value of $\alpha^* = 2$, there exists a decentralized algorithm, called "greedy forwarding", that finds paths of lengths poly-logarithmic in the size of the network. He also established that for $\alpha < \alpha^*$ such short path exists but no decentralized algorithm will be able to find them. When $\alpha > \alpha^*$, short paths no longer exist. As a result, to route a

---

[1] This chapter is the result of a collaboration with S. Ioannidis and L. Massoulié.

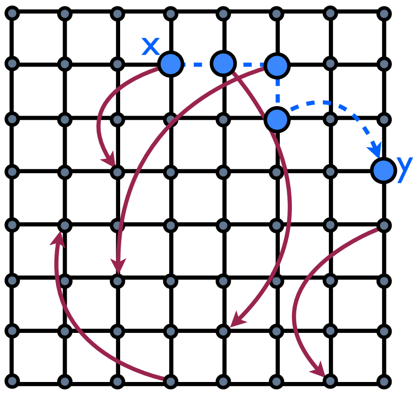


Figure 5.1 – Each node of the graph is augmented with a shortcut edge (a few of them are shown). As shown in this figure, the greedy forwarding algorithm takes the blue dashed path to route a message from $x$ to $y$.

message in a decentralized manner from one node to another, the number of hops/steps it takes is of the order of network size, unless $\alpha = \alpha^*$. This result can be easily generalized to a $k$-dimensional grid in which case a shortcut edge between nodes $x$ and $y$ should be sampled with a probability proportional to $d^{-k}(x, y)$.

In this chapter, we consider the following general framework for designing small world networks which naturally extends Kleinbreg's result:

> Given a graph embedded in a metric space, how should one augment this graph by adding edges in order to minimize the expected cost of greedy forwarding over this graph?

Note that in this framework, nodes are not necessarily embedded in a 2-dimensional gird. Moreover, we consider the above problem under the scenario of *heterogeneous demand*, i.e., the probability that a message is sent from a node $x$ to another node $y$ is not uniform. Earlier work on small-world network design focuses only on homogeneous demand. Our contributions in this chapter are as follows:

- We show that the small-world network design problem under general heterogeneous demand is NP-hard. Given earlier work on this problem under homogeneous demand [FG10, FLL06], this result is interesting in its own right.
- We propose a novel mechanism for edge addition in the small-world design problem under the heterogeneous demand, and provide an upper bound on its performance.

Our analysis generalizes and combines classical results from small-world network design in the context of heterogeneity. In particular, our upper bound relates the cost of greedy forwarding to two important properties of the demand distribution, namely its *entropy* and its *doubling constant*. We thus provide performance guarantees in terms of the *bias* of the distribution of targets, captured by the entropy, as well as the *topology* of their embedding, captured by the doubling constant.

## 5.1 Related Work

Small world network design has been an active field of research in the past decade. It has found numerous applications, including decentralized searching [Kle01], recommendation systems [WBS08], and peer-to-peer networks [ZGG02], to name a few. A recent survey of the field with a detailed account of new directions can be found in [Kle06]. Following Kleinberg's result, many tried to extend it to more general settings. In [Kle01], Kleinberg extended his model to hierarchical structures using a complete $b$-ary tree. The small world networks with grid-like structures were analysed in [NM05], where the diameter of such networks were computed. A more general framework, where nodes are embedded in a metric space of bounded doubling dimension was considered in [Sli05].

Our work is most similar to Fraigneaud *et al.* [FLL06], where a condition under which graphs embedded in a doubling metric space can be made navigable is identified. The same idea was explored in more general spaces by Fraigneaud and Giakkoupis [FG10]. The main difference in our approach to small-world network design lies in considering heterogeneous demand, an aspect of small-world networks not investigated in earlier work.

## 5.2 Definitions and Notation

In this section, we introduce some definitions and notation which will be used throughout this chapter.

### Objects and Metric Embedding

Consider a set of objects $\mathcal{N}$, where $|\mathcal{N}| = n$. We assume that there exists a metric space $(\mathcal{M},d)$, where $d(x,y)$ denotes the distance between $x, y \in \mathcal{M}$, such that objects in $\mathcal{N}$ are embedded in $(\mathcal{M},d)$: *i.e.*, there exists a one-to-one mapping from $\mathcal{N}$ to a subset of $\mathcal{M}$.

The objects in $\mathcal{N}$ may represent, for example, nodes of a peer-to-peer network or the set of individuals in a social network.

Given an object $z \in \mathcal{N}$, we can order objects according to their distance from $z$. We will write $x \preccurlyeq_z y$ if $d(x,z) \le d(y,z)$. Moreover, we will write $x \sim_z y$ if $d(x,z) = d(y,z)$ and $x \prec_z y$ if $x \preccurlyeq_z y$ but not $x \sim_z y$. Note that $\sim_z$ is an equivalence relation, and hence partitions $\mathcal{N}$ into equivalence classes. Moreover, $\preccurlyeq_z$ defines a total order over these equivalence classes, with respect to their distance from $z$. Given a non-empty set $A \subseteq \mathcal{N}$, we denote by $\min_{\preccurlyeq_z} A$ the object in $A$ closest to $z$, *i.e.* $\min_{\preccurlyeq_z} A = w \in A$ s.t. $w \preccurlyeq_z v$ for all $v \in A$.

### Demand

We denote by $\mathcal{N} \times \mathcal{N}$ the set of all ordered pairs of objects in $\mathcal{N}$. For $(s,t) \in \mathcal{N} \times \mathcal{N}$, we will call $s$ the *source* and $t$ the *target* of the ordered pair. We will consider a probability distribution $\lambda$ over all ordered pairs of objects in $\mathcal{N}$ which we will call the *demand*. In other words, $\lambda$ will be a non-negative function such that $\sum_{(s,t)\in\mathcal{N}\times\mathcal{N}} \lambda(s,t) = 1$. In general, the demand can be *heterogeneous* as $\lambda(s,t)$ may vary across different sources and targets. We refer to the marginal distributions

$$\nu(s) = \sum_t \lambda(s,t), \quad \mu(t) = \sum_s \lambda(s,t),$$

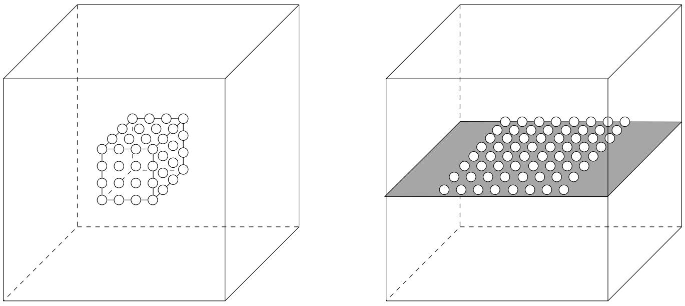


Figure 5.2 – Example of dependence of $c(\sigma)$ on the topology of the support $\mathsf{supp}(\sigma)$. When $\mathsf{supp}(\sigma)$ consists of $n = 64$ objects arranged in a cube, $c(\sigma) = 2^3$. If, on the other hand, these $n$ objects are placed on a plane, $c(\sigma) = 2^2$. In both cases $\sigma$ is assumed to be uniform, and $H(\sigma) = \log N$

as the *source* and *target* distributions, respectively. Moreover, will refer to the support of the target distribution

$$\mathcal{T} = \mathsf{supp}(\mu) = \{x \in \mathcal{N} : \text{ s.t. } \mu(x) > 0\}$$

as the *target set* of the demand.

As we will see in Section 5.4, the target distribution $\mu$ will play an important role in our analysis. In particular, two quantities that affect the performance of searching in our scheme will be the *entropy* and the *doubling constant* of the target distribution. We introduce these two notions formally below.

### Entropy and Doubling Constant

Let $\sigma$ be a probability distribution over $\mathcal{N}$. The *entropy* of $\sigma$ is defined as

$$H(\sigma) = \sum_{x \in \mathsf{supp}(\sigma)} \sigma(x) \log \frac{1}{\sigma(x)}. \tag{5.1}$$

We define the *max-entropy* of $\sigma$ as

$$H_{\max}(\sigma) = \max_{x \in \mathsf{supp}(\sigma)} \log \frac{1}{\sigma(x)}. \tag{5.2}$$

Given an object $x \in \mathcal{N}$, we denote by

$$B_x(r) = \{y \in \mathcal{M} : d(x, y) \leq r\} \tag{5.3}$$

the closed ball of radius $r \geq 0$ around $x$. Given a probability distribution $\sigma$ over $\mathcal{N}$ and a set $A \subset \mathcal{N}$ let $\sigma(A) = \sum_{x \in A} \sigma(x)$. We define the *doubling constant* $c(\sigma)$ of a distribution $\sigma$ to be the minimum $c > 0$ for which

$$\sigma(B_x(2r)) \leq c \cdot \sigma(B_x(r)), \tag{5.4}$$

for any $x \in \mathsf{supp}(\sigma)$ and any $r \geq 0$. Moreover, we will say that $\sigma$ is *c-doubling* if $c(\mu) = c$.

Note that, contrary to the entropy $H(\sigma)$, the doubling constant $c(\sigma)$ depends on the topology of $\mathsf{supp}(\sigma)$, determined by the embedding of $\mathcal{N}$ in the metric space $(\mathcal{M}, d)$. This is illustrated in Fig. 5.2. In this example, $|\mathcal{N}| = 64$, and the set $\mathcal{N}$ is embedded in a 3-dimensional cube. Assume that $\sigma$ is the uniform distribution over the $N$ objects; if these objects are arranged uniformly in a cube, then $c(\sigma) = 2^3$; if however these $n$ objects are arranged uniformly in a 2-dimensional plane, $c(\sigma) = 2^2$. Note that, in contrast, the entropy of $\sigma$ in both cases equals $\log n$ (and so does the max-entropy).

## 5.3 Problem Statement

In the small network design problem, we consider the objects in $\mathcal{N}$, embedded in $(\mathcal{M}, d)$. It is also assumed that the objects in $\mathcal{N}$ are connected to each other. The network formed by such connections is represented by a directed graph $G(\mathcal{N}, \mathcal{L} \cup \mathcal{S})$, where $\mathcal{L}$ is the set of *local* edges and $\mathcal{S}$ is the set of *shortcut* edges. These edge sets are disjoint, *i.e.*, $\mathcal{L} \cap \mathcal{S} = \emptyset$.

The edges in $\mathcal{L}$ are typically assumed to satisfy the following property:

**Property 5.1.** *For every pair of distinct objects $x, t \in \mathcal{N}$ there exists an object $u$ adjacent to $x$ such that $(x, u) \in \mathcal{L}$ and $u \prec_t x$.*

In other words, for any object $x$ and a target $t$, $x$ has a local edge leading to an object closer to $t$.

Here the goal is to route a message from $s$ to $t$ over the links in graph $G$ in a decentralized manner. In particular, given graph $G$, we define *greedy forwarding* [Kle00] over $G$ as follows. Let $\Gamma(s)$ be the neighborhood of $s$, *i.e.*, $\Gamma(s) = \{u \in \mathcal{N} \text{ s.t. } (s, u) \in \mathcal{L} \cup \mathcal{S}\}$. Given a source $s$ and a target $t$, greedy forwarding sends a message to neighbor $w$ of $s$ that is as close to $t$ as possible, *i.e.*,

$$w = \min_{\preccurlyeq_t} \Gamma(s). \tag{5.5}$$

If $w \neq t$, the above process is repeated at $w$; if $w = t$, greedy forwarding terminates.

Note that local edges, through Property 5.1, guarantee that greedy forwarding from any source $s$ will eventually reach $t$: there will always be a neighbor that is closer to $t$ than the object currently having the message. Moreover, the closest neighbour $w$ selected through (5.5) can be found by using only local information. In particular, if the message is at an object $x$, $|\Gamma(x)|$ comparisons will suffice to find the neighbor that is closest to the target.

The edges in $\mathcal{L}$ are typically called “local” because they are usually determined by object proximity. For example, in the classical paper by [Kle00], objects are arranged uniformly in a rectangular $k$-dimensional grid—with no gaps—and $d$ is taken to be the Manhattan distance on the grid. Moreover, there exists an $r \geq 1$ such that any two objects at distance less than $r$ have an edge in $\mathcal{L}$. In other words,

$$\mathcal{L} = \{(x, y) \in \mathcal{N} \times \mathcal{N} \text{ s.t. } d(x, y) \leq r\}. \tag{5.6}$$

Assuming every position in the rectangular grid is occupied, such edges indeed satisfy Property 5.1. In this work, we will not require that edges in $\mathcal{L}$ are given by (5.6) or some other locality-based definition; our only assumption is that they satisfy Property 5.1. Nevertheless, for the sake of consistency with prior work, we also refer to edges in $\mathcal{L}$ as “local”.

The shortcut edges $\mathcal{S}$ need not satisfy Property 5.1; our goal is to select these shortcut edges in a way so that greedy forwarding is as efficient as possible.

In particular, we assume that we can select no more than $\beta$ shortcut edges, where $\beta$ is a positive integer. For $S$ a subset of $\mathcal{N} \times \mathcal{N}$ such that $|S| \leq \beta$, we denote by $C_S(s,t)$ the cost of greedy forwarding, in message hops, for forwarding a message from $s$ to $t$ given that $\mathcal{S} = S$. We allow the selection of shortcut edges to be random: the set $\mathcal{S}$ can be a random variable over all subsets $S$ of $\mathcal{N} \times \mathcal{N}$ such that $|S| \leq \beta$. We denote by

$$\Pr(\mathcal{S} = S), \quad S \subseteq \mathcal{N} \times \mathcal{N} \text{ s.t. } |S| \leq \beta \tag{5.7}$$

the distribution of $\mathcal{S}$. Given a source $s$ and a target $t$, let

$$\mathbb{E}[C_{\mathcal{S}}(s,t)] = \sum_{S \subseteq \mathcal{N} \times \mathcal{N} : |S| \leq \beta} C_S(s,t) \cdot \mathbb{P}(\mathcal{S} = S)$$

be the expected cost of forwarding a message from $s$ to $t$ with greedy forwarding, in message hops.

We consider again a heterogeneous demand: a source and target object are selected at random from $\mathcal{N} \times \mathcal{N}$ according to a demand probability distribution $\lambda$. The *small-world network design problem* can then be formulated as follows.

> Small-World Network Design (SWND): Given an embedding of $\mathcal{N}$ into $(\mathcal{M}, d)$, a set of local edges $\mathcal{L}$, a demand distribution $\lambda$, and an integer $\beta > 0$, select a r.v. $\mathcal{S} \subset \mathcal{N} \times \mathcal{N}$ that minimizes
>
> $$\bar{C}_{\mathcal{S}} = \sum_{(s,t) \in \mathcal{N} \times \mathcal{N}} \lambda(s,t) \mathbb{E}[C_{\mathcal{S}}(s,t)]$$
>
> subject to $|\mathcal{S}| \leq \beta$.

In other words, we wish to select $\mathcal{S}$ so that the cost of greedy forwarding is minimized. Note that, since $\mathcal{S}$ is a random variable, the free variable of the above optimization problem is essentially the distribution of $\mathcal{S}$, given by (5.7).

## 5.4 Main Results

We formally state our results (Theorems 5.2 and 5.3) in this section, and present their proofs in Sections 5.5.1 and 5.5.2.

Our first result is negative: optimizing greedy forwarding is a hard problem.

**Theorem 5.2.** *SWND is NP-hard.*

In short, the proof reduces DominatingSet to the decision version of SWND. Interestingly, the reduction is to a SWND instance in which (a) the metric space is a 2-dimensional grid, (b) the distance metric is the Manhattan distance on the grid and (c) the local edges are given by (5.6). Thus, SWND remains NP-hard even in the original setup considered by [Kle00].

The NP-hardness of SWND suggests that this problem cannot be solved in its full generality. Motivated by this we focus our attention to the following version of the SWND problem, in which we place additional restrictions on the shortcut edge set $\mathcal{S}$. First,

$|\mathcal{S}| = |\mathcal{N}|$, and for every $x \in \mathcal{N}$ there exists *exactly one* shortcut edge $(x, y) \in \mathcal{S}$. Second, the object $y$ to which $x$ connects is selected independently at each $x$, according to a probability distribution $\ell_x(y)$. More specifically, for $\mathcal{N} = \{x_1, x_2, \ldots, x_n\}$, the joint distribution of shortcut edges has the form:

$$\Pr(\mathcal{S} = \{(x_1, y_1), \ldots (x_n, y_n)\}) = \prod_{i=1}^{n} \ell_{x_i}(y_i). \tag{5.8}$$

We call this version of the SWND problem the *one edge per object* version, and denote it by 1-SWND. Note that, in 1-SWND, the free variables are the distributions $\ell_x$, $x \in \mathcal{N}$.

For a given demand $\lambda$, recall that $\mu$ is the marginal distribution of the demand $\lambda$ over the target set $\mathcal{T}$, and that for $A \subset \mathcal{N}$, $\mu(A) = \sum_{x \in A} \mu(x)$. Then, for any two objects $x, y \in \mathcal{N}$, we define the *rank* of object $y$ w.r.t. object $x$ as follows:

$$r_x(y) \equiv \mu(B_x(d(x, y))) \tag{5.9}$$

where $B_x(r)$ is the closed ball with radius $r$ centered at $x$.

Suppose now that shortcut edges are generated according to the joint distribution (5.8), where the outgoing link from an object $x \in \mathcal{N}$ is selected according to the following probability:

$$\ell_x(y) \propto \frac{\mu(y)}{r_x(y)}, \tag{5.10}$$

for $y \in \mathsf{supp}(\mu)$, while for $y \notin \mathsf{supp}(\mu)$ we define $\ell_x(y)$ to be zero. Eq. (5.10) implies the following appealing properties:

- For two objects $y, z$ that have the same distance from $x$, if $\mu(y) > \mu(z)$ then $\ell_x(y) > \ell_x(z)$, *i.e.*, y has a higher probability of being connected to $x$.
- When two objects $y, z$ are equally likely to be targets, if $y \prec_x z$ then $\ell_x(y) > \ell_x(z)$.

The distribution (5.10) thus biases both towards objects close to $x$ as well as towards objects that are likely to be targets. Finally, if the metric space $(\mathcal{M}, d)$ is a $k$-dimensional grid and the targets are uniformly distributed over $\mathcal{N}$ then $\ell_x(y) \propto (d(x, y))^{-k}$. This is the shortcut distribution used by [Kle00]; (5.10) is thus a generalization of this distribution to heterogeneous targets as well as to more general metric spaces.

Our next theorem, whose proof is in Section 5.5.2, relates the cost of greedy forwarding under (5.10) to the entropy $H$, the max-entropy $H_{\max}$ and the doubling parameter $c$ of the target distribution $\mu$.

**Theorem 5.3.** *Given a demand $\lambda$, consider the set of shortcut edges $\mathcal{S}$ sampled according to* (5.8)*, where $\ell_x(y)$, $x, y \in \mathcal{N}$, are given by* (5.10)*. Then*

$$\bar{C}_{\mathcal{S}} \leq 6c^3(\mu) \cdot H(\mu) \cdot H_{\max}(\mu).$$

Note that the bound in Theorem 5.3 depends on $\lambda$ only through the target distribution $\mu$. In particular, it holds for *any* source distribution $\nu$, and *does not require* that sources are selected independently of the targets $t$. Moreover, if $\mathcal{N}$ is a $k$-dimensional grid and $\mu$ is the uniform distribution over $\mathcal{N}$, the above bound becomes $O(\log^2 n)$, retrieving thus the result of [Kle00].

## 5.5 Analysis

This section includes the proofs of our theorems.

### 5.5.1 Proof of Theorem 5.2

We first prove that the randomized version of SWND is no harder than its deterministic version. Define DetSWND to be the same as SWND with the additional restriction that $\mathcal{S}$ is deterministic. For any random variable $\mathcal{S} \subset \mathcal{N}$ that satisfies $|\mathcal{S}| \leq \beta$, there exists a deterministic set $S^*$ s.t. $|S^*| \leq \beta$ and $\bar{C}_{S^*} \leq \bar{C}_{\mathcal{S}}$. In particular, this is true for

$$S^* = \arg \min_{S \in \mathcal{N}, |S| \leq \beta} C_S(s,t).$$

Thus, SWND is equivalent to DetSWND. In particular, any solution of DetSWND will also be a solution of SWND. Moreover, given a solution $\mathcal{S}$ of SWND any deterministic $S$ belonging to the support of $\mathcal{S}$ will be a solution of DetSWND.

We therefore turn our attention on DetSWND. Without loss of generality, we can assume that the weights $\lambda(s,t)$ are arbitrary non-negative numbers, as dividing every weight by $\sum_{s,t} \lambda(s,t)$ does not change the optimal solution. The decision problem corresponding to DetSWND is as follows

> DetSWND-D: Given an embedding of $\mathcal{N}$ into $(\mathcal{M}, d)$, a set of local edges $\mathcal{L}$, a non-negative weight function $\lambda$, and two constants $\alpha > 0$ and $\beta > 0$, is there a directed edge set $S$ such that $|S| \leq \beta$ and $\sum_{(s,t) \times \mathcal{N} \times \mathcal{N}} \lambda(s,t) C_S(s,t) \leq \alpha$?

Note that, given the set of shorcut edges $S$, forwarding a message with greedy forwarding from any $s$ to $t$ can take place in polynomial time. As a result, DetSWND-D is in NP. We will prove it is also NP-hard by reducing the following NP-complete problem to it:

> DominatingSet: Given a graph $G(V,E)$ and a constant $k$, is there a set $A \subseteq V$ such that $|A| \leq k$ and $\Gamma(A) \cup A = V$, where $\Gamma(A)$ the neighborhood of $A$ in $G$?

Given an instance $(G(V,E), k)$ of DominatingSet, we construct an instance of DetSWND-D as follows. The set $\mathcal{N}$ in this instance will be embedded in a 2-dimensional grid, and the distance metric $d$ will be the Manhattan distance on the grid. In particular, let $n = |V|$ be the size of the graph $G$ and, w.l.o.g., assume that $V = \{1, 2, \ldots, n\}$. Let

$$\ell_0 = 6n + 3, \tag{5.11}$$

$$\ell_1 = n\ell_0 + 2 = 6n^2 + 3n + 2, \tag{5.12}$$

$$\ell_2 = \ell_1 + 3n + 1 = 6n^2 + 6n + 3. \tag{5.13}$$

$$\ell_3 = \ell_0 = 6n + 3, \tag{5.14}$$

We construct a $n_1 \times n_2$ grid, where $n_1 = (n-1) \cdot \ell_0 + 1$ and $n_2 = \ell_1 + \ell_2 + \ell_3 + 1$. That is, the total number of nodes in the grid is

$$N = [(n-1) \cdot \ell_0 + 1] \cdot (\ell_1 + \ell_2 + \ell_3 + 1) = \Theta(n^4).$$

The object set $\mathcal{N}$ will be the set of nodes in the above grid, and the metric space will be $(\mathbb{Z}^2, d)$ where $d$ is the Manhattan distance on $\mathbb{Z}^2$. The local edges $\mathcal{L}$ is defined according to (5.6) with $r = 1$, *i.e.*, and any two adjacent nodes in the grid are connected by an edge in $\mathcal{L}$.

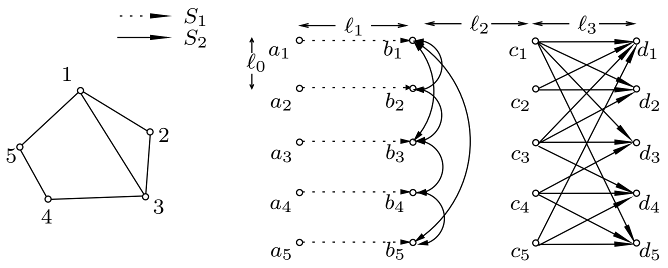


Figure 5.3 – A reduction of an instance of DominatingSet to an instance of DetSWND-D. Only the nodes on the grid that have non-zero incoming or outgoing demands (weights) are depicted. The dashed arrows depict $A_1$, the set of pairs that receive a weight $W_1$. The solid arrows depict $A_2$, the set of pairs that receive weight $W_2$.

Denote by $a_i$, $i = 1, \dots, n$, the node on the first column of the grid that resides at row $(i-1)\ell_0 + 1$. Similarly, denote by $b_i$, $c_i$ and $d_i$ the nodes on the columns $(\ell_1 + 1)$, $(\ell_1 + \ell_2 + 1)$ and $(\ell_1 + \ell_2 + \ell_3 + 1)$ the grid, respectively, that reside at the same row as $a_i$, $i = 1, \dots, n$. These nodes are depicted in Figure 5.3. We define the weight function $\lambda(i, j)$ over the pairs of nodes in the grid as follows. The pairs of grid nodes that receive a non-zero weight are the ones belonging to one of the following sets:

$$\begin{aligned}
A_1 &= \{(a_i, b_i) \mid i \in V\}, \\
A_2 &= \{(b_i, b_j) \mid (i, j) \in E\} \cup \{(c_i, d_j) \mid (i, j) \in E\} \cup \{(c_i, d_i) \mid i \in V\}, \\
A_3 &= \{(a_i, d_i) \mid i \in V\}.
\end{aligned}$$

The sets $A_1$ and $A_2$ are depicted in Fig. 5.3 with dashed and solid lines, respectively. Note that $|A_1| = n$ as it contains one pair for each vertex in $V$, $|A_2| = 4|E| + n$ as it contains four pairs for each edge in $E$ and one pair for each vertex in $V$, and, finally, $|A_3| = n$. The pairs in $A_1$ receive a weight equal to $W_1 = 1$, the pairs in $A_2$ receive a weight equal to $W_2 = 3n + 1$ and the pairs in $A_3$ receive a weight equal to $W_3 = 1$.

For the bounds $\alpha$ and $\beta$ take

$$\alpha = 2W_1|A_1| + W_2|A_2| + 3|A_3|W_3 = (3n+1)(4|E|+n) + 5n \tag{5.15}$$

$$\beta = |A_2| + n + k = 4|E| + 2n + k. \tag{5.16}$$

The above construction can take place in polynomial time in $n$. Moreover, if the graph $G$ has a dominating set of size no more than $k$, one can construct a deterministic set of shortcut edges $\mathcal{S}$ that satisfies the constraints of DetSWND-D.

**Lemma 5.4.** *If the instance of* DominatingSet *is a "yes" instance, then the constructed instance of* DetSWND-D *is also a "yes" instance.*

*Proof.* To see this, suppose that there exists a dominating set $A$ of the graph with size $|A| \leq k$. Then, for every $i \in V \setminus A$, there exists a $j \in A$ such that $i \in \Gamma(j)$, *i.e.*, $i$ is a neighbor of $j$. We construct $\mathcal{S}$ as follows. For every $i \in A$, add the edges $(a_i, b_i)$ and $(b_i, c_i)$ in $\mathcal{S}$. For every $i \in V \setminus A$, add an edge $(a_i, b_j)$ in $\mathcal{S}$, where $j$ is such that $j \in A$ and $i \in \Gamma(j)$. For every pair in $A_2$, add this edge in $\mathcal{S}$. The size of $\mathcal{S}$ is

$$|\mathcal{S}| = 2|A| + (|V| - |A|) + |A_2| = |A| + n + 4|E| + n \leq 4|E| + 2n + k.$$

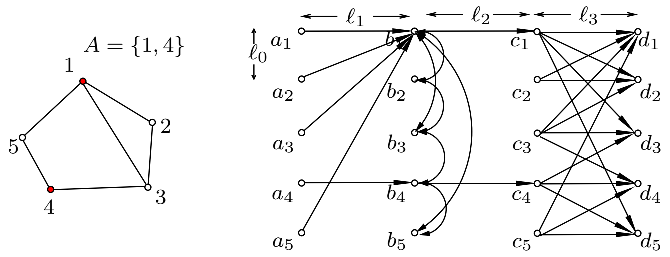


Figure 5.4 – A “yes” instance of DOMINATINGSET and the corresponding “yes” instance of DETSWND-D. The graph on the left can be dominated by two nodes, 1 and 4. The corresponding set $\mathcal{S}$ of shortcut contacts that satisfies the constraints of DETSWND-D is depicted on the right.

Moreover, the weighted forwarding distance is

$$\bar{C}^w_{\mathcal{S}} = \sum_{(i,j)\in A_1} W_1 C_{\mathcal{S}}(i,j) + \sum_{(i,j)\in A_2} W_2 C_{\mathcal{S}}(i,j) + \sum_{(i,j)\in A_3} W_3 C_{\mathcal{S}}(i,j).$$

We have

$$\sum_{(i,j)\in A_2} W_2 C_{\mathcal{S}}(i,j) = W_2 |A_2|$$

as every pair in $A_2$ is connected by an edge in $\mathcal{S}$. Consider now a pair $a_i, b_i) \in A_1$, $i \in V$. There is exactly one edge in $\mathcal{S}$ departing from $a_i$ which has the form $(a_i, b_j)$, where where either $j = i$ is or $j$ a neighbor of $i$. The distance of the closest local neighbor of $a_i$ from $b_i$ is $\ell_1 - 1$. The distance of $b_j$ from $b_i$ is at most $n \cdot \ell_0$. As $\ell_1 - 1 = n\ell_0 + 2 - 1 > n\ell_0$ greedy forwarding will follow $(a_i, b_j)$. If $b_j = b_i$, then $C_{\mathcal{S}}(a_i, b_i) = 1$. If $b_j \neq b_i$, as $j$ is a neighbor of $i$, $\mathcal{S}$ contains the edge $(b_j, b_i)$. Hence, if $b_j \neq b_i$, $C_{\mathcal{S}}(a_i, b_i) = 2$. As $i$ was arbitrary, we get that

$$\sum_{(i,j)\in A_1} W_1 C_{\mathcal{S}}(i,j) \leq 2W_1 n.$$

Next, consider a pair $(a_i, d_i) \in A_3$. For the same reasons as for the pair $(a_i, b_i)$, the shortcut edge $(a_i, b_j)$ in $\mathcal{S}$ will be used by the greedy forwarding algorithm. In particular, the distance of the closest local neighbor of $a_i$ from $d_i$ is $\ell_1 + \ell_2 + \ell_3 - 1$ and $d(b_j, d_i)$ is at most $\ell_2 + \ell_3 + n \cdot \ell_0$. As $\ell_1 - 1 > n\ell_0$, greedy forwarding will follow $(a_i, b_j)$.

By the construction of $\mathcal{S}$, $b_j$ is such that $j \in A$. As a result, again by the construction of $\mathcal{S}$, $(b_j, c_j) \in \mathcal{S}$. The closest local neighbor of $b_j$ to $d_i$ has $\ell_2 + \ell_3 + d(b_j, b_i) - 1$ Manhattan distance from $d_j$. Any shortcut neighbor $b_k$ of $b_j$ has at least $\ell_2 + \ell_3$ Manhattan distance from $b_i$. On the other hand, $c_j$ has $\ell_3 + d(b_j, b_i)$ Manhattan distance from $d_i$. As $\ell_2 > 1$ and $\ell_2 > n\ell_0 \geq d(b_j, b_i)$, the greedy forwarding algorithm will follow $(b_j, c_j)$. Finally, as $A_2 \subset \mathcal{S}$, and $j = i$ or $j$ is a neighbor of $i$, the edge $(c_j, d_i)$ will be in $\mathcal{S}$. Hence, the greedy forwarding algorithm will reach $d_j$ in exactly 3 steps. As $i \in V$ was arbitrary, we get that

$$\sum_{(i,j)\in A_3} W_3 C_{\mathcal{S}}(i,j) = 3W_3 n.$$

Hence,

$$\bar{C}^w_{\mathcal{S}} \leq 2W_1 n + W_2 |A_2| + 3W_3 n = \alpha$$

and, therefore, the instance of DETSWND-D is a “yes” instance. □

To complete the proof, we show that a dominating set of size $k$ exists only if there exists a $\mathcal{S}$ that satisfies the constraints in constucted instance of DetSWND-D.

**Lemma 5.5.** *If the constucted instance of* DetSWND-D *is a "yes" instance, then the instance of* DominatingSet *is also a "yes" instance.*

*Proof.* Assume that there exists a set $\mathcal{S}$, with $|\mathcal{S}| \leq \beta$ such that the augmented graph has a weighted forwarding distance less than or equal to $\alpha$. Then

$$A_2 \subseteq \mathcal{S}. \tag{5.17}$$

To see this, suppose that $A_2 \not\subseteq \mathcal{S}$. Then, there is at least one pair of nodes $(i, j)$ in $A_2$ with $C_{\mathcal{S}}(i, j) \geq 2$. Therefore,

$$\begin{aligned}\bar{C}^w_{\mathcal{S}} &\geq 1 \cdot W_1|A_1| + [(|A_2| - 1) \cdot 1 + 2] \cdot W_2 + 1 \cdot W_3|A_3| \\ &= (3n + 1)(4|E| + n) + 5n + 1 > \alpha,\end{aligned}$$

a contradiction.

Essentially, by choosing $W_2$ to be large, we enforce that all "demands" in $A_2$ are satisfied by a direct edge in $\mathcal{S}$. The next lemma shows a similar result for $A_1$. Using shortcut edges to satisfy these "demands" is enforced by making the distance $\ell_1$ very large.

**Lemma 5.6.** *For every $i \in V$, there exists at least one shortcut edge in $\mathcal{S}$ whose origin is in the same row as $a_i$ and in a column to the left of $b_i$. Moreover, this edge is used during the greedy forwarding of a message from $a_i$ to $b_i$.*

*Proof.* Suppose not. Then, there exists an $i \in V$ such that no shortcut edge has its origin between $a_i$ and $b_i$, or such an edge exists but is not used by the greedy forwarding from $a_i$ to $b_i$ (*e.g.*, because it points too far from $b_i$). Then, the greedy forwarding from $a_i$ to $b_i$ will use only local edges and, hence, $C_{\mathcal{S}}(a_i, b_i) = \ell_1$. We thus have that

$$\bar{C}^w_{\mathcal{S}} \geq \ell_1 + 2n - 1 + W_2|A_2| \overset{(5.12)}{=} 6n^2 + 5n + 1 + W_2|A_2|$$

On the other hand, by (5.15) $\alpha = 5n + W_2|A_2|$ so $\bar{C}^w_{\mathcal{S}} > \alpha$, a contradiction. □

Let $S_1$ be the set of all edges whose origin is between some $a_i$ and $b_i$, $i \in V$, and that are used during forwarding from this $a_i$ to $b_i$. Note that Lemma 5.6 implies that $|S_1| \geq n$. The target of any edge in $S_1$ must lie to the left of the $2\ell_1 + 1$-th column of the grid This is because the Manhattan distance of $a_i$ to $b_i$ is $\ell_1$, so its left local neighbor lies at $\ell_1 - 1$ steps from $b_i$. Greedy forwarding is monotone, so the Manhattan distance from $b_i$ of any target of an edge followed subsequently to route towards $b_i$ must be less than $\ell_1$.

Essentially, all edges in $S_1$ must point close enough to $b_i$, otherwise they would not be used in greedy forwarding. This implies that, to forward the "demands" in $A_3$ an *additional* set of shortcut edges need to be used.

**Lemma 5.7.** *For every $i \in V$, there exists at least one shortcut edge in $\mathcal{S}$ that is used when forwarding a message from $a_i$ to $d_i$ that is neither in $S_1$ nor in $A_2$.*

*Proof.* Suppose not. We established above that the target of any edge in $S_1$ is to the left of the $2\ell_1{+}1$ column. Recall that $A_2 = \{(b_i, b_j) \mid (i,j) \in E\} \cup \{(c_i, d_j) \mid (i,j) \in E\} \cup \{(c_i, d_i) \mid i \in V\}$. By the definition of $b_i$, $i \in V$, the targets of the edges in $\{(b_i, b_j) \mid (i,j) \in E\}$ lie on the $(\ell_1 + 1)$-th column. Similarly, the origins of the edges in $\{(c_i, d_j) \mid (i,j) \in E\} \cup \{(c_i, d_i) \mid i \in V\}$ lie on the $\ell_1 + \ell_2 + 1$-th column. As a result, if the lemma does not hold, there is a demand in $A_3$, say $(a_i, d_i)$, that does not use any additional shortcut edges. This means that the distance between the $2\ell + 1$ and the $\ell_1 + \ell_2 + 1$-th column is traversed by using local edges. Hence, $C_{\mathcal{S}}(a_i, d_i) \geq \ell_2 - \ell_1 + 1$ as at least one additional step is needed to get to the $2\ell_1 + 1$-th column from $a_i$. This implies that

$$\bar{C}^w_{\mathcal{S}} \geq = 2n + W_2|A_2| + \ell_2 - \ell_1 \overset{(5.13)}{=} W_2|A_2| + 5n + 1 > \alpha,$$

a contradiction. □

Let $S_3 = \mathcal{S} \setminus (S_1 \cup A_2)$. Lemma 5.7 implies that $S_3$ is non-empty, while (5.17) and Lemma 5.6, along with the fact that $|\mathcal{S}| \leq \beta = |A_2| + n + k$, imply that $|S_3| \leq k$. The following lemma states that some of these edges must have targets that are close enough to the destinations $d_i$.

**Lemma 5.8.** *For each $i \in V$, there exists an edge in $S_3$ whose target is within Manhattan distance $3n + 1$ of either $d_i$ or $c_j$, where $(c_j, d_i) \in A_2$. Moreover, this edge is used for forwarding a message from $a_i$ to $d_i$ with greedy forwarding.*

*Proof.* Suppose not. Then there exists an $i$ for which greedy forwarding from $a_i$ to $d_i$ does not employ any edge fitting the description in the lemma. Then, the destination $d_i$ can not be reached by a shortcut edge in either $S_3$ or $A_1$ whose target is closer than $3n + 1$ steps. Thus, $d_i$ is reached in one of the two following ways: either $3n + 1$ steps are required in reaching it, through forwarding over local edges, or an edge $(c_j, d_i)$ in $A_2$ is used to reach it. In the latter case, reaching $c_i$ also requires at least $3n + 1$ steps of local forwarding, as no edge in $A_2$ or $S_3$ has an target within $3n$ steps from it, and any edge in $S_1$ that may be this close is not used (by the hypothesis). As a result, $C_{\mathcal{S}}(a_i, d_i) \geq 3n + 2$ as at least one additional step is required in reaching the ball of radius $3n$ centered around $d_i$ or $c_i$ from $a_i$. This gives $\bar{C}^w_{\mathcal{S}} \geq 5n + W_2|A_2| + 1 > \alpha$, a contradiction. □

When forwarding from $a_i$ to $d_i$, $i \in V$, there may be more than one edges in $S_3$ fitting the description in Lemma 5.8. For each $i \in V$, consider the *last* of all these edges. Denote the resulting subset by $S_3'$. By definition, $|S_3'| \leq |S_3| \leq k$. For each $i$, there exists *exactly one* edge in $S_3'$ that is used to forward a message from $a_i$ to $d_i$. Moreover, recall that $\ell_0 = \ell_3 = 6n + 3$. Therefore, the Manhattan distance between any two nodes in $\{c_1, \ldots, c_n\} \cup \{d_1, \ldots, d_n\}$ is $2(3n + 1) + 1$. As a result, the targets of the edges in $S_3'$ will be within distance $3n + 1$ of *exactly one* of the nodes in the above set.

Let $A \subset V$ be the set of all vertices $i \in V$ such that the unique edge in $S_3'$ used in forwarding from $a_i$ to $d_i$ has an target within distance $3n + 1$ of either $c_i$ or $d_i$. Then $A$ is a dominating set of $G$, and $|A| \leq k$. To see this, note first that $|A| \leq k$ because each target of an edge in $S_3'$ can be within distance $3n + 1$ of only one of the nodes in $\{c_1, \ldots, c_n\} \cup \{d_1, \ldots, d_n\}$, and there are at most $k$ edges in $S_3'$.

To see that $A$ dominates the graph $G$, suppose that $j \in V \setminus A$. Then, by Lemma 5.8, the edge in $S_3'$ corresponding to $i$ is either pointing within distance $3n + 1$ of either $d_j$ or a $c_i$ such that $(c_i, d_j) \in A_2$. By the construction of $A$, it cannot point in the proximity of $d_j$, because then $j \in A$, a contradiction. Similarly, it cannot point in the proximity of

$c_j$, because then, again, $j \in A$, a contradiction. Therefore, it points in the proximity of some $c_i$, where $i \neq j$ and $(c_i, d_j) \in A_2$. By the construction of $A$, $i \in A$. Moreover, by the definition of $A_2$, $(c_i, d_j) \in A_2$ if and only if $(i, j) \in E$. Therefore, $j \in \Gamma(A)$. As $j$ was arbitrary, $A$ is a dominating set of $G$. □

### 5.5.2 Proof of Theorem 5.3

According to (5.10), the probability that object $x$ links to $y$ is given by $\ell_x(y) = \frac{1}{Z_x} \frac{\mu(y)}{r_x(y)}$, where $Z_x = \sum_{y \in \mathcal{T}} \frac{\mu(y)}{r_x(y)}$ is a normalization factor bounded as follows.

**Lemma 5.9.** *For any $x \in \mathcal{N}$, let $x^* \in \min_{\preccurlyeq_x} \mathcal{T}$ be any object in $\mathcal{T}$ among the closest targets to $x$. Then $Z_x \leq 1 + \ln(1/\mu(x^*)) \leq 3H_{\max}$.*

*Proof.* Sort the target set $\mathcal{T}$ from the closest to furthest object from $x$ and index objects in an increasing sequence $i = 1, \ldots, k$, so the objects at the same distance from $x$ receive the same index. Let $A_i$, $i = 1, \ldots, k$, be the set containing objects indexed by $i$, and let $\mu_i = \mu(A_i)$ and $\mu_0 = \mu(x)$. Furthermore, let $Q_i = \sum_{j=0}^{i} \mu_j$. Then $Z_x = \sum_{i=1}^{k} \frac{\mu_i}{Q_i}$.

Define $f_x(r) : \mathbb{R}^+ \to \mathbb{R}$ as

$$f_x(r) = \frac{1}{r} - \mu(x).$$

Clearly, $f_x(\frac{1}{Q_i}) = \sum_{j=1}^{i} \mu_j$, for $i \in \{1, 2 \ldots, k\}$. This means that we can rewrite $Z_x$ as

$$Z_x = \sum_{i=1}^{k} (f_x(1/Q_i) - f_x(1/Q_{i-1}))/Q_i.$$

By reordering the terms involved in the sum above, we get

$$Z_x = f_x(\frac{1}{Q_k})/Q_k + \sum_{i=1}^{k-1} f_x(1/Q_i) \left( \frac{1}{Q_i} - \frac{1}{Q_{i+1}} \right).$$

First note that $Q_k = 1$, and second that since $f_x(r)$ is a decreasing function,

$$Z_x \leq 1 - \mu_0 + \int_{1/Q_k}^{1/Q_1} f_x(r) dr = 1 - \frac{\mu_0}{Q_1} + \ln \frac{1}{Q_1}.$$

This shows that if $\mu_0 = 0$ then $Z_x \leq 1 + \ln \frac{1}{\mu_1}$ or otherwise $Z_x \leq 1 + \ln \frac{1}{\mu_0}$ . □

Given the set $\mathcal{S}$, recall that $C_{\mathcal{S}}(s, t)$ is the number of steps required by the greedy forwarding to reach $t \in \mathcal{N}$ from $s \in \mathcal{N}$. We say that a message at object $v$ is in *phase* $j$ if

$$2^j \mu(t) \leq r_t(v) \leq 2^{j+1} \mu(t).$$

Notice that the number of different phases is at most $\log_2 1/\mu(t)$. We can write $C_{\mathcal{S}}(s, t)$ as

$$C_{\mathcal{S}}(s, t) = X_1 + X_2 + \cdots + X_{\log \frac{1}{\mu(t)}}, \tag{5.18}$$

where $X_j$ are the hops occurring in phase $j$.Assume that $j > 1$, and let

$$I = \left\{ w \in \mathcal{N} : r_t(w) \leq \frac{r_t(v)}{2} \right\}.$$

The probability that $v$ links to an object in the set $I$, and hence moving to phase $j-1$, is

$$\sum_{w\in I}\ell_{v,w} = \frac{1}{Z_v}\sum_{w\in I}\frac{\mu(w)}{r_v(w)}.$$

Let $\mu_t(r) = \mu(B_t(r))$ and $\rho > 0$ be the smallest radius such that $\mu_t(\rho) \geq r_t(v)/2$. Since we assumed that $j > 1$ such a $\rho > 0$ exists. Clearly, for any $r < \rho$ we have $\mu_t(r) < r_t(v)/2$. In particular,

$$\mu_t(\rho/2) < \frac{1}{2}r_t(v). \tag{5.19}$$

On the other hand, since the doubling parameter is $c(\mu)$ we have

$$\mu_t(\rho/2) > \frac{1}{c(\mu)}\mu_t(\rho) \geq \frac{1}{2c(\mu)}r_t(v). \tag{5.20}$$

Therefore, by combining (5.19) and (5.20) we obtain

$$\frac{1}{2c(\mu)}r_t(v) < \mu_t(\rho/2) < \frac{1}{2}r_t(v). \tag{5.21}$$

Let $I_\rho = B_t(\rho)$ be the set of objects within radius $\rho/2$ from $t$. Then $I_\rho \subset I$, so

$$\sum_{w\in I}\ell_{v,w} \geq \frac{1}{Z_v}\sum_{w\in I_\rho}\frac{\mu(w)}{r_v(w)}.$$

By triangle inequality, for any $w \in I_\rho$ and $y$ such that $d(y,v) \leq d(v,w)$ we have

$$d(t,y) \leq d(v,y) + d(v,t) \leq d(w,y) + d(v,t) \leq d(w,t) + 2d(v,t) \leq \frac{5}{2}d(v,t).$$

This means that $r_v(w) \leq \mu_t(\frac{5}{2}d(v,t))$, and consequently, $r_v(w) \leq c^2(\mu)r_t(v)$. Therefore,

$$\sum_{w\in I}\ell_{v,w} \geq \frac{1}{Z_v}\frac{\sum_{w\in I_\rho}\mu(w)}{c^2(\mu)r_t(v)} = \frac{1}{Z_v}\frac{\mu_t(\rho/2)}{c^2(\mu)r_t(v)}.$$

By (5.21), the probability of terminating phase $j$ is uniformly bounded by

$$\sum_{w\in I}\ell_{v,w} \geq \min_v \frac{1}{2c^3(\mu)Z_v} \overset{\text{Lem. 5.9}}{\geq} \frac{1}{6c^3(\mu)H_{\max}(\mu)} \tag{5.22}$$

As a result, the probability of terminating phase $j$ is stochastically dominated by a geometric random variable with the parameter given in (5.22). This is because (a) if the current object does not have a shortcut edge which lies in the set $I$, by Property 5.1, greedy forwarding sends the message to one of the neighbours that is closer to $t$ and (b) shortcut edges are sampled independently across neighbours. Hence, given that $t$ is the target object and $s$ is the source object,

$$\mathbb{E}[X_j|s,t] \leq 6c^3(\mu)H_{\max}(\mu). \tag{5.23}$$

Suppose now that $j = 1$. By the triangle inequality, $B_v(d(v,t)) \subseteq B_t(2d(v,t))$ and $r_v(t) \leq c(\mu)r_t(v)$. Hence,

$$\ell_{v,t} \geq \frac{1}{Z_v}\frac{\mu(t)}{c(\mu)r_t(v)} \geq \frac{1}{2c(\mu)Z_v} \geq \frac{1}{6c(\mu)H_{\max}(\mu)}$$

since object $v$ is in the first phase and thus $\mu(t) \leq r_t(v) \leq 2\mu(t)$. Consequently,

$$\mathbb{E}[X_1|s,t] \leq 6c(\mu)H_{\max}(\mu). \tag{5.24}$$

Combining (5.18), (5.23), (5.24) and using the linearity of expectation, we get

$$\mathbb{E}[C_{\mathcal{S}}(s,t)] \leq 6c^3(\mu)H_{\max}(\mu)\log\frac{1}{\mu(t)}$$

and, thus, $\bar{C}_{\mathcal{S}} \leq 6c^3(\mu)H_{\max}(\mu)H(\mu)$.

# Part III

# What Is It?

·– ·–·· ·· –·–· · –––···   ·––––· ·–– ···· ·· –·–· ····
·–· ––– ·– –··   –·· –––   ··   – ·– –·– ·
–·–· ·– – –––···   ·–– ···· · ·–· ·   –·· –––
–·–– ––– ··–   ·–– ·– –· –   – –––   ––· –––
·– ·–·· ·· –·–· · –––···   ··   –·· ––– –· ·––––· –
–·– –· ––– ·––   –·–· ·– – –––··· – ···· · –·   ·· –
–·· ––– · ··· –· ·––––· –   –– ·– – – · ·–· ······

·– ·–·· ·· –·–· ·   ·· –·
·–– ––– –· –·· · ·–· ·–·· ·– –· –··

# Content Search Through Comparisons

# 6

The problem we study in this chapter[1] is content search through comparisons. In short, a user searching for a target object navigates through a database in the following manner. The user is asked to select the object most similar to her target from a small list of objects. A new object list is then presented to the user based on her earlier selection. This process is repeated until the target is included in the list presented, at which point the search terminates.

Searching through comparisons is a typical example of *exploratory search* [WR09, Mar06], the need for which arises when users are unable to state and submit explicit queries to the database. Exploratory search has several important real-life applications [Rut08]. An often-cited example is navigating through a database of pictures of humans in which subjects are photographed under diverse uncontrolled conditions [TD10, TDDM11]. For example, the pictures may be taken outdoors, from different angles or distances, while the subjects assume different poses, are partially obscured, *etc.* Automated methods may fail to extract meaningful features from such photos, so the database cannot be queried in the traditional fashion. On the contrary, a human searching for a particular person can easily select from a list of pictures the subject most similar to the person she has in mind. Moreover, in many practical cases, images that present similar low-level descriptors may have very different semantic content and high level descriptions, and thus perceived differently by users [SWS$^+$00, LSDJ06].

As described in [DJLW08], users may also be unable to state queries because they are unfamiliar with the search domain. For example, a novice classical music listener may not be able to express that she is, *e.g.*, looking for a fugue or a sonata. She might however identify among samples of different musical pieces the closest to the one she has in mind.

Users may also resort to exploratory search when they do not have a clear target in mind, but are able to identify interesting items when presented with them. For example, a user surfing the blogosphere may not know a priori which post she wishes to read; presenting a list of blog posts and letting the surfer identify which one she likes best can be used to steer the search in the right direction. This problem is sometimes referred to as the *page zero problem* in the literature [FG09].

---

[1] This chapter is the result of a collaboration with S. Ioannidis and L. Massoulié.

A lot of interest has been generated in content-based retrieval techniques by such scenarios from both academia and industry [FISS03, GS04]. Note that in all the above applications, the problem amounts to determining which objects to present to the user in order to find the target object as quickly as possible. Formally, the behavior of a human user can be modeled by a so-called *comparison oracle*: given a target and a choice between two objects, the oracle outputs the one closest to the target. The goal is thus to find a sequence of proposed pairs of objects that leads to the target object with as few oracle queries as possible. This problem was introduced by [GLS08] and has recently received considerable attention (see, for example, [LZ09, TD10, TDDM11, JN11]).

Content search through comparisons is also naturally related to the following problem: given a graph embedded in a metric space, how should one augment this graph by adding edges in order to minimize the expected cost of greedy forwarding over this graph? Recall that this is precisely the problem we looked at in the previous chapter under the name of "small-world network design".

In this chapter, we again consider our problem under the scenario of *heterogeneous demand*. This is very interesting in practice: objects in a database are indeed unlikely to be requested with the same frequency. Ideally, assuming that some objects are more likely to be targets than others, we would like to exploit this information to further reduce the number of queries submitted to the oracle. Our contributions are as follows:

- We show that our mechanism for edge addition in the small-world design problem in Chapter 5 has a natural equivalent in the context of content search through comparisons, and we provide a matching upper bound for the performance of this mechanism.
- We also establish a lower bound on any mechanism solving the content search through comparisons problem.
- Based on these results, we propose an adaptive learning algorithm for content search that, given access only to a comparison oracle, can meet the performance guarantees achieved by the above mechanism.

Our analysis relates the cost of content search to two important properties of the demand distribution, namely its *entropy* and its *doubling constant*. We thus provide performance guarantees in terms of the *bias* of the distribution of targets, captured by the entropy, as well as the *topology* of their embedding, captured by the doubling constant.

## 6.1 Related Work

Content search through comparisons is a special case of nearest neighbour search (NNS), a problem that has been extensively studied [Cla06, IM98]. Our work can be seen as an extension of earlier work [KR02, KL04, Cla06] considering the NNS problem for objects embedded in a metric space. It is also assumed that the embedding has a small intrinsic dimension, an assumption that is supported by many practical studies [ML09, GM07]. In particular, [KL04] introduce navigating nets, a deterministic data structure for supporting NNS in doubling metric spaces. A similar technique was considered by [Cla06] for objects embedded in a space satisfying a certain sphere-packing property, while [KR02] relied on growth restricted metrics; all of the above assumptions have connections to the doubling constant we consider in this chapter. In all of these works, however, the underlying metric space is fully observable by the search mechanism while, in our work, we are restricted to accesses to a comparison oracle. Moreover, in all of the above works the demand over the target objects is assumed to be homogeneous.

NNS with access to a comparison oracle was first introduced by [GLS08], and further explored by [LZ09] and [TD09, TD10]. A considerable advantage of the above works is that the assumption that objects are a-priori embedded in a metric space is removed; rather than requiring that similarity between objects is captured by a distance metric, the above works only assume that any two objects can be ranked in terms of their similarity to any targer by the comparison oracle. To provide performance guarantees on the search cost, [GLS08] introduced a so-called "disorder-constant", capturing the degree to which object rankings violate the triangle inequality. This disorder-constant plays roughly the same role in their analysis as the doubling constant does in ours. Nevertheless, these works also assume homogeneous demand, so our work can be seen as an extension of searching with comparisons to heterogeneity, with the caveat of restricting our analysis to the case where a metric embedding exists.

An additional important distinction between [GLS08, LZ09, TD09, TD10] and our work is the existence of a learning phase, during which explicit questions are placed to the comparison oracle. A data-structure is constructed during this phase, which is subsequently used to answer queries submitted to the database during a "search" phase. The above works establish different tradeoffs between the length of the learning phase, the space complexity of the data structure created, and the cost incurred during searching. In contrast, the learning scheme we consider in Section 6.6 is adaptive, and learning occurs while users search; the drawback lies in that our guarantees on the search cost are asymptotic. Again, the main advantage of our approach lies in dealing with heterogeneity.

The use of interactive methods (*i.e.*, that incorporate human feedback) for content search has a long history in the literature. Arguably, the first oracle considered to model such methods is the so-called membership oracle [Gar72], which allows the search mechanism to ask a user questions of the form "does the target belong to set $A$" (see also our discussion in Section 5.2). [BWS$^+$10] deploy such an interactive method for object classification and evaluate it on the *Animals with attributes* database. A similar approach was used by [GJ93] who formulated shape recognition as a coding problem and applied this approach to handwritten numerals and satellite images. Having access to a membership oracle however is a strong assumption, as humans may not necessarily be able to answer queries of the above type for *any* object set $A$. Moreover, the large number of possible sets makes the cost of designing optimal querying strategies over large datasets prohibitive. In contrast, the comparison oracle model makes a far weaker assumption on human behavior—namely, the ability to compare different objects to the target—and significantly limits the design space, making search mechanisms using comparisons practical even over large datasets.

The first practical scheme for image retrieval, based on the pairwise comparisons between images, was proposed in [CMM$^+$00]. It was then extended by [FG05] and [FG07] to the context of content search. The use of a comparison oracle is not limited only to content retrieval/search. As explained in [JS08, JS11], the individuals' rating scale tends to fluctuate a lot. In addition, ratings scales may vary between people. For these reasons it is more natural to use the pairwise comparisons as the basis for the recommendation systems. The advantages of this approach and the challenges of how to make such a system operational is well described in [FJs11].

The relationship between the small-world network design and content search has been also observed in earlier work [GLS08] and was exploited by [LZ09] in proposing their data structures for content search through comparisons; we further expand on this issue in Section 6.4, as this is an approach we also follow.

## 6.2 Definitions and Notation

Our notation in this chapter has a lot in common with the one in Chapter 5, some of which are summarized in Table 6.1. Whenever in doubt, please look at Section 5.2.

### Objects and Metric Embedding

As in Chapter 5, consider a set of objects $\mathcal{N}$, where $|\mathcal{N}| = n$. We assume that there exists a metric space $(\mathcal{M}, d)$, where $d(x, y)$ denotes the distance between $x, y \in \mathcal{M}$, such that objects in $\mathcal{N}$ are embedded in $(\mathcal{M}, d)$.

The objects in $\mathcal{N}$ may represent, for example, pictures in a database. The metric embedding can be thought of as a mapping of the database entries to a set of features (*e.g.*, the age of person depicted, her hair and eye color, *etc.*). The distance between two objects would then capture how "similar" two objects are w.r.t. these features. In what follows, we will abuse notation and write $\mathcal{N} \subseteq \mathcal{M}$, keeping in mind that there might be a difference between the physical objects (the pictures) and their embedding (the attributes that characterize them).

### Comparison Oracle

A *comparison oracle* [GLS08] is an oracle that, given two objects $x, y$ and a target $t$, returns the closest object to $t$. More formally,

$$\text{Oracle}(x, y, t) = \begin{cases} x & \text{if } x \prec_t y, \\ y & \text{if } x \succcurlyeq_t y. \end{cases} \tag{6.1}$$

Note that a tie $d(x, t) = d(y, t)$ is revealed by two calls $\text{Oracle}(x, y, t)$ and $\text{Oracle}(y, x, t)$.

This oracle basically aims to capture the behavior of human users. A human interested in locating, *e.g.*, a target picture $t$ within the database, may be able to compare other pictures with respect to their similarity to this target but cannot associate a numerical value to this similarity. We thus assume, effectively, that the user can order objects w.r.t. their distance from $t$, but does not need to disclose (or even know) the exact values of these distances.

It is important to note here that although we write $\text{Oracle}(x, y, t)$ to stress that a query always takes place with respect to some target $t$, in practice the target is hidden and only known to the oracle. Alternatively, following the "oracle as human" analogy, the human user has a target in mind and uses it to compare the two objects, but never discloses it until actually being presented with it.

### Demand and Entropy

We will consider a probability distribution $\mu$ over the set of objects in $\mathcal{N}$ which we will call the *demand*. In other words, $\mu$ will be a non-negative function such that $\sum_{t \in \mathcal{N}} \mu(t) = 1$. In general, the demand can be *heterogeneous* as $\mu(t)$ may vary across different targets.

Table 6.1 – Summary of Notation Borrowed From Chapter 5.

| | | | |
|---|---|---|---|
| $\mathcal{N}$ | set of objects | $d(x,y)$ | distance between $x, y \in \mathcal{M}$ |
| $(\mathcal{M}, d)$ | metric space | $\nu$ | starting point distribution |
| $H(\mu)$ | entropy of $\mu$ | $\mu$ | demand distribution |
| $\mathcal{T}$ | target set | $x \sim_z y$ | $x$ and $y$ at same distance from $z$ |
| $H_{\max}(\mu)$ | max-entropy of $\mu$ | $B_x(r)$ | ball of radius $r$ centered at $x$ |
| $c(\mu)$ | doubling constant of $\mu$ | $x \preccurlyeq_z y$ | ordering w.r.t. distance from $z$ |

Recall that the *entropy* and the *max-entropy* of $\mu$ are defined as $H(\mu) = \sum_{x \in \mathsf{supp}(\mu)} \mu(x) \log \frac{1}{\mu(x)}$ and $H_{\max}(\mu) = \max_{x \in \mathsf{supp}(\mu)} \log \frac{1}{\mu(x)}$, respectively, where $\mathcal{T} = \mathsf{supp}(\mu)$ is the support of $\mu$.

The entropy has strong connections with the content search problem. More specifically, suppose that we have access to a so-called *membership oracle* [CT91] that can answer queries of the following form:

"Given a target $t$ and a subset $A \subseteq \mathcal{N}$, does $t$ belong to $A$?"

Assume now that an object $t$ is selected according to a distribution $\mu$. It is well known that to find a target $t$ one needs to submit at least $H(\mu)$ queries, on average, to the oracle described above (see [CT91, chap. 2]). Moreover, there exists an algorithm (Huffman coding) that finds the target with only $H(\mu)+1$ queries on average [CT91]. In the worst case, which occurs when the target is the least frequently selected object, the algorithm requires $H_{\max}(\mu)+1$ queries to identify $t$. This problem is also known as the game of twenty questions.

Our work identifies similar bounds assuming that one only has access to a comparison oracle, like the one described by (6.1). Not surprisingly, the entropy of the demand distribution $H(\mu)$ shows up in the performance bounds that we obtain (Theorems 6.1 and 6.2). However, searching for an object will depend not only on the entropy of the demand distribution, but also on the topology of the target set $\mathcal{T}$. This will be captured by the doubling constant of $\mu$ (for the precise definition, look at Section 5.2).

## 6.3 Problem Statement

For the content search problem, we consider the object set $\mathcal{N}$, embedded in $(\mathcal{M}, d)$. Although this embedding exists, we are constrained by not being able to directly compute object distances. Instead, we only have access to a comparison oracle, like the one defined in Section 6.2.

Given access to a comparison oracle, we would like to navigate through $\mathcal{N}$ until we find a target object. In particular, we define *greedy content search* as follows. Let $t$ be the target object and $s$ some object that serves as a starting (or source) point. The greedy content search algorithm proposes an object $w$ and asks the oracle to select, between $s$ and $w$, the object closest to the target $t$, *i.e.*, it evokes Oracle$(s, w, t)$. This process is repeated until the oracle returns something other than $s$, *i.e.*, the proposed object is "more similar" to the target $t$. Once this happens, say at the proposal of some $w'$, if $w' \neq t$, the greedy content search repeats the same process now from $w'$. If at any point the proposed object is $t$, the process terminates.

Recall that in the "oracle as a human" analogy the human cannot reveal $t$ before actually being presented with it. We similarly assume here that $t$ is never "revealed" before actually being presented to the oracle. Though we write $\text{Oracle}(x, y, t)$ to stress that the submitted query is w.r.t. proximity to $t$, the target $t$ is not a priori known. In particular, as we see below, the decision of which objects $x$ and $y$ to present to the oracle cannot directly depend on $t$.

More formally, let $x_k, y_k$ be the $k$-th pair of objects submitted to the oracle: $x_k$ is the *current object*, which greedy content search is trying to improve upon, and $y_k$ is the *proposed object*, submitted to the oracle for comparison with $x_k$. Let

$$o_k = \text{Oracle}(x_k, y_k, t) \in \{x_k, y_k\}$$

be the oracle's response, and define

$$\mathcal{H}_k = \{(x_i, y_i, o_i)\}_{i=1}^{k}, \quad k = 1, 2, \ldots$$

to be the sequence of the first $k$ inputs given to the oracle, as well as the responses obtained; $\mathcal{H}_k$ is the "history" of the content search up to and including the $k$-th access to the oracle.

The starting object is always one of the first two objects submitted to the oracle, *i.e.*, $x_1 = s$. Moreover, in greedy content search,

$$x_{k+1} = o_k, \quad k = 1, 2, \ldots$$

*i.e.*, the current object is always the closest to the target among the ones submitted so far.

On the other hand, the selection of the proposed object $y_{k+1}$ will be determined by the history $\mathcal{H}_k$ and the object $x_k$. In particular, given $\mathcal{H}_k$ and the current object $x_k$ there exists a mapping $(\mathcal{H}_k, x_k) \mapsto \mathcal{F}(\mathcal{H}_k, x_k) \in \mathcal{N}$ such that

$$y_{k+1} = \mathcal{F}(\mathcal{H}_k, x_k), \quad k = 0, 1, \ldots,$$

where here we take $x_0 = s \in \mathcal{N}$ (the starting object) and $\mathcal{H}_0 = \emptyset$ (*i.e.*, before any comparison takes place, there is no history).

We will call the mapping $\mathcal{F}$ the *selection policy* of the greedy content search. In general, we will allow the selection policy to be randomized; in this case, the object returned by $\mathcal{F}(\mathcal{H}_k, x_k)$ will be a random variable, whose distribution

$$\mathbb{P}(\mathcal{F}(\mathcal{H}_k, x_k) = w), \quad w \in \mathcal{N}, \tag{6.2}$$

is fully determined by $(\mathcal{H}_k, x_k)$. Observe that $\mathcal{F}$ depends on the target $t$ only indirectly, through $\mathcal{H}_k$ and $x_k$; this is consistent with our assumption that $t$ is only "revealed" when it is eventually located.

We will say that a selection policy is *memoryless* if it depends on $x_k$ but not on the history $\mathcal{H}_k$. In other words, the distribution (6.2) is the same when $x_k = x \in \mathcal{N}$, irrespectively of the comparisons performed prior to reaching $x_k$.

Assuming that when $x_k = t$, the search effectively terminates (*i.e.*, the human reveals that this is indeed the target), our goal is to select $\mathcal{F}$ so that we minimize the number of

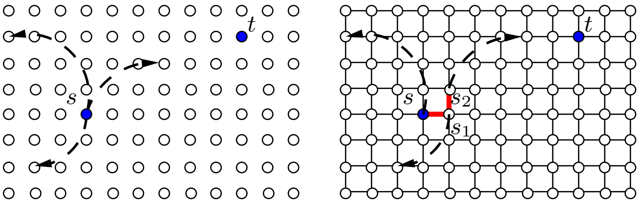


Figure 6.1 – An illustration of the relationship between 1-SWND and CTSC. In CTSC, the source $s$ samples objects independently from the same distribution until it locates an object closest to the target $t$. In 1-SWND, the re-sampling is emulated by the movement to new neighbors. Each neighbor "samples" a new object independently, from a slightly perturbed distribution, until one closest to the target $t$ is found.

accesses to the oracle. In particular, given a target $t$ and a selection policy $\mathcal{F}$, we define the search cost

$$C_{\mathcal{F}}(t) = \inf\{k : x_k = t\}$$

to be the number of proposals to the oracle until $t$ is found. This is a random variable, as $\mathcal{F}$ is randomized; let $\mathbb{E}[C_{\mathcal{F}}(t)]$ be its expectation. The Content Search Through Comparisons problem is then defined as follows:

> Content Search Through Comparisons (CSTC): Given an embedding of $\mathcal{N}$ into $(\mathcal{M}, d)$ and a demand distribution $\mu(t)$, select $\mathcal{F}$ that minimizes the expected search cost
>
> $$\bar{C}_{\mathcal{F}} = \sum_{t\in\mathcal{N}} \mu(t)\mathbb{E}[C_{\mathcal{F}}(t)].$$

Note that, as $\mathcal{F}$ is randomized, the free variable in the above optimization problem is the distribution (6.2).

## 6.4 Relationship Between SWND and CSTC

In what follows, we try to give some intuition about how SWND and CSTC are related and why the upper bounds we obtain for these two problems are identical, without resorting to the technical details appearing in our proofs.

Consider the following content selection policy for CTSC:

$$\Pr(\mathcal{F}(x_k) = w) = \ell_{x_k}(w), \qquad \text{for all } w \in \mathcal{N}.$$

In other words, the proposed object at $x_k$ is sampled according to the same distribution as the shortcut edge in 1-SWND (see Section 5.4, equation (5.10)). This selection policy is memoryless as it does not depend on the history $\mathcal{H}_k$ of objects presented to the oracle so far.

A parallel between these two problems can be drawn as follows. Suppose that the same source/target pair $(s, t)$ is given in both problems. In content search, while starting from node $s$, the memoryless selection policy draws independent samples from distribution $\ell_s$ until an object closer to the target than $s$ is found.

In contrast, greedy forwarding in 1-SWND can be described as follows. Since shorcut edges are generated independently, we can assume that they are generated while the

**Algorithm 6.1** Memoryless Content Search

```
Input: Oracle(·,·,t) , demand distribution $\mu$, starting object $s$.
Output: target $t$.
x ← s
while $x \neq t$ do
    Sample $y \in \mathcal{N}$ from the probability distribution
        $\Pr_x(y) \propto \frac{\mu(y)}{\mu(B_x(d(x,y)))}$.    (6.3)
    $x \leftarrow$ Oracle$(x, y, t)$.
end while
```

message is being forwarded. Then, greedy forwarding at the source object can be seen as sampling an object from distribution $\ell_s$, namely, the one incident to its shortcut edge. If this object is not closer to the target than $s$, the message is forwarded to a neigboring node $s_1$ over a local edge of $s$. Node $s_1$ then samples independently a node from distribution $\ell_{s_1}$ this time—the one incident to its shorcut edge.

Suppose that the distributions $\ell_x$ vary only slightly across neighboring nodes. Then, forwarding over local edges corresponds to the independent resampling occuring in the content search problem. Each move to a new neighbor samples a new object (the one incident to its shortcut edge) independently of previous objects but from a slightly perturbed distribution. This is repeated until an object closer to the target $t$ is found, at which point the message moves to a new neighborhood over the shortcut edge.

Effectively, re-sampling is "emulated" in 1-SWND by the movement to new neighbors. This is, of course, an informal argument; we refer the interested reader to the proofs of Theorems 5.3 and Theorem 6.1 for a rigorous statement of the relationship between the two problems.

## 6.5 Main Results

Exploiting an underlying relationship between 1-SWND and CSTC, we can obtain an efficient selection policy for greedy content search. In particular,

**Theorem 6.1.** *The expected search cost of Algorithm 6.1 is bounded by* $\bar{C}_{\mathcal{F}} \leq 6c^3(\mu) \cdot H(\mu) \cdot H_{\max}(\mu)$.

Like Theorem 5.3, Theorem 6.1 characterises the search cost in terms of the doubling constant, the entropy and the max-entropy of $\mu$. This is very appealing, given (a) the relationship between $c(\mu)$ and the topology of the target set and (b) the classic result regarding the entropy and accesses to a membership oracle, as outlined in Section 6.2.

A question arising from Theorem 6.1 is how tight this bound is. Intuitively, we expect that the optimal shortcut set $\mathcal{S}$ and the optimal selection policy $\mathcal{F}$ depend both on the entropy of the demand distribution and on its doubling constant. Our next theorem, whose proof is in Section 6.7.2, establishes that this is the case for $\mathcal{F}$.

**Theorem 6.2.** *For any integer $K$ and $D$, there exists a metric space $(\mathcal{M}, d)$ and a target measure $\mu$ with entropy $H(\mu) = K\log(D)$ and doubling constant $c(\mu) = D$ such that the* *average search cost of any selection policy $\mathcal{F}$ satisfies*

$$\bar{C}_{\mathcal{F}} \geq H(\mu)\frac{c(\mu)-1}{2\log(c(\mu))}. \tag{6.4}$$

Note that the above lower bound is established for search strategies that utilize the entire search history. Hence, it is *not* restricted to memoryless search. By comparing the bound in Theorem 6.1 with the one in Theorem 6.2 we see that it is tight within a $c^2(\mu)\log(c(\mu))H_{\max}$ factor.

## 6.6 Learning Algorithm

Section 6.5 established bounds on the cost of greedy content search provided that the distribution (5.10) is used to propose items to the oracle. Hence, if the embedding of $\mathcal{N}$ in $(\mathcal{M}, d)$ and demand distribution $\mu$ are known, it is possible to perform greedy content search with the performance guarantees provided by Theorem 6.1.

In this section, we turn our attention to how such bounds can be achieved if *neither* the embedding in $(\mathcal{M}, d)$ *nor* the demand distribution $\mu$ are a priori known. To this end, we propose a novel adaptive algorithm that achieves the performance guarantees of Theorem 6.1 without access to the above information.

Our algorithm effectively learns the ranks $r_x(y)$ of objects and the demand distribution $\mu$ as time progresses. It does not require that distances between objects are at any point disclosed; instead, we assume that it only has access to a comparison oracle, slightly stronger than the one described in Section 5.3.

It is important to note that our algorithm is *adaptive*: though we prove its convergence under a stationary regime, the algorithm can operate in a dynamic environment. For example, new objects can be added to the database while old ones can be removed. Moreover, the popularity of objects can change as time progresses. Provided that such changes happen infrequently, at a larger timescale compared to the timescale in which database queries are submitted, our algorithm will be able to adapt and converge to the desired behavior.

### Demand Model and Probabilistic Oracle

We assume that time is slotted and that at each timeslot $\tau = 0, 1, \ldots$ a new query is generated in the database. We also assume that the starting point of the search and the target of the new query are selected according to a probability distribution $\lambda$ over $\mathcal{N} \times \mathcal{N}$. We denote by $\nu$, $\mu$ the (marginal) distributions, respectively.

Our algorithm will require that the support of both $\nu$, $\mu$ is $\mathcal{N}$, and more precisely that

$$\lambda(x, y) > 0, \text{ for all } x, y \in \mathcal{N}. \tag{6.5}$$

The requirement that the target set $\mathcal{T} = \mathsf{supp}(\mu)$ is $\mathcal{N}$ is necessary to ensure learning; we can only infer the relative order w.r.t. objects $t$ for which questions of the form Oracle$(x, y, t)$ are submitted to the oracle. Moreover, it is natural in our model to assume that the distribution $\nu$ is at the discretion of our algorithm: we can choose which objects to propose first to the user/oracle. In this sense, for a given demand distribution $\mu$ s.t. $\mathsf{supp}(\mu) = \mathcal{N}$, (6.5) can be enforced, *e.g.*, by selecting starting objects uniformly at random from $\mathcal{N}$ and independently of the target.

We consider a slightly weaker oracle than the one described in Section 6.2. In particular, we again assume that

$$\text{Oracle}(x, y, t) = \begin{cases} x & \text{if } x \prec_t y, \\ y & \text{if } x \succ_t y. \end{cases} \tag{6.6}$$

However, we further assume that if $x \sim_t y$, then Oracle$(x, y, t)$ can return either of the two possible outcomes *with non-zero probability.* This is weaker than the oracle in Section 6.2 that correctly identifies $x \sim_t y$.

### Data Structures

For every object $x \in \mathcal{N}$, the database storing $x$ also maintains the following associated data structures. The first data structure is a counter keeping track of how often the object $x$ has been requested so far. The second data structure maintains an order of the objects in $\mathcal{N}$; at any point in time, this total order is an "estimator" of $\preccurlyeq_x$, the order of objects with respect to their distance from $x$. We describe each one of these two data structures in more detail below.

**Estimating the Demand Distribution** The first data structure associated with an object $x$ is an estimator of $\mu(x)$, *i.e.*, the probability with which $x$ is selected as a target. A simple method for keeping track of this information is through a counter $C_x$. This counter $C_x$ is initially set to zero and is incremented every time object $x$ is the target. If $C_x(\tau)$ is the counter at timeslot $\tau$, then

$$\hat{\mu}(x) = C_x(\tau)/\tau \tag{6.7}$$

is an unbiased estimator of $\mu(x)$. To avoid counting to infinity a "moving average" (*e.g.*, and exponentially weighted moving average) could be used instead.

**Maintaining a Partial Order** The second data structure $\mathcal{O}_x$ associated with each $x \in \mathcal{N}$ maintains a total order of objects in $\mathcal{N}$ w.r.t. their similarity to $x$. It supports an operation called `order`() that returns a partition of objects in $\mathcal{N}$ along with a total order over this partition. In particular, the output of $\mathcal{O}_x$.`order`() consists of an ordered sequence of disjoint sets $A_1, A_2, \ldots, A_j$, where $\bigcup A_i = \mathcal{N} \setminus \{x\}$. Intuitively, any two objects in a set $A_i$ are considered to be at equal distance from $x$, while among two objects $u \in A_i$ and $v \in A_j$ with $i < j$ the object $u$ is assumed to be the closer to $x$.

Moreover, every time that the algorithm evokes Oracle$(u, v, x)$, and learns, *e.g.*, that $u \preccurlyeq_x v$, the data structure $\mathcal{O}_x$ should be updated to reflect this information. In particular, if the algorithm has learned so far the order relationships

$$u_1 \preccurlyeq_x v_1, \quad u_2 \preccurlyeq_x v_2, \quad \ldots, \quad u_i \preccurlyeq_x v_i \tag{6.8}$$

$\mathcal{O}_x$.`order`() should return the objects in $\mathcal{N}$ sorted in such a way that all relationships in (6.8) are respected. In particular, object $u_1$ should appear before $v_1$, $u_2$ before $v_2$, and so forth. To that effect, the data structure should also support an operation called $\mathcal{O}_x$.`add`($u$,$v$) that adds the order relationship $u \preccurlyeq_x v$ to the constraints respected by the output of $\mathcal{O}_x$.`order`().

A simple (but not the most efficient) way of implementing this data structure is to represent order relationships through a directed acyclic graph (DAG). Initially, the graph's vertex set is $\mathcal{N}$ and its edge set is empty. Every time an operation `add`$(u,v)$ is executed, an edge is added between vertices $u$ and $v$. If the addition of the new edge creates a cycle then all nodes in the cycle are collapsed to a single node, keeping thus the graph acyclic. Note that the creation of a cycle $u \to v \to \ldots \to w \to u$ implies that $u \sim_x v \sim_x \ldots \sim_x w$, *i.e.*, all these nodes are at equal distance from $x$.

Cycles can be detected by using depth-first search over the DAG [CLRS01]. The sets $A_i$ returned by `order`() are the sets associated with each collapsed node, while a total order among them that respects the constraints implied by the edges in the DAG can be obtained either by depth-first search or by a topological sort [CLRS01]. Hence, the `add`() and `order`() operations have a worst case cost of $\Theta(n+m)$, where $m$ is the total number of edges in the graph.

Several more efficient algorithms exist in literature (see, for example, [HKM$^+$08, PK03, BFG09]), where the best (in terms of performance) proposed by [BFG09] yielding a cost of $O(n)$ for `order`() and an aggregate cost of at most $O(n^2 \log n)$ for any sequence of `add` operations. We stress here that any of these more efficient implementations could be used for our purposes. We refer the reader interested in such implementations to [HKM$^+$08, PK03, BFG09] and, to avoid any ambiguity, we assume the above naïve approach for the remainder of this chapter.

### Greedy Content Search

Our learning algorithm implements greedy content search, as described in Section 6.3, in the following manner. When a new query is submitted to the database, the algorithm first selects a starting object $s$ uniformly at random. It then performs greedy content search using a memoryless selection policy $\hat{\mathcal{F}}$ with distribution $\hat{\ell}_x$, *i.e.*,

$$\Pr(\mathcal{F}(\mathcal{H}_k, x_k) = w) = \hat{\ell}_{x_k}(w) \quad w \in \mathcal{N}. \tag{6.9}$$

Below, we discuss in detail how $\hat{\ell}_x$, $x \in \mathcal{N}$, are computed.

When the current object $x_k$, $k = 0, 1, \ldots$, is equal to $x$, the algorithm evokes $\mathcal{O}_{x_k}$.`order`() and obtains an ordered partition $A_1, A_2, \ldots, A_j$ of items in $\mathcal{N} \setminus \{x\}$. We define

$$\hat{r}_x(w) = \sum_{j=1}^{i:w\in A_i} \hat{\mu}(A_j), \quad w \in \mathcal{N} \setminus \{x\}.$$

This can be seen as an "estimator" of the true rank $r_x$ given by (5.9). The distribution $\hat{\ell}_x$ is then computed as follows:

$$\hat{\ell}_x(w) = \frac{\hat{\mu}(w)}{\hat{r}_x(w)} \frac{1-\epsilon}{\hat{Z}_x} + \frac{\epsilon}{n-1}, \quad i = 1, \ldots, n-1, \tag{6.10}$$

where $\hat{Z}_x = \sum_{w \in \mathcal{N}\setminus\{x\}} \hat{\mu}(w)/\hat{r}_x(w)$ is a normalization factor and $\epsilon > 0$ is a small constant. An alternative view of (6.10) is that the object proposed is selected uniformly at random with probability $\epsilon$, and proportionally to $\hat{\mu}(w_i)/\hat{r}_x(w_i)$ with probability $1-\epsilon$. The use of $\epsilon > 0$ guarantees that every search eventually finds the target $t$.

Upon locating a target $t$, any access to the oracle in the history $\mathcal{H}_k$ can be used to update $\mathcal{O}_t$; in particular, a call Oracle$(u,v,t)$ that returns $u$ implies the constraint $u \preccurlyeq_t v$, which should be added to the data structure through $\mathcal{O}_t$.`add`$(u, v)$. Note that this operation can take place only at the *end of the greedy content search*; the outcomes of calls to the oracle can be observed, but the target $t$ is revealed only after it has been located.

Our main result is that, as $\tau$ tends to infinity, the above algorithm achieves performance guarantees arbitrarily close to the ones of Theorem 6.1. Let $\hat{\mathcal{F}}(\tau)$ be the selection policy defined by (6.9) at timeslot $\tau$ and denote by

$$\bar{C}(\tau) = \sum_{(s,t)\in\mathcal{N}\times\mathcal{N}} \lambda(s,t) \sum_{s\in\mathcal{N}} \mathbb{E}[C_{\hat{F}(\tau)}(s,t)]$$

the expected search cost at timeslot $\tau$. Then the following theorem holds:

**Theorem 6.3.** *Assume that for any two targets $u, v \in \mathcal{N}$, $\lambda(u,v) > 0$.*

$$\limsup_{\tau\to\infty} \bar{C}(\tau) \leq \frac{6c^3(\mu)H(\mu)H_{\max}(\mu)}{(1-\epsilon)}$$

*where $c(\mu)$, $H(\mu)$ and $H_{\max}(\mu)$ are the doubling parameter, the entropy and the max entropy, respectively, of the demand distribution $\mu$.*

The proof of this theorem can be found in Section 6.7.3.

## 6.7 Analysis

### 6.7.1 Proof of Theorem 6.1

The idea of the proof is very similar to the one presented in Section 5.5.2 and follows the same path. Recall that the selection policy is memoryless and determined by

$$\mathbb{P}(\mathcal{F}(\mathcal{H}_k, x_k) = w) = \ell_{x_k}(w).$$

We assume that the desired object is $t$ and the content search starts from $s$. Since there are no local edges, the only way that the greedy search moves from the current object $x_k$ is by proposing an object that is closer to $t$. Like in the SWND case, we are in particular interested in bounding the probability that the rank of the proposed object is roughly half the rank of the current object. This way we can compute how fast we make progress in our search.

As the search moves from $s$ to $t$ we say that the search is in phase $j$ when the rank of the current object $x_k$ is between $2^j\mu(t)$ and $2^{j+1}\mu(t)$. As stated earlier, the greedy search algorithm keeps making comparisons until it finds another object closer to $t$. We can write $C_{\mathcal{F}}(s,t)$ as

$$C_{\mathcal{F}}(s,t) = X_1 + X_2 + \cdots + X_{\log\frac{1}{\mu(t)}},$$

where $X_j$ denotes the number of comparisons done by comparison oracle in phase $j$. Let us consider a particular phase $j$ and denote $I$ the set of objects whose ranks from $t$ are at most $r_t(x_k)/2$. Note that phase $j$ will terminate if the comparison oracle proposes an object from set $I$. The probability that this happens is

$$\sum_{w\in I} \mathbb{P}(\mathcal{F}(\mathcal{H}_k, x_k) = w) = \sum_{w\in I} \ell_{x_k,w}.$$

Note that the sum on the right hand side depends on the distribution of shortcut edges and is independent of local edges. To bound this sum we can use (5.22). Hence, with probability at least $1/(6c^3(\mu)H_{\max}(\mu))$, phase $j$ will terminate. In other words, using the above selection policy, if the current object $x_k$ is in phase $j$, with probability $1/(6c^3(\mu)H_{\max}(\mu))$ the proposed object will be in phase $(j-1)$. This defines a geometric random variable which yields to the fact that on average the number of queries needed to halve the rank is at most $6c(\mu)^3H_{\max}$ or $\mathbb{E}[X_j|s,t] \leq 6c(\mu)^3H_{\max}$. Taking average over the demand $\lambda$, we can conclude that the average number of comparisons is less than $\bar{C}_{\mathcal{F}} \leq 6c^3(\mu)H_{\max}(\mu)H(\mu)$.

## 6.7.2 Proof of Theorem 6.2

Our proof amounts to constructing a metric space and a demand distribution $\mu$ for which the bound holds. Our construction will be as follows. For some integers $D, K$, the target set $\mathcal{N}$ is taken as $\mathcal{N} = \{1, \dots, D\}^K$. The distance $d(x, y)$ between two distinct elements $x, y$ of $\mathcal{N}$ is defined as $d(x, y) = 2^m$, where

$$m = \max\left\{i \in \{1, \dots, K\} : x(K-i) \neq y(K-i)\right\}.$$

We then have the following.

**Lemma 6.4.** *Let $\mu$ be the uniform distribution over $\mathcal{N}$. Then (i) $c(\mu) = D$, and (ii) if the demand distribution is $\mu$, the optimal average search cost $C^*$ based on a comparison oracle satisfies $C^* \geq K\frac{D-1}{2}$.*

Before proving Lemma 6.4, we note that Theorem. 6.2 immediately follows as a corollary.

### Proof of Lemma 6.4

Part (i): Let $x = (x(1), \dots x(K)) \in \mathcal{N}$, and fix $r > 0$. Assume first that $r < 2$; then, the ball $B(x, r)$ contains only $x$, while the ball $B(x, 2r)$ contains either only $x$ if $r < 1$, or precisely those $y \in \mathcal{N}$ such that

$$(y(1), \dots, y(K-1)) = (x(1), \dots, x(K-1))$$

if $r \geq 1$. In the latter case $B(x, 2r)$ contains precisely $D$ elements. Hence, for such $r < 2$, and for the uniform measure on $\mathcal{N}$, the inequality

$$\mu(B(x, 2r)) \leq D\mu(B(x, r)) \tag{6.11}$$

holds, and with equality if in addition $r \geq 1$.

Consider now the case where $r \geq 2$. Let the integer $m \geq 1$ be such that $r \in [2^m, 2^{m+1})$. By definition of the metric $d$ on $\mathcal{N}$, the ball $B(x, r)$ consists of all $y \in \mathcal{N}$ such that

$$(y(1), \dots, y(K-m)) = (x(1), \dots, x(K-m)),$$

and hence contains $D^{\min(K,m)}$ points. Similarly, the ball $B(x, 2r)$ contains $D^{\min(K,m+1)}$ points. Hence (6.11) also holds when $r \geq 2$.

Part (ii): We assume that the comparison oracle, in addition to returning one of the two proposals that is closer to the target, also reveals the distance of the proposal it returns to the target. We further assume that upon selection of the initial search candidate $x_0$, its distance to the unknown target is also revealed. We now establish that the lower bound on $C^*$ holds when this additional information is available; it holds a fortiori for our more resticted comparison oracle.

We decompose the search procedure into phases, depending on the current distance to the destination. Let $L_0$ be the integer such that the initial proposal $x_0$ is at distance $2^{L_0}$ of the target $t$, i.e.,

$$(x_0(1),\dots,x_0(K-L_0))=(t(1),\dots,t(K-L_0)) \quad \& \quad x_0(K-L_0+1)\neq t(K-L_0+1).$$

No information on $t$ can be obtained by submitting proposals $x$ such that $d(x,x_0)\neq 2^{L_0}$. Thus, to be useful, the next proposal $x$ must share its $(K-L_0)$ first components with $x_0$, and differ from $x_0$ in its $(K-L_0+1)$-th entry. Now, keeping track of previous proposals made for which the distance to $t$ remained equal to $2^{L_0}$, the best choice for the next proposal consists in picking it again at distance $2^{L_0}$ from $x_0$, but choosing for its $(K-L_0+1)$-th entry one that has not been proposed so far. It is easy to see that, with this strategy, the number of additional proposals after $x_0$ needed to leave this phase is uniformly distributed on $\{1,\dots D-1\}$, the number of options for the $(K-L_0+1)$-th entry of the target.

A similar argument entails that the number of proposals made in each phase equals 1 plus a uniform random variable on $\{1,\dots,D-1\}$. It remains to control the number of phases. We argue that it admits a Binomial distribution, with parameters $(K,(D-1)/D)$. Indeed, as we make a proposal which takes us into a new phase, no information is available on the next entries of the target, and for each such entry, the new proposal makes a correct guess with probability $1/D$. This yields the announced Binomial distribution for the numbers of phases (when it equals 0, the initial proposal $x_0$ coincided with the target).

Thus the optimal number of search steps $C$ verifies $C\geq\sum_{i=1}^{X}(1+Y_i)$, where the $Y_i$ are i.i.d., uniformly distributed on $\{1,\dots,D-1\}$, and independent of the random variable $X$, which admits a Binomial distribution with parameters $(K,(D-1)/D)$. Thus using Wald's identity, we obtain that $\mathbb{E}[C]\geq\mathbb{E}[X]\mathbb{E}[Y_1]$, which readily implies (ii).

### 6.7.3 Proof of Theorem 6.3

Let $\Delta_\mu=\sup_{x\in\mathcal{N}}|\hat{\mu}(x)-\mu(x)|$. Observe first that, by the weak law of large numbers, for any $\delta>0$

$$\lim_{\tau\to\infty}\Pr(\Delta_\mu>\delta)=0, \tag{6.12}$$

*i.e.*, $\hat{\mu}$ converges to $\mu$ in probability. The lemma below states, for every $t\in\mathcal{N}$, the order data structure $\mathcal{O}_t$ will learn the correct order of any two objects $u,v$ in finite time.

**Lemma 6.5.** *Consider $u,v,t\in\mathcal{N}$ such that $u\preccurlyeq_t v$. Then, the order data structure in $t$ evokes $\mathcal{O}_t$.*`add`*$(u,v)$ after a finite time, with probability one.*

*Proof.* Recall that $\mathcal{O}_t$.***add***$(u,v)$ is evoked if and only if a call Oracle$(u,v,t)$ takes place and it returns $u$. If $u\prec_t v$ then Oracle$(u,v,t)=u$. If, on the other hand, $u\sim_t v$, then Oracle$(u,v,t)$ returns $u$ with non-zero probability. It thus suffices to show that such, for large enough $\tau$, a call Oracle$(u,v,t)$ occurs at timeslot $\tau$ with a non-zero probability. By the hypothesis of Theorem 6.3, $\lambda(u,t)>0$. By (6.10), given that the starting objects is

$u$, the probability that $\hat{\mathcal{F}}(u) = v$ conditioned on $\hat{\mu}$ is

$$\hat{\ell}_u(v) \geq \frac{\mu(v)-\Delta_\mu}{1+(n-1)\Delta_\mu}\frac{1-\epsilon}{n-1}+\frac{\epsilon}{n-1} \geq \frac{\mu(v)-\Delta_\mu}{(1+(n-1)\Delta_\mu)(n-1)}$$

as $\hat{Z}_v \leq n-1$ and $|\hat{\mu}(x)-\mu(x)| \leq \Delta_\mu$ for every $x \in \mathcal{N}$. Thus, for any $\delta > 0$, the probability that is lower-bounded by

$$\lambda(u,t)\Pr(\hat{\mathcal{F}}(u) = v) \geq \frac{\mu(v) - \delta}{1 + (n-1)\delta}\Pr(\Delta_\mu < \delta).$$

By taking $\delta > 0$ smaller than $\mu(v)$, we have by (6.12) that there exists a $\tau^*$ s.t. for all $\tau > \tau^*$ the probability that Oracle$(u, v, t)$ takes place at timeslot $\tau$ is bounded away from zero, and the lemma follows. □

Thus if $t$ is a target then, after a finite time, for any two $u, v \in \mathcal{N}$ the ordered partition $A_1, \ldots, A_j$ returned by $\mathcal{O}_t$.`order`() will respect the relationship between $u, v$. In particular for $u \in A_i$,$v \in A_{i'}$, if $u \sim_t v$ then $i = i'$, while if $u \prec_t v$ then $i < i'$. As a result, the estimated rank of an object $u \in A_i$ w.r.t. $t$ will satisfy

$$\hat{r}_t(u) = \sum_{x\in\mathcal{T}:x\prec_v u}\hat{\mu}(x) + \sum_{x\in\mathcal{N}\setminus\mathcal{T}:x\in A_{i'},i'\leq i}\hat{\mu}(x) = r_t(u) + O(\Delta_\mu)$$

*i.e.* the estimated rank will be close to the true rank, provided that $\Delta_\mu$ is small. Moreover, as in Lemma 5.9, it can be shown that

$$\hat{Z}_v \leq 1+\log^{-1}\hat{\mu}(v) = 1+\log^{-1}[\mu_v + O(\Delta_\mu)]$$

for $v \in \mathcal{N}$. From these, for $\Delta_\mu$ small enough, we have that for $u, v \in \mathcal{N}$,

$$\hat{\ell}_u(v) = [\ell_u(v) + O(\Delta_\mu)](1 - \epsilon) + \epsilon\frac{1}{n-1}.$$

Following the same steps as the proof of Theorem 5.3 we can show that, given that $\Delta_\mu \leq \delta$, the expected search cost is upper bounded by $\frac{6c^3HH_{\max}}{(1-\epsilon)+O(\delta)}$. This gives us that

$$\bar{C}(\tau) \leq \Big[\frac{6c^3HH_{\max}}{(1-\epsilon)} + O(\delta)\Big]\Pr(\Delta_\mu \leq \delta) + \frac{n-1}{\epsilon}\Pr(\Delta_\mu > \delta)$$

where the second part follows from the fact that, by using the uniform distribution with probability $\epsilon$, we ensure that the cost is stochastically upper-bounded by a geometric r.v. with parameter $\frac{\epsilon}{n-1}$. Thus, by (6.12),

$$\limsup_{\tau\to\infty}\bar{C}(\tau) \leq \frac{6c^3HH_{\max}}{(1-\epsilon)} + O(\delta).$$

As this is true for all small enough $\delta$, the theorem follows.

> “ ·– –– ·· –· ·–·–·–   –·· –––
> ··· ––– –– · – ···· ·· –· ––· ······ ”
>
> ·–· ··– –·· ·· ––· · ·–·   ··– ·–· –··· ·– –· –·– ·

# 7

# Comparison-Based Active Learning

In this chapter [1] we consider again the problem of search through comparisons, where a user is presented with two candidate objects and reveals which is closer to her intended target. In contrast to the previous chapter, we study adaptive strategies for finding the target. As we will see, they require knowledge of rank relationships but not actual distances between objects.

We have mentioned in the previous chapter that content search through comparisons in this setup can be framed as an active learning problem. A well known algorithm for active learning is the *Generalized Binary Search* (GBS) or *splitting* algorithm [Das05]. Applying this algorithm to submit queries to the oracle can locate the target within $OPT \cdot \big(H_{\max}(\mu) + 1\big)$ queries, where $OPT$ is the number of queries submitted by an optimal algorithm. In practice, GBS performs very well in terms of query complexity, suggesting that this bound can be tightened. However, the computational complexity of GBS is $\Theta(n^3)$ for $n = |\mathcal{N}|$, which makes it prohibitive for very large databases.

In the previous chapter, we proposed a memoryless algorithm (Algorithm 6.1) that determines the target in $O\big(c^3 H(\mu) H_{\max}(\mu)\big)$ factor from the optimal, whose computational complexity is $O(1)$ per query. We also showed that $OPT = \Omega(cH(\mu))$, indicating that Algorithm 6.1 is within $c^2 H_{\max}(\mu)$ from the optimal. Our contributions in this chapter can be summarized as follows:

- First, we propose a new adaptive algorithm, RankNetSearch, locating the target within $O(c^6 H(\mu))$ queries to the oracle, in expectation. Our algorithm therefore improves on GBS and Algorithm 6.1 by removing the term $H_{\max}$—which can be quite large in practice—at a cost of a higher exponent in the dependence on the constant $c$. Its computational complexity is $O\big(n(\log n + c^6) \log c\big)$ per query, which is manageable compared to GBS; moreover, this cost can be reduced to $O(1)$ by pre-computing an additional data structure.
- Second, we extend RankNetSearch to the case of a *faulty* oracle, that lies with a probability $0 < \epsilon < 0.5$. We show that the algorithm can locate the target with arbitrarily high probability at an expected cost close to $O(\sum_{x \in \mathcal{N}} \mu(x) \log \frac{1}{\mu(x)} \log \log(\mu(x)))$ and, thereby, close to $H(\mu)$.

---

[1] This chapter is the result of a collaboration with S. Ioannidis and L. Massoulié.

– Third, we evaluate extensively RankNetSearch, and prior art algorithms over several datasets. We observe that RankNetSearch establishes a desirable trade-off between query and computational complexity.

## 7.1 Definitions and Preliminaries

Consider a large finite set of objects $\mathcal{N}$ of size $n = |\mathcal{N}|$, endowed with a distance metric $d$, capturing the "dissimilarity" between objects. A user selects a target $t \in \mathcal{N}$ from a prior distribution $\mu$; our goal will be to design an interactive algorithm that queries the user with the purpose of discovering $t$.

**Comparison Oracle.** Like in the previous chapter, we assume that the metric $d$ exists, but our view of distances is constrained to only observing order relationships. More precisely, we only have access to information that can be obtained through a *comparison oracle* [GLS08]. Given an object $z$, a comparison oracle Oracle receives as a query an ordered pair $(x, y) \in \mathcal{N}^2$ and answers the question "is $z$ closer to $x$ than to $y$?", *i.e.*,

$$\text{Oracle}(x, y, z) = \begin{cases} +1 & \text{if } d(x, z) < d(y, z), \\ -1 & \text{if } d(x, z) \geq d(y, z) \end{cases} \tag{7.1}$$

Note that in the above, we used an equivalent definition of the comparison oracle (see Section 6.2) in order to be able to frame the content search through comparisons as an active learning problem. We will first assume that the oracle always gives correct answers; in Section 7.4, we relax this assumption by considering a *faulty* oracle that lies with probability $\epsilon < 0.5$.

**Prior Knowledge and Performance Metrics.** The algorithms we study rely only on a priori knowledge of (a) the distribution $\mu$ and (b) the values of the mapping $\text{Oracle}(\cdot, \cdot, z) : \mathcal{N}^2 \to \{-1, +1\}$, for every $z \in \mathcal{N}$. This is in line with our assumption that, although the distance metric $d$ exists, it cannot be directly observed. Our focus is on *adaptive* algorithms, whose decision on which query in $\mathcal{N}^2$ to submit next are determined by the oracle's previous answers.

As we mentioned in the previous chapter, the prior $\mu$ can be estimated empirically as the frequency with which objects have been targets in the past. The order relationships can be computed off-line by submitting $\Theta(n^2 \log n)$ queries to a comparison oracle, and requiring $\Theta(n^2)$ space: for each possible target $z \in \mathcal{N}$, objects in $\mathcal{N}$ can be sorted w.r.t. their distance from $z$ with $\Theta(n \log n)$ queries to $\text{Oracle}(\cdot, \cdot, z)$.

We store the result of this sorting in (a) a linked list, whose elements are sets of objects at equal distance from $z$, and (b) a hash-map, that associates every element $y$ with its rank in the sorted list. Note that $\text{Oracle}(x, y, z)$ can thus be retrieved in $O(1)$ time by comparing the relative ranks of $x$ and $y$ with respect to their distance from $z$.

We measure the performance of an algorithm through two metrics. The first is the *query complexity* of the algorithm, determined by the expected number of queries the algorithm needs to submit to the oracle to determine the target. The second is the *computational complexity* of the algorithm, determined by the time-complexity of determining the query to submit to the oracle at each step.

## 7.2 Active Learning

Search through comparisons can be seen as a special case of *active learning* [Das05, Now09]. In active learning, a *hypothesis space* $\mathcal{H}$ is a set of binary valued functions defined over a finite set $\mathcal{Q}$, called the *query space*. Each hypothesis $h \in \mathcal{H}$ generates a label from $\{-1, +1\}$ for every query $q \in \mathcal{Q}$. A target hypothesis $h^*$ is sampled from $\mathcal{H}$ according to some prior $\mu$; asking a query $q$ amounts to revealing the value of $h^*(q)$, thereby restricting the possible candidate hypotheses. The goal is to determine $h^*$ in an adaptive fashion, by asking as few queries as possible.

In our setting, the hypothesis space $\mathcal{H}$ is the set $\mathcal{N}$, and the query space $\mathcal{Q}$ is the set of ordered pairs $\mathcal{N}^2$. The target hypothesis sampled from $\mu$ is the unknown target $t$. Each hypothesis/object $z \in \mathcal{N}$ is uniquely[2] identified by the mapping $\text{Oracle}(\cdot, \cdot, z) : \mathcal{N}^2 \to \{-1, +1\}$, which we have assumed to be a priori known.

**Generalized Binary Search** A well-known algorithm for determining the true hypothesis in the general active-learning setting is the so-called *generalized binary search* (GBS) or *splitting* algorithm [Das05, Now09]. Define the *version space* $V \subseteq \mathcal{H}$ to be the set of possible hypotheses that are consistent with the query answers observed so far. At each step, GBS selects the query $q \in \mathcal{Q}$ that minimizes $|\sum_{h \in V} \mu(h)h(q)|$. Put differently, GBS selects the query that separates the current version space into two sets of roughly equal probability mass; this leads, in expectation, to the largest reduction in the mass of the version space as possible, so GBS can be seen as a greedy query selection policy.

A bound on the query complexity of GBS originally obtained by Dasgupta [Das05] and recently tightened (w.r.t. constants) by Golovin and Krause [GK10] is given by the following theorem:

**Theorem 7.1.** *GBS makes at most* $OPT \cdot \big(H_{\max}(\mu) + 1\big)$ *queries in expectation to identify hypothesis* $h^* \in \mathcal{N}$*, were* $OPT$ *is the minimum expected number of queries made by any adaptive policy.*

**GBS in Search through Comparisons.** In our setting, the version space $V$ comprises all possible objects in $z \in \mathcal{N}$ that are consistent with oracle answers given so far. In other words, $z \in V$ iff $\text{Oracle}(x, y, z) = \text{Oracle}(x, y, t)$ for all queries $(x, y)$ submitted to the oracle. Selecting the next query therefore amounts to finding the pair $(x, y) \in \mathcal{N}^2$ that minimizes

$$f(x, y) = \big|\textstyle\sum_{z \in V} \mu(z)\text{Oracle}(x, y, z)\big|. \tag{7.2}$$

As the simulations in Section 7.5 show, the query complexity of GBS is excellent in practice. This suggests that the general bound of Theorem 7.1 could be improved in the specific context of search through comparisons.

Nevertheless, the computational complexity of GBS is $\Theta(n^2|V|)$ operations per query, as it requires minimizing $f(x, y)$ over all pairs in $\mathcal{N}^2$. For large sets $\mathcal{N}$, this can be truly prohibitive. This motivates us to propose a new algorithm, RankNetSearch, whose computational complexity is almost linear and its query complexity is within a $O(c^5(\mu))$ factor from the optimal.

---

[2] Note that, for any two objects/hypotheses $z, z' \in \mathcal{N}$, there exists at least one query in $\mathcal{N}^2$ that differentiates them, namely $(z', z)$.

**Algorithm 7.1** RankNetSearch

```
Input: Oracle Oracle$(\cdot,\cdot,t)$
Output: Target $t$
Let $E \leftarrow \mathcal{N}$; select arbitrary $x \in E$
repeat
    ( $\mathcal{R}$, $\{B_y(r_y)\}_{y\in\mathcal{R}}$ )$\leftarrow$RankNet($x$,$E$)
    Find $y^*$, the object in $\mathcal{R}$ closest to $t$, using Oracle$(\cdot,\cdot,t)$.
    Let $E \leftarrow B_{y^*}(r_{y^*})$ and $x \leftarrow y^*$;
until $E$ is a singleton
return $y$
```

**Algorithm 7.2** RankNet($x$,$E$)

```
Input: Root object $x$, Ball $E = B_x(R)$
Output: $\rho$-rank net $\mathcal{R}$, Voronoi balls $\{B_y(r_y)\}_{y\in\mathcal{R}}$
$\rho \leftarrow 1$
repeat
    $\rho \leftarrow \rho/2$; construct a $\rho$-net $\mathcal{R}$ of $E$
    $\forall\ y \in \mathcal{R}$, construct ball $B_y(r_y)$
    Let $\mathcal{I} \leftarrow \{y \in E : |B_y(r_y)| > 1\}$
until $\mathcal{I} = \emptyset$ or $\max_{y\in\mathcal{I}} \mu(B_y(r_y)) \leq 0.5\mu(E)$
return ($\mathcal{R}$,$\{B_y(r_y)\}_{y\in\mathcal{R}}$)
```

## 7.3 An Efficient Adaptive Algorithm

Our algorithm is inspired by $\epsilon$-nets, a structure introduced by Clarkson [Cla06] in the context of Nearest Neighbor Search (NNS). The main challenge that we face is that, contrary to standard NNS, we have *no access to the underlying distance metric.* In addition, the bounds on the number of comparisons made by $\epsilon$-nets are worst-case (*i.e.*, prior-free); our construction takes the prior $\mu$ into account to provide bounds in expectation.

### 7.3.1 Rank Nets

To address the above issues, we introduce the notion of *rank nets*, which will play the role of $\epsilon$-nets in our setting. For some $x \in \mathcal{N}$, consider the ball $E = B_x(R) \subseteq \mathcal{N}$. For any $y \in E$, we define

$$d_y(\rho, E) = \inf\{r : \mu(B_y(r)) \geq \rho\mu(E)\} \tag{7.3}$$

to be the radius of the smallest ball around $y$ that maintains a mass above $\rho\mu(E)$. Using this definition[3], we define a $\rho$-rank net as follows.

**Definition 7.2.** *For some $\rho < 1$, a $\rho$-rank net of $E = B_x(r) \subseteq \mathcal{N}$ is a maximal collection of points*[4] *$\mathcal{R} \subset E$ such that for any two distinct $y, y' \in \mathcal{R}$*

$$d(y, y') > \min\{d_y(\rho, E), d_{y'}(\rho, E)\}. \tag{7.4}$$

---

[3] Whenever $\rho$ and $E$ are unambiguous, we simply write $d_y$ rather than $d_y(\rho, E)$.

[4] i.e. one to which no more nodes can be added.

For any $y \in \mathcal{R}$, consider the Voronoi cell

$$V_y = \{z \in E : d(y, z) \leq d(y', z), \forall y' \in \mathcal{R}, y' \neq y\}.$$

We also define the radius $r_y$ of the Voronoi cell $V_y$ as

$$r_y = \inf\{r : V_y \subseteq B_y(r)\}.$$

Critically, a rank net and the Voronoi tessellation it defines *can both be computed using only ordering information*:

**Lemma 7.3.** *A $\rho$-rank net $\mathcal{R}$ of $E$ can be constructed in $O(|E|(\log|E| + |\mathcal{R}|))$ steps, and the balls $B_y(r_y) \subset E$ circumscribing the Voronoi cells around $\mathcal{R}$ can be constructed in $O(|E||\mathcal{R}|)$ steps using only (a) $\mu$ and (b) the mappings Oracle$(\cdot, \cdot, z) : \mathcal{N}^2 \to \{-1, +1\}$ for every $z \in E$.*

The proof is in Section 7.6.1. Equipped with this result, we turn our attention to how the selection of $\rho$ affects the size of the net as well as the mass of the Voronoi balls around it. Our next lemma, whose proof is in Section 7.6.2, bounds $|\mathcal{R}|$.

**Lemma 7.4.** *The size of the net $\mathcal{R}$ is at most $c^3/\rho$.*

Finally, our last lemma determines the mass of the Voronoi balls in the net.

**Lemma 7.5.** *If $r_y > 0$ then $\mu(B_y(r_y)) \leq c^3\rho\mu(E)$.*

The proof is in Section 7.6.3. Note that Lemma 7.5 *does not* bound the mass of Voronoi balls of radius zero.

### 7.3.2 Rank Net Data Structure and Algorithm

Rank nets can be used to identify a target $t$ using a comparison oracle Oracle$(\cdot, \cdot, t)$ as described in Algorithm 7.1. Initially, a net $\mathcal{R}$ covering $\mathcal{N}$ is constructed; nodes $y \in \mathcal{R}$ are compared w.r.t. their distance from $t$, and the closest to the target is determined, say $y^*$. Note that this requires submitting $|\mathcal{R}| - 1$ queries to the oracle. The version space $V$ (the set of possible hypotheses) is thus the Voronoi cell $V_{y^*}$, and is a subset of the ball $B_{y^*}(r_{y^*})$. The algorithm then proceeds by limiting the search to $B_{y^*}(r_{y^*})$ and repeating the above process. Note that, at all times, the version space is included in the current ball to be covered by a net. The process terminates when this ball becomes a singleton which, by construction, must contain the target.

An obvious question in the above setup is how to select $\rho$: by Lemma 7.5, small values lead to a sharp decrease in the mass of Voronoi balls from one level to the next, hence reaching the target with fewer iterations. On the other hand, by Lemma 7.4, small values also imply larger nets, leading to more queries to the oracle at each iteration. We select $\rho$ in an iterative fashion, as indicated in the pseudocode of Algorithm 7.2: we repeatedly halve $\rho$ until all non-singleton Voronoi balls $B_y(r_y)$ of the resulting net have a mass bounded by $0.5\mu(E)$.

This selection leads to the following bounds on the corresponding query and computational complexity of RankNetSearch:

**Theorem 7.6.** RankNetSearch *locates the target by making* $4c^6(1+H(\mu))$ *queries to a comparison oracle, in expectation. The cost of determining which query to submit next is* $O\big(n(\log n + c^6)\log c\big)$.

In light of the lower bound on the query complexity of $\Omega(cH(\mu))$ that we established in the previous chapter (Theorem 6.2), RankNetSearch is within a $O(c^5)$ factor of the optimal algorithm in terms of query complexity, and is thus order optimal for constant $c$. Moreover, the computational complexity per query is $O\big(n(\log n + c^6)$, in contrast to the cubic cost of the GBS algorithm. As shown in Section 7.5, in practice, this leads to drastic reductions in the computational complexity compared to GBS.

Note that the above computational cost can if fact be reduced to $O(1)$ through amortization. In particular, it is easy to see that the possible paths followed by RankNetSearch define a hierarchy, whereby every object serves as a parent to the objects covering its Voronoi ball. This tree can be pre-constructed, and search reduces to a descent over this tree; we further study this Section 7.5.

## 7.4 Noisy Comparison Oracle

In a noisy setting the search must be robust against erroneous answers. Specifically, assume that for any query Oracle$(x, y, t)$, the *noisy oracle* returns the wrong answer with a probability bounded by $\epsilon$, for some $0 < \epsilon < 1/2$.

In this context, the problem with RankNetSearch arises in line 4 of Algorithm 7.1: it is not clear how to identify the object closest to the target among elements in a net. We resolve this by introducing repetitions at each iteration. Specifically, at the $\ell$-th step of the search, $\ell \geq 1$, and rank-net size $m$, we define a repetition factor $k_\delta(\ell, m)$ by

$$k_\delta(\ell, m) := \frac{2\log\left((\ell + 1/\delta)^2\lceil\log_2(m)\rceil\right)}{(1-\epsilon)^2} \tag{7.5}$$

for some design parameter $\delta \in (0, 1)$. The modified algorithm then proceeds down the hierarchy, starting at the top level for $\ell = 1$. The basic step at step $\ell$ with a net $\mathcal{R}$ proceeds as follows. A *tournament* is organized among elements of $\mathcal{R}$, who are initially paired. Pairs of competing members are compared $k_\delta(\ell, |\mathcal{R}|)$ times. The "player" from a given pair winning the largest number of games moves to the next stage, where it will be paired again with another winner of the first round, and so forth until only one player is left. Note that the number of repetitions $k_\delta(\ell, m)$ increases only logarithmically with the level $\ell$.

To find the closest object to target $t$ with the noiseless oracle, clearly we need to make $O(|\mathcal{R}|)$ number of queries. The proposed algorithm achieves the same goal with high probability by making at most a factor $2k_\delta(\ell, |\mathcal{R}|)$ more comparisons. In this context we have the following

**Theorem 7.7.** *For a comparison oracle with error probability* $\epsilon$*, the algorithm with repetitions (7.5) outputs the correct target with probability at least* $1-\delta$ *in*

$$\#\ \textit{of queries} = O\left(\frac{1}{(\frac{1}{2}-\epsilon)^2}\sum_{x\in\mathcal{N}}\mu(x)\log\frac{1}{\mu(x)}\log(\frac{1}{\delta} + \log\frac{1}{\mu(x)})\right),$$

*with constants depending on c.*

Table 7.1 – Table of size, dimension (number of features), as well as the size of the Rank Net Tree hierarchy constructed for each dataset.

| Dataset | Size | Dim. | Tree |
|---|---|---|---|
| *iris* | 147 | 7 | 399 |
| *abalone* | 4177 | 11 | 21707 |
| *faces* | 698 | 4096 | 3664 |
| *ad* | 1924 | 1559 | 5816 |
| *swiss roll* | 1000 | 3 | 4229 |
| *netflix* | 1001 | 50 | 14585 |

The proof is given in Section 7.6.5. For the uniform distribution $\mu(x) \equiv 1/n$, for all $x \in \mathcal{N}$, this yields an extra $\log\log(n)$ factor in addition to the term of order $H(\mu) = \log(n)$ which, by Theorem 6.2, is optimal.

## 7.5 Numerical Evaluation

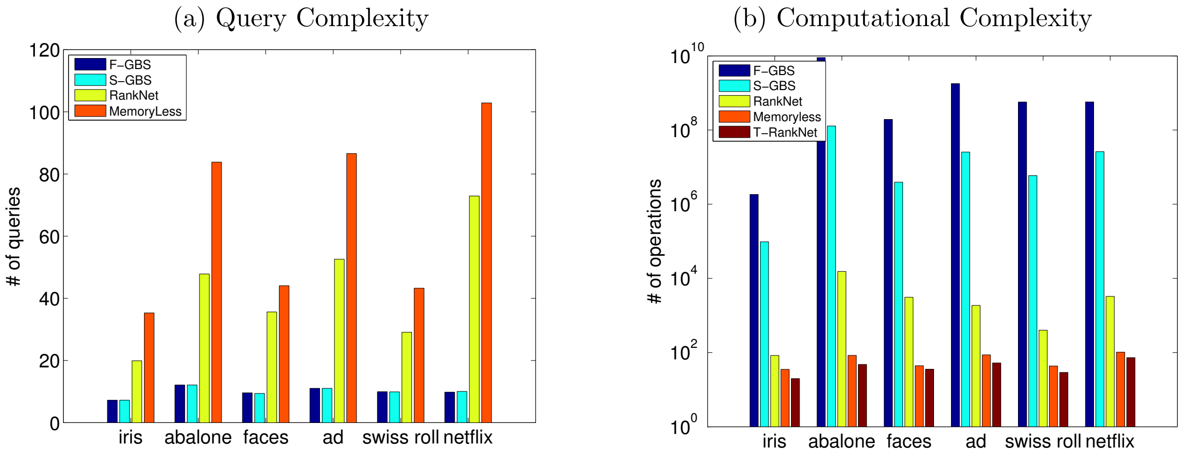
We evaluate RankNetSearch over six publicly available datasets[5]: *iris*, *abalone*, *ad*, *faces*, *swiss roll*, and *netflix*. We subsampled the latter two, taking 1000 randomly selected data points from *swiss roll*, and the 1001 most rated movies in *netflix*. We map these datasets to $\mathbb{R}^d$ (categorical variables are mapped to binary values in the standard fashion) for $d$ as shown in Tabel 7.1. For *netflix*, movies were mapped to 50-dimensional vectors by obtaining a low rank approximation of the user/movie rating matrix through SVD. For all our experiments, we use $\ell_2$ distance and select targets from a power-law prior with $\alpha = 0.4$.

We evaluated the performance of two versions of RankNetSearch: one as described by Algorithm 7.1, and another one (T-RankNetSearch) in which the hierarchy of rank nets is precomputed and stored as a tree. Both propose exactly the same queries to the oracle, so have the same query complexity; however, T-RankNetSearch has only $O(1)$ computational cost per query. The sizes of the trees precomputed by T-RankNetSearch for each dataset are shown in Tabel 7.1.

We compare these algorithms to (a) the memoryless policy proposed in the previous chapter (Algorithm 6.1) and (b) two heuristics based on GBS (the $\Theta(n^3)$ computational cost of GBS per query makes it intractable over the datasets we consider). The first heuristic, termed F-GBS for *fast* GBS, selects like GBS the query that minimizes (7.2); however, it does so by restricting the queries to pairs of objects in the current version space $V$. This reduces the computational cost per query to $\Theta(|V|^3)$, rather than $\Theta(n^2|V|)$. The second heuristic, termed S-GBS for *sparse* GBS, exploits rank nets in the following way. First, we costruct the rank net hierarchy over the dataset, as in T-RankNetSeach. Then, in minimizing (7.2), we restrict queries *only* to pairs of objects that appear in the same net. Intuitively, S-GBS assumes that an equitable partition of the objects exists among such pairs.

---

[5] Sources: `www.netflixprize.com`,
`http://isomap.stanford.edu/datasets.html`,
`http://archive.ics.uci.edu/ml`.


(a) Query Complexity

(b) Computational Complexity

Figure 7.1 – (a) Expected query complexity, per search, of the five algorithms applied on each data set. As RankNet and T-RankNet have the same query complexity, only one is shown. (b) Expected computational complexity, per search, of the five algorithms applied on each dataset. For Memoryless and T-RankNet this equals the query complexity.

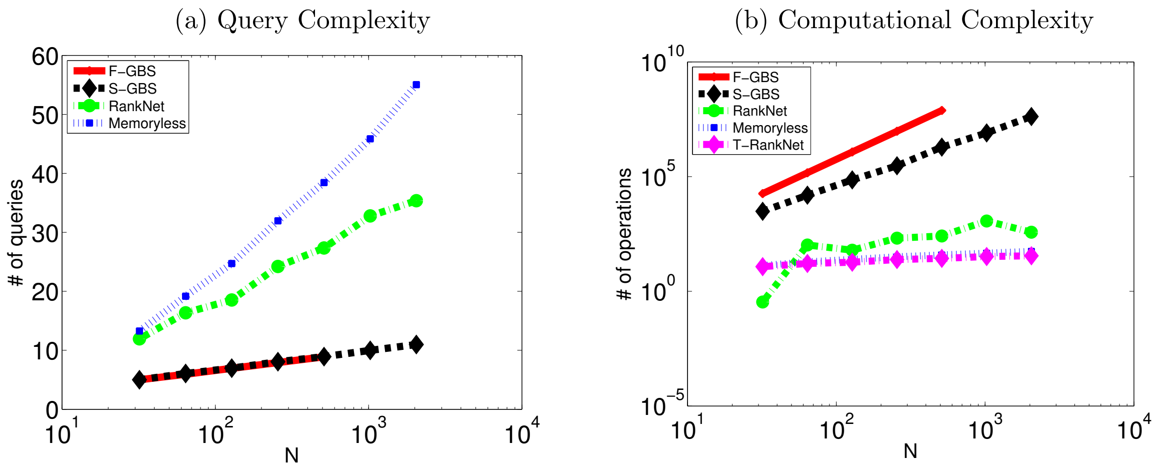
### Query versus Computational Complexity

The query complexity of different algorithms, expressed as average number of queries per search, is shown in Fig. 7.1(a). Although there are no known guarantees for either F-GBS nor S-GBS both algorithms are excellent in terms of query complexity across all datasets, finding the target within about 10 queries, in expectation. As GBS should perform as well as these algorithms, these suggest that it should also have low query cost. The query complexity of RankNetSearch between 2 to 10 times higher; the impact is greater for high-dimensional datasets, as expected through the dependence of the rank net size on the $c$ doubling constant. Finally, Memoryless performs worse compared to all other algorithms. As shown in Fig. 7.1, the above ordering is fully reversed w.r.t. computational costs, measured as the aggregate number of operations performed per search. Differences from one algorithm to the next range between 50 to 100 orders of magnitude. F-GBS requires close to $10^9$ operations in expectation for some datasets; in contrast, RankNetSearch ranges between 100 and 1000 operations and, in conclusion, presents an excellent trade-off between query and computational complexity.

### Scalability and Robustness

To study how the above algorithms scale with the dataset size, we also evaluate them on a synthetic dataset comprising objects placed uniformly at random at $\mathbb{R}^3$. The query and computational complexity of the five algorithms is shown in Fig. 7.2(a) and (b).

We observe the same discrepancies betwen algorithms we noted in Fig. 7.1. The linear growth in terms of $\log n$ implies a linear relationship between both measures of complexity w.r.t. the entropy $H(\mu)$ for all methods (we ommit the relevant figure for lack of space). In Fig. 7.2(b), we plot the query complexity of the robust RankNetSearch algorithm outlined in Section 7.4. For all simulations, the target success rate was set to 0.9, but the actual success rates we observed were considerably higher, close to 0.99. As shown in Fig. 7.3 we observe that, even for high error rates $\epsilon$, the query complexity remains low.


(a) Query Complexity

(b) Computational Complexity

Figure 7.2 – (a) Query and (b) computational complexity of the five algorithms as a function of the dataset size. The dataset is selected *u.a.r.* from the $\ell_1$ ball of radius 1.

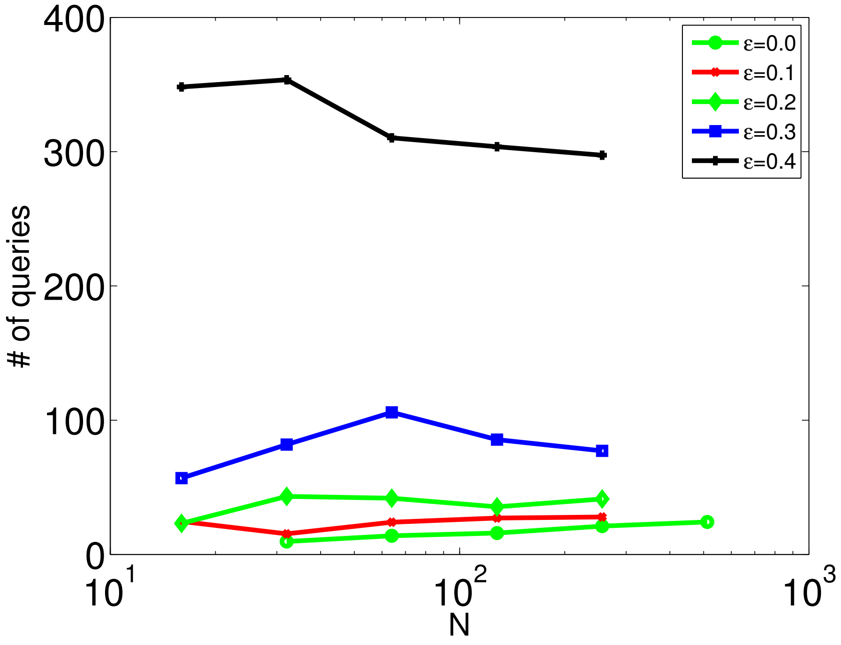


Figure 7.3 – Query complexity of RankNet as a function of $n$ under a faulty oracle.

Moreover, the high success rates that we observe, combined with the independence of the cost on $n$, suggest that we can further reduce the number of queries to lower values than the ones required by (7.5).

## 7.6 Analysis

### 7.6.1 Proof of Lemma 7.3

Using the ordered list containing the sets of equidistant objects described in Section 7.1, for any $z \in \mathcal{N}$, we can partition $\mathcal{N}$ into equivalence classes $A_1^z, A_2^z, \dots, A_k^z$ such that for any two objects $y, y' \in \mathcal{N}$, $y \in A_i^z$ and $y' \in A_j^z$ with $i < j$ if and only if $d(y,z) < d(y',z)$.

To construct $\mathcal{R}$, it suffices to show that (7.4) can be verified for any $z, z' \in E$ using only the above partition and $\mu$. If so, a $\rho$-rank net can be constructed in a greedy fashion as a maximal set whose points verify (7.4). This can be obtained by adding sequentially an arbitrary object to the net and excluding from future selections any nodes that violate (7.4) w.r.t. this newly added object. Indeed, for all $y \in E$,

$$B_y(d_y) = \bigcup_{j=1}^{\ell} A_j^y,$$

where

$$\ell = \inf\{i : \sum_{j=1}^{i} \mu(A_j^y) \geq \rho\mu(E)\}.$$

The statement thus follows as (7.4) is equivalent to

$$y' \notin B_y(d_y) \vee y \notin B_{y'}(d_{y'}).$$

To construct the Voronoi balls $B_y(r_y) \subseteq E$, $y \in \mathcal{R}$, we initialize each such ball to contain its center $y$. For each $z \in E \setminus \mathcal{R}$, let $j_{\min}$ be the smallest $j$ such that $\mathcal{R} \cap A_j^z \neq \emptyset$; the object $z$ is then added to the ball $B_y(r_y)$ of every $y \in \mathcal{N} \cap A_{j_{\min}}^z$.

For each $y$, $B_y(d_y)$ can constructed in $O(\log|E|)$ time via binary search on the ordered list of equidistant objects. Constructing the rank net in a greedy fashion requires determining which objects violate (7.4) w.r.t. a newly added object on the net, which may take $O(|E|)$ time. Hence, the overall complexity of constructing $\mathcal{R}$ is $O(|E|(|\mathcal{R}| + \log|E|))$. Finally, the construction of the Voronoi balls requires $O(|\mathcal{R}|)$ steps per object in $E$ to assign each object to a ball.

### 7.6.2 Proof of Lemma 7.4

Note first that, for all distinct $y, y' \in \mathcal{R}$, the balls

$$B(y, d_y(\rho, E)/4) \cap B(y', d_{y'}(\rho, E)/4) = \emptyset.$$

To see this, assume w.l.o.g. that $d_y \geq d_{y'}$ which implies that $d(y,y') \geq d_y - d_{y'}$. This is due to the fact that $\mu(B(y, d_{y'})) \geq \rho\mu(E)$, and hence, by (7.3), $d_y$ can be at most $d_{y'} + d(y,y')$. In case $d_y$ or $d_{y'}$ is zero, clearly

$$d(y,y') > d_y/2 > d_y/4 + d_{y'}/4.$$

If $0 < d_{y'} < d_y/2$, then

$$d(y,y') > d_y/2 \geq d_y/4 + d_{y'}/4.$$

If $d_{y'} \geq d_y/2 > 0$, then

$$d(y,y') \geq d_{y'} \geq d_y/2 > d_y/4 + d_{y'}/4.$$

Hence, in all cases $d(y,y') > d_y/4 + d_{y'}/4$ and as a result

$$B(y, d_y/4) \cap B(y', d_{y'}/4) = \emptyset.$$

To prove Lemma 7.4, observe that $d_y \leq 2R$ for all $y \in \mathcal{R}$ since $\mu(y,d) \geq \mu(E) > \rho\mu(E)$ for $d \leq 2R$. Therefore, $d_y/4 \leq R/2$ and thus $B(y, d_y/4) \subseteq B(x, 2R)$. Hence, by the definition of $c(\mu)$,

$$\sum_{y \in \mathcal{R}} \mu(B(y, d_y/4)) \leq \mu(B(x, 2R)) \leq c\mu(E).$$

Moreover,

$$\sum_{y \in \mathcal{R}} \mu(B(y, d_y/4)) \geq c^{-2} \sum_{y \in \mathcal{R}} \mu(B(y, d_y)) \geq c^{-2}\rho\mu(E)|\mathcal{R}|.$$

Therefore, $|\mathcal{R}| \leq c^3/\rho$.

### 7.6.3 Proof of Lemma 7.5

Observe first that, for all $z \in E$, there exists a $y \in \mathcal{R}$ such that $z \in B(y, d_y(\rho, E))$. To see this, assume otherwise. Then for any $y \in \mathcal{R}$,

$$d(z,y) > d_y(\rho, E) \geq \min\{d_y(\rho, E), d_z(\rho, E)\}$$

and we can add $z$ to $\mathcal{R}$, which contradicts its maximality.

To prove Lemma 7.5, we consider the following two cases. Suppose first $0 < r_y \leq d_y$. By (7.3), for any $\tilde{r} < d_y$, we have $\mu(B(y, \tilde{r}) < \rho\mu(E)$. In particular, $\mu(B(y, d_y/2) < \rho\mu(E)$. By the definition of $c$,

$$\mu(B(y, r_y) \leq \mu(B(y, d_y) \leq c\rho\mu(E).$$

For the second case, suppose that $r_y > d_y$. Let $z \in V_y$ is the point for which $d(y,z) = r_y$. By the above observation, we know that there exists a $y' \in \mathcal{R}$ such that $d(z, y') \leq d_{y'}$. As $r_y > d_y$, $y \neq y'$. On the other hand, $d(z, y') \geq d(z, y)$ since $z \in V_y$. Using the triangle inequality, we get

$$d(y, y') \leq d(y, z) + d(y', z) \leq 2d(y', z) \leq 2d_{y'}.$$

We know that

$$B(y, r_y) \subseteq B(y', d(y, y') + r_y).$$

Since $r_y = d(y,z) \leq d_{y'}$ we can say $B(y, r_y) \subseteq B(y', 3d_{y'})$. Finally, by the definition of $c$, we have

$$\mu(B(y, r_y)) \leq \mu(B(y', 3d_{y'})) \leq c^2\mu(B(y', d_{y'})) \leq c^3\rho\mu(E).$$

### 7.6.4 Proof of Theorem 7.6

Note first that, by induction, it can be shown that the version space $V$ is a subset of $E$; correctness is implied by this fact and the termination condition. To bound the number of queries, we first show that the process RankNet constructs a net with small cardinality.

**Lemma 7.8.** RankNet *terminates at* $\rho > \frac{1}{4c^3}$.

*Proof.* To see that the **while** loop terminates, observe that, by Lemma 7.5, for small enough

$$\rho < \min_{z \in E} \mu(z)/(c^3\mu(E)),$$

all Voronoi balls $B_y(r_y)$ of the $\rho$-rank net $\mathcal{R}$ will be singletons, so $\mathcal{I}$ will indeed be empty. Suppose thus that the loop terminates at some $\rho = \rho^*$. Since it did not terminate at $\rho = 2\rho^*$, there exists a ball $B_y(r_y)$ of the $2\rho$-rank net $\mathcal{R}$ such that $r_y > 0$ and $\mu(B_y(r_y)) > 0.5\mu(E)$. By Lemma 7.5, $\mu(B_y(r_y)) \leq c^3 2\rho\mu(E)$, and the lemma follows. □

Hence, from Lemmas 7.4 and 7.8, we get that the rank nets returned by RankNet have cardinality at most $4c^6$. On the other hand, by construction, a net covering a ball $B_y r_y$ consists of either singletons or balls with mass less than $0.5\mu(B_y(r_y))$. As a result, at each iteration, moving to the next object either halves the mass of the version space or leads to a leaf, and the search terminates. As at any point the version space has a mass greater than $\mu(t)$, the search will terminate after traversing most $\lceil \log_2(1/\mu(t)) \rceil$ iterations. Since, at each level, the number of accesses to the oracle are $\mathcal{R} - 1 \leq 4c^3$, the total query cost for finding target $t$ is at most $4c^6\lceil \log_2(1/\mu(t) \rceil$, and the query complexity statement follows. Finally, from Lemmas 7.3 and 7.8, the computational complexity of each RankNet call is at most $O\big(n(\log n + c^3)\log c\big)$.

### 7.6.5 Proof of Theorem 7.7

We first show the following auxiliary result.

**Lemma 7.9.** *Given a target $t$ and a noisy oracle with error probability bounded by $\epsilon$, the tournament among elements of the net $\mathcal{R}$ with repetitions $k_\delta(\ell, |\mathcal{R}|)$ returns the element in the set $\mathcal{R}$ that is closest to target $t$ with probability at least $1 - (\ell + 1/\delta)^{-2}$.*

*Proof.* We assume for simplicity that there are no ties, i.e. there is a unique point in $\mathcal{R}$ that is closest to $t$. The case with ties can be deduced similarly. We first bound the probability $p(k)$ that upon repeating $k$ times queries Oracle$(x, y, t)$, among $x$ and $y$ the one that wins the majority of comparisons is not the closest to $t$. Because of the bound $\epsilon$ on the error probability, one has

$$p(k) \leq \Pr(\mathrm{Bin}(k, \epsilon) \geq k/2),$$

where $\mathrm{Bin}(\cdot, \cdot)$ denotes the Binomial distribution. Azuma-Hoeffding inequality ensures that the right-hand side of the above is no larger than $\exp(-k(1/2 - \epsilon)^2/2)$. Upon replacing the number of repetitions $k$ by the expression (7.5), one finds that

$$p(k_\delta(\ell, |\mathcal{R}|)) \leq (\ell + 1/\delta)^{-2} / \lceil \log_2(|\mathcal{R}|) \rceil.$$

Consider now the games to be played by the element within $\mathcal{R}$ that is closest to $t$. There are at most $\lceil \log_2(|\mathcal{R}|) \rceil$ such games. By the union bound, the probability that the closest element loses on any one of these games is no less than $(\ell + 1/\delta)^{-2}$, as announced. □

By the union bound and the previous lemma we have conditionally on any target $t \in \mathcal{N}$ that

$$\Pr(\text{success} | T = t) \geq 1 - \sum_{\ell \geq 1} (\ell + 1/\delta)^{-2}).$$

The latter sum is readily bounded by $\delta$. The number of comparisons given that the target is $T = t$ is at most

$$\sum_{\ell=1}^{\lceil \log_2(1/\mu(t)) \rceil} 2|\mathcal{R}_\ell| k_\delta(\ell, |\mathcal{R}_\ell|) = O\left( \frac{1}{(\frac{1}{2} - \epsilon)^2} \log \frac{1}{\mu(t)} \log(\frac{1}{\delta} + \log \frac{1}{\mu(t)}) \right),$$

where the $O$-term depends only on the doubling constant $c$. The bound on the expected number of queries follows by averaging over $t \in \mathcal{N}$.

“ ··   ·–·· ·–– ·– –·–– ···   ·–– ·– –· – · –··
– –––   –··· ·   ··· ––– –– · –··· ––– –·· –·–– ······
··   ··· ···· ––– ··– ·–·· –··   ···· ·– ···– ·   –··· · · –·
–– ––– ·–· ·   ··· ·––· · –·–· ·· ··–· ·· –·–· ”

·–·· ·· ·–·· –·––   – ––– –– ·–·· ·· –·

# 8
# Conclusion

In this thesis, we investigated the scaling laws of graph-based information processing and its applications in large scale networks. In particular, all the applications we considered shared an important feature: a low-dimensional graph structure can reliably represent the available information. Based on this limited information, the goal was

- to position the nodes of the underlying graph (Chapters 2 and 3),
- to identify the defective nodes/edges of the graph and then route efficiently the information (Chapter 4 and 5),
- to navigate through a database of objects by establishing a dynamic graph (chapters 6 and 7).

We conclude the thesis by a brief discussion of the obtained results, open problems and possible directions for future research.

**Robust Positioning from Incomplete Information** In Chapter 2, we investigated the centralized and distributed sensor localization problem from a theoretical point of view and provided analytical bounds on the performance of such algorithms. More precisely, we analyzed the MDS-MAP and HOP-TERRAIN algorithms and showed that even when only the connectivity information is given and in the presence of detection failure, the resulting error of both algorithms is bounded and decays at a rate inversely proportional to the detection range.

There are many open questions that one might be interested in. For instance, in our model we assumed that all nodes have the same radio range. It would be interesting (and more realistic) to generalize our results for the case where different nodes have difference radio ranges. Another fruitful direction would be to understand the information theoretic limits of the localization problem, given the aforementioned limited and noisy information.

**Calibration Through Matrix Completion** In Chapter 3, we introduced a theoretical framework for calibration in circular ultrasound tomography devices. As we discussed, we faced three sources of uncertainty: structured missing ToFs, random missing ToFs, and an unknown time delay. To solve the calibration problem under this set of uncertainties, we developed a novel method based on structured matrix completion that first estimates the missing ToFs and then returns the positions. We also established an analytic bound on the overall error of the process under the assumption that the time delay was known.

Even though we introduced a recursive algorithm for finding the time-delay, we were not able to provide theoretical guarantees on its convergence (we mainly observed its convergence through simulations). This is still an interesting theoretical challenge and would require further work. It is quite reasonable to believe that our approach could be applied to other calibration scenarios (beyond the circular topology) with simple geometry.

**Graph-Constrained Group Testing** In Chapter 4, we looked at a well-studied inverse problem in statistics, called *non-adaptive group testing*, under graph constraints. For interesting classes of graphs we obtained a rather surprising result: the number of tests required to identify defective items is essentially similar to what is required in conventional group testing problems, where no such constraints on pooling is imposed. Specifically, if $T(n)$ corresponds to the mixing time of the constraint graph, we showed that with $m = O(d^2T^2(n)\log(n/d))$ non-adaptive tests, we can identify the $d$ defective items in a population of size $n$.

An important open question is how to come up with a better lower bound than $d\log(n/d)$ that also captures the constraints imposed by the graph. Another interesting direction would be to investigate the benefits of adaptive versus non-adaptive graph-constrained group tests. It might be possible to dramatically reduce the number of required tests by deploying an adaptive strategy. And finally, in this chapter, we only considered static graphs. It would be very interesting to see whether similar results could be obtained in dynamic graphs.

**Navigability with a Bias** In Chapter 5, we considered the problem of small-world network design. We first established a negative result (not a good start;)) where we showed that under the general heterogeneous demand the small-world network design is NP-hard. We then proposed a novel mechanism for edge addition and showed that its performance under greedy forwarding algorithm is bounded by two quantities: the bias of the target distribution (captured by the entropy and max-entropy) and the bias of the embedding (captured by the doubling constant).

One natural question that would arise after any NP-hard result is whether or not we can effectively approximate the optimum solution with a polynomial time algorithm. Another important aspect would be to obtain a uniform lower bound on the cost of the greedy forwarding algorithm, given the demand distribution and the embedding. This lower bound obviously depends on both the entropy and doubling constant. Although, how it does so is not clear.

**Content Search Through Comparisons** In Chapter 6, we talked about navigating through a database of objects by the means of comparisons. Based on the connection we made with the small-world network design, we devised a memoryless algorithm and bounded its performance. We also established a lower bound on any mechanism that solves content search through comparisons problem. Finally, an adaptive learning algorithm was proposed; it could effectively learn the demand distribution, as well as the ranks, and meet the performance guarantees of the memoryless algorithm. (We will talk about the open problems in the next paragraph)

**Comparison-Based Active Learning** In Chapter 7, we studied adaptive strategies for the problem of search through comparisons. We proposed a new strategy based on rank nets and we showed that for target distributions with a *bounded* doubling constant, it finds the target in a number of comparisons close to the entropy of the target distribution and, hence, of the optimum. We also extended these results to the case of noisy oracles.

An obvious obstacle against the wide adoption of our method as the basis for navigating through a database is the space complexity. The results of thousands of pairwise comparisons could, of course, be contradictory. A challenging problem that still remains open is to devise a search algorithm that, even in the presence of incomplete information (e.g., few comparisons), can still navigate through the database. Throughout Chapters 6 and 7, we assumed that human inference of proximity is accurately captured by a metric space structure. Another interesting research direction would be to assess the validity of this assumption through user trials.

# Morse Code

# A

| Input | Output | Value | Input | Output | Value | Input | Output | Value |
|---|---|---|---|---|---|---|---|---|
| `a` | ·– | a | `n` | –· | n | `3` | ···–– | 3 |
| `\aAcute` | ·––·– | á | `o` | ––– | o | `4` | ····– | 4 |
| `\ae` | ·–·– | a̋ | `\oe` | –––· | ő | `5` | ····· | 5 |
| `b` | –··· | b | `p` | ·––· | p | `6` | –···· | 6 |
| `c` | –·–· | c | `q` | ––·– | q | `7` | ––··· | 7 |
| `d` | –·· | d | `r` | ·–· | r | `8` | –––·· | 8 |
| `\ch` | –––– | ch | `s` | ··· | s | `9` | ––––· | 9 |
| `e` | · | e | `t` | – | t | `0` | ––––– | 0 |
| `\eAcute` | ··–·· | é | `u` | ··– | u | `;` | –·–·–· | ; |
| `f` | ··–· | f | `\ue` | ··–– | ű | `,` | ·–·–·– | , |
| `g` | ––· | g | `v` | ···– | v | `:` | –––··· | : |
| `h` | ···· | h | `w` | ·–– | w | `?` | ··––·· | ? |
| `i` | ·· | i | `x` | –··– | x | `!` | ––··–– | ! |
| `j` | ·––– | j | `y` | –·–– | y | `\dq` | –····– | ” |
| `k` | –·– | k | `z` | ––·· | z | `\sq` | ·––––· | ’ |
| `l` | ·–·· | l | `1` | ·–––– | 1 | `/` | –––––– | / |
| `m` | –– | m | `2` | ··––– | 2 | `.` | ······ | . |

The epigraph of Chapter 1 is a Chinese proverb:

> “A journey of a thousand miles starts with a single step.”

Chapter 2 talks about localization. For the epigraph of this chapter I liked the story, perhaps apocryphal, of legendary pianist Arthur Rubinstein who was approached by a passer-by in the street near New York’s world-famous Carnegie Hall.

> Passer-by: ’Pardon me sir, how do I get to Carnegie Hall?’
> Arthur Rubinstein: ’Practice, practice, practice!’

Chapter 3 is about the calibration of circular ultrasound tomography devices. Because they are mainly used for breast cancer screening, I quoted Jennifer Jergens as she opens up her TED talk with,

> “Yes, they are fake, the real ones tried to kill me,”

of course with a red face.

Chapter 4 is in the middle of this thesis. I think it makes perfect sense to quote Rosabeth Moss Cantor on,

> “The middle of every successful project looks like a disaster.”

Indeed, it looked like a disaster when I started writing.

Chapter 5 discusses the small-world phenomenon. I like the following quote by Steven Wright:

> “Everywhere is within walking distance if you have the time.”

When I was reading Alice in Wonderland, I fully appreciated the following conversation between Alice and the Cheshire Cat. It also resonates well with the content of Chapter 6.

> ’Which road do I take?’ Alice asked.
> ’Where do you want to go?’ responded the Cheshire Cat.
> ’I don’t know,’ Alice answered.
> ’Then,’ said the Cat, ’it doesn’t matter.’

I should probably start the last technical chapter (Chapter 7) with Rudiger’s favorite quote:

> “Amin, do something.”

I hope I have done enough.

And finally, this thesis is concluded by one of Lily Tomlin’s quotes. The reason is pretty obvious.

> “I always wanted to be somebody. I should have been more specific.”

# Bibliography

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, New York, USA, 2009.

[Aig88] M. Aigner. *Combinatorial Search.* Wiley-Teubner Series in Computer Science, Wiley, New York, 1988.

[AS09] G. Atia and V. Saligrama. Noisy group testing: An information theoretic perspective. In *Proceedings of the Allerton Conference on Communication, Control, and Computation, UIUC*, 2009.

[BC88] B. Bollobás and F. R. K. Chung. The diameter of a cycle plus a random matching. *SIAM J. Discret. Math.*, 1(3):328–333, August 1988.

[BFG09] M. A. Bender, J. T. Fineman, and S. Gilbert. A new approach to incremental topological ordering. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009.

[BG02] A. Blass and Y. Gurevich. Pairwise testing. *Bulletin of the EATCS*, 78:100–132, 2002.

[BG05] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications.* Springer, 2005.

[BGI$^+$08] R. Berinde, A. C. Gilbert, P. Indyk, H. J. Karloff, and M. J. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. *CoRR*, 2008.

[BISV08] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli. The hitchhiker's guide to successful wireless sensor network deployments. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008.

[BLT$^+$06] P. Biswas, T. Liang, K. Toh, T. Wang, and Y. Ye. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *IEEE Transactions on Automation Science and Engineering*, 3, 2006.

[BWS$^+$10] S. Branson, C. Wah, F. Schroff, B. Babenko, P. Welinder, P. Perona, and S. Belongie. Visual recognition with humans in the loop. In *European Conference on Computer Vision (ECCV)*, pages 438–451, 2010.

[BY04] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004.

[CC01] T. F. Cox and M. A. Cox. *Multidimensional Scaling.* Chapman & Hall, 2001.

[CCL$^+$04] R. Castro, M. Coates, G. Liang, R. Nowak, and B. Yu. Network tomography: recent developments. *Statistical Science*, 19:499–517, 2004.

[CHB04] J. Chen, Y. Huang, and J. Benesty. Time delay estimation. In *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, chapter 8, pages 197–227. 2004.

[CHH02] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-free Positioning in Mobile Ad-Hoc Networks. 2002.

[CHKV09] M. Cheraghchi, A. Hormati, A. Karbasi, and M. Vetterli. Compressed sensing with probabilistic measurements: A group testing solution. In *Proceedings of the Allerton Conference on Communication, Control, and Computation, UIUC*, 2009.

[Cla06] K. L. Clarkson. Nearest-neighbor searching and metric space dimensions. In *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, pages 15–59. MIT Press, 2006.

[CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 2nd edition, 2001.

[CMM$^+$00] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos. The bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments. *IEEE Transactions on Image Processing*, 9(1):20–37, 2000.

[CR08] E. J. Candès and Benjamin R. Exact Matrix Completion via Convex Optimization. 2008.

[CRT06] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, Feb. 2006.

[CT91] T. M. Cover and J. Thomas. *Elements of Information Theory.* 1991.

[Das05] S. Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17:337–344, 2005.

[DH99] D. Z. Du and F. Hwang. *Combinatorial Group Testing and its Applications.* World Scientific Series on Applied Mathematics, 1999.

[DJLW08] R. Datta, D. Joshi, J. Li, and Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 39, 2008.

[DJMI$^+$06] P. Drineas, A. Javed, M. Magdon-Ismail, G. Pandurangant, R. Virrankoski, and A. Savvides. Distance matrix reconstruction from incomplete distance information for sensor network localization. In *Proceedings of Sensor and Ad-Hoc Communications and Networks Conference (SECON)*, volume 2, pages 536–544, Sept. 2006.

[DLP$^+$07] N. Duric, P. Littrup, L. Poulo, A. Babkin, R. Pevzner, E. Holsapple, O. Rama, and C. Glide. Detection of breast cancer with ultrasound tomography: First results with the computed ultrasound risk evaluation (cure) prototype. *Medical Physics*, 34(2):773–785, 2007.

[DMW03] P. S. Dodds, R. Muhamad, and D. J. Watts. An Experimental Study of Search in Global Social Networks. *Science*, 301:827–829, 2003.

[Don06] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr. 2006.

[Duf06] N. Duffield. Network tomography of binary network performance characteristics. *IEEE Transactions on Information Theory*, 52(12):5373–5388, 2006.

[EK10] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World.* Cambridge University Press, 2010.

[Faz02] M. Fazel. Matrix rank minimization with applications. *Elec Eng Dept Stanford University*, 2002.

[FG05] Y. Fang and D. Geman. Experiments in mental face retrieval. In *Proc. of Audio- and Video-based Biometric Person Authentication*, 2005.

[FG07] M. Ferecatu and D. Geman. Interactive search for image categories by mental matching. In *IEEE 11th International Conference on Computer Vision (ICCV)*, 2007.

[FG09] M. Ferecatu and D. Geman. A statistical framework for image category search from a mental picture. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1087–1101, jun 2009.

[FG10] P. Fraigniaud and G. Giakkoupis. On the searchability of small-world networks with arbitrary underlying structure. In *ACM Symposium on Theory of Computing (STOC)*, 2010.

[FISS03] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *The Journal of Machine Learning Research (JMLR)*, 4, December 2003.

[FJs11] V. F. Farias, S. Jagabathula, and D. shah. A nonparametric approach to modeling choice with limited data. 2011.

[FLL06] P. Fraigniaud, E. Lebhar, and Z. Lotker. A doubling dimension threshold $\theta(\log\log n)$ for augmented graph navigability. In *ESA*, 2006.

[Gar72] M. R. Garey. Optimal binary identification procedures. In *SIAM Journal on Applied Mathematics*, 1972.

[GJ93] D. Geman and B. Jedynak. Shape recognition and twenty questions. In *Proc. Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, 1993.

[GK98] P. Gupta and P.R. Kumar. Critical power for asymptotic connectivity. In *Proceedings of the 37th IEEE Conference on Decision and Control*, volume 1, 1998.

[GK10] D. Golovin and A. Krause. Adaptive submodularity: A new approach to active learning and stochastic optimization. In *COLT*, 2010.

[GLS08] N. Goyal, Y. Lifshits, and H. Schutze. Disorder inequality: a combinatorial approach to nearest neighbor search. In *ACM International Conference on Web Search and Data Mining (WSDM)*, 2008.

[GM07] I. C. Gormley and T. B. Murphy. A latent space model for rank data. In *Proceedings of the 2006 conference on Statistical network analysis*, pages 90–102. Springer-Verlag, 2007.

[GS04] T. Gevers and A. Smeulders. Content-based image retrieval an overview. *Emerging Topics in Computer Vision*, 2004.

[HJ85] R. Horn and C. Johnson. *Matrix analysis.* Cambridge University Press, 1985.

[HKM$^+$08] B. Haeupler, T. Kavitha, R. Mathew, S. Sen, and R. E. Tarjan. Incremental cycle detection, topological ordering, and strong component maintenance. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the AMS*, 43(4):439–561, 2006.

[IM98] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing (STOC)*, 1998.

[JH01] G. Borriello J. Hightower. Location Systems for Ubiquitous Computing. 2001.

[JHSV07] I. Jovanović, A. Hormati, L. Sbaiz, and M. Vetterli. Efficient and stable acoustic tomography using sparse reconstruction methods. In *19th International Congress on Acoustics*, 2007.

[JM11] A. Javanmard and A. Montanari. Localization from Incomplete Noisy Distance Measurements. *arXiv.org*, 2011.

[JN11] K. G. Jamieson and R. D. Nowak. Active ranking using pairwise comparisons. In *Neural Information Processing Systems (NIPS)*, 2011.

[Joh77] D. B. Johnson. Efficient algorithms for shortest paths in sparse networks. *Journal of ACM*, 24(1):1–13, 1977.

[Jov08] I. Jovanović. *Inverse problems in acoustic tomography.* PhD thesis, EPFL, Lausanne, 2008.

[JS08] Srikanth Jagabathula and Devavrat Shah. Inferring rankings under constrained sensing. In *Neural Information Processing Systems (NIPS)*, 2008.

[JS11] S. Jagabathula and D. Shah. Inferring rankings under constrained sensing. In *IEEE Transactions on Information Theory*, 2011.

[JSV09] I. Jovanovic, L. Sbaiz, and M. Vetterli. Acoustic tomography for scalar and vector fields: theory and application to temperature and wind estimation. *Journal of Atmospheric and Oceanic Technology*, 26(8):1475 – 1492, 2009.

[KF09] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[KL04] R. Krauthgamer and J. R. Lee. Navigating nets: simple algorithms for proximity search. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2004.

[Kle00] J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *ACM Symposium on Theory of Computing (STOC)*, 2000.

[Kle01] J. Kleinberg. Small-world phenomena and the dynamics of information. In *Neural Information Processing Systems (NIPS)*, 2001.

[Kle06] J. Kleinberg. Complex networks and decentralized search algorithms. In *Proceedings of the International Congress of Mathematicians (ICM 2006), Madrid, Spain*, volume 3, pages 1019–1044, 2006.

[KM06] A. A. Kannan and G. Mao. Simulated annealing based wireless sensor network localization. *Journal of Computers*, 2006.

[KM10] R. H. Keshavan and A. Montanari. Matrix completion from noisy entries. *The Journal of Machine Learning*, 2010.

[KMO09] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. In *Advances in Neural Information Processing Systems*, December 2009.

[KMO10] R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Trans. Inform. Theory*, 2010.

[KMV06] A. A. Kannan, G. Mao, and B. Vucetic. Simulated Annealing based Wireless Sensor Network Localization with Flip Ambiguity Mitigation. In *Vehicular Technology Conference*, pages 1022–1026, 2006.

[KO09] R. H. Keshavan and S. Oh. OptSpace: A gradient descent algorithm on the grassman manifold for matrix completion. *IEEE Trans. Inform. Theory*, 2009.

[KO10] A. Karbasi and S. Oh. Distributed sensor network localization from local connectivity: Performance analysis for the HOP-TERRAIN algorithm. In *ACM SIGMETRICS*, June 2010.

[KOPV10] A. Karbasi, S. Oh, R. Parhizkar, and M. Vetterli. Ultrasound Tomography Calibration Using Structured Matrix Completion. In *The 20th International Congress on Acoustics*, 2010.

[KR02] D. R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2002.

[LHD+09] C. Li, L. Huang, N. Duric, H. Zhang, and C. Rowe. An improved automatic time-of-flight picker for medical ultrasound tomography. *Ultrasonics*, 49(1):61 – 72, 2009.

[LR03] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Comput. Netw.*, 43(4):499–518, 2003.

[LSDJ06] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-based multimedia information retrieval: State of the art and challenges. *ACM Trans. Multimedia Comput. Commun. Appl.*, 2, February 2006.

[LZ09] Y. Lifshits and S. Zhang. Combinatorial algorithms for nearest neighbors, near-duplicates and small-world design. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2009.

[Mar06] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[MFA07] G. Mao, B. Fidan, and B Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10):2529–2553, July 2007.

[Mil67] S. Milgram. The small world problem. In *Psycology Today*, pages 61–67, 1967.

[ML09] B. Mcfee and G. Lanckriet. Partial order embedding with multiple kernels. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 721–728. ACM, 2009.

[MP05] S. Muthukrishnan and G. Pandurangan. The bin-covering technique for thresholding random geometric graph properties. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 989–998, Philadelphia, PA, USA, 2005.

[MR95] R. Motwani and P. Raghavan. *Randomized Algorithms.* Cambridge University Press, 1995.

[MU05] M. Mitzenmacher and E. Upfal. *Probability and Computing.* Cambridge University Press, 2005.

[Nat08] F. Natterer. Acoustic mammography in the time domain. Technical report, University of Münster, Germany, 2008.

[New10] M. E. J. Newman. *Networks: An Introduction.* Oxford University Press, 2010.

[NM05] V. Nguyen and C. Martel. Analyzing and characterizing small-world graphs. In *Proceedings of the 16th annual ACM-SIAM Symposium On Discrete algorithms*, pages 311–320, 2005.

[NN01] D. Niculescu and B. Nath. Ad hoc positioning system (aps). In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, pages 2926–2931, 2001.

[NN03] D. Niculescu and B. Nath. DV based positioning in ad hoc networks. *Journal of Telecommunication Systems*, 22:267–280, 2003.

[NNK$^+$05] L. D. Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102:11623–11628, 2005.

[Now09] R. D. Nowak. The geometry of generalized binary search. *Transactions on Information Theory*, 5, 2009.

[NT06] H. X. Nguyen and P. Thiran. Using end-to-end data to infer lossy links in sensor networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM'06)*, 2006.

[NT07] H. X. Nguyen and P. Thiran. The boolean solution to the congested IP link location problem: Theory and practice. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pages 2117–2125, May 2007.

[NW01] F. Natterer and F. Wübbeling. *Mathematical Methods in Image Reconstruction.* SIAM, 2001.

[OKM10] S. Oh, A. Karbasi, and A. Montanari. Sensor network localization from local connectivity : performance analysis for the MDS-MAP algorithm. In *Proc. of the IEEE Inform. Theory Workshop*, January 2010.

[PCB00] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

[PK03] D. J. Pearce and P. Kelly. Online algorithms for maintaining the topological order of a directed acyclic graph. Technical report, Imperial College, 2003.

[PKV11] R. Parhizkar, A. Karbasi, and M. Vetterli. Calibration in circular ultrasound tomography devices. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 549 –552, may 2011.

[PL94] P. Pevzner and R. J. Lipshutz. Towards DNA sequencing chips. In *Proceedings of MFCS*, volume 841 of *Lecture Notes in Computer Science*, pages 143–158, 1994.

[Rec09] B. Recht. A simpler approach to matrix completion. *Arxiv preprint arXiv:0910.0651*, 2009.

[Rut08] I. Ruthven. Interactive information retrieval. *Annual Review of Information Science and Technology*, 42(1):43–91, 2008.

[RXH11] B. Recht, W. Xu, and B. Hassibi. Null space conditions and thresholds for rank minimization. *Math. Program.*, 127(1):175–202, Mar 2011.

[SG59] M. Sobel and P. A. Groll. Group testing to eliminate efficiently all defectives in a binomial sample. *Bell Systems Technical Journal*, 38:1179–1252, 1959.

[SHD07] F. Simonetti, L. Huang, and N. Duric. On the spatial sampling of wave fields with circular ring apertures. *Journal of Applied Physics*, 101(8):083103, 2007.

[SHS01] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 166–179. ACM, 2001.

[SJ89] A. Sinclair and M. Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, 1989.

[Sli05] A. Slivkins. Distance estimation and object location via rings of neighbors. In *PODC*, 2005.

[SLR02] C. Savarese, K. Langendoen, and J. Rabaey. Robust positioning algorithms for distributed ad-hoc wireless sensor networks. In *USENIX Technical Annual Conference*, pages 317–328, Monterey, CA, June 2002.

[SPS03] A Savvides, H Park, and M. Srivastava. The n-hop multilateration primitive for node localization problems. *Mob. Netw. Appl.*, 8(4):443–451, 2003.

[SRB01] C. Savarese, J. Rabaey, and J. Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *ICASSP*, pages 2037–2040, 2001.

[SRZF03] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz. Localization from mere connectivity. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking and computing*, pages 201–212, New York, NY, USA, 2003. ACM.

[SRZF04] Y. Shang, W. Ruml, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 15(11):961–974, 2004.

[SWS$^+$00] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1349–1380, 2000.

[TD09] D. Tschopp and S. N. Diggavi. Approximate nearest neighbor search through comparisons. 2009.

[TD10] D. Tschopp and S. N. Diggavi. Facebrowsing: Search and navigation through comparisons. In *ITA workshop*, 2010.

[TDDM11] D. Tschopp, S. N. Diggavi, P. Delgosha, and S. Mohajer. Randomized algorithms for comparison-based search. In *Neural Information Processing Systems (NIPS)*, 2011.

[Wat99] D. J. Watts. *Small worlds: The dynamics of networks between order and randomness.* Princeton University Press, 1999.

[WBS08] F. E. Walter, S. Battiston, and F. Schweitzer. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, 16:57–74, February 2008.

[Wil96] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing*, pages 296–303, 1996.

[Wol85] J. Wolf. Born-again group testing: multiaccess communications. *IEEE Transactions on Information Theory*, 31:185–191, 1985.

[WR09] R. White and R. Roth. *Exploratory Search: Beyond the Query-Response Paradigm.* Morgan & Claypool, 2009.

[WS98] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393:440–442, 1998.

[XMT11] W. Xu, E. Mallada, and A. Tang. Compressive sensing over graphs. In *INFOCOM*, pages 2087–2095, 2011.

[Xu02] N. Xu. A survey of sensor network applications. *IEEE Communications Magazine*, 40, 2002.

[ZGG02] H. Zhang, A. Goel, and R. Govindan. Using the small-world model to improve freenet performance. In *INFOCOM*, 2002.

# Curriculum Vitae

---

**Amin Karbasi**
Rue de la Barre 6
CH-1005 Lasuanne
Switzerland

+41 788 988 344 (phone)
amin.karbasi@epfl.ch
`http://people.epfl.ch/amin.karbasi`

## Research Interests

- Graphical Models.
- Compressed Sensing
- Sensor Networks
- Biomedical Signal Processing
- Recommender Systems
- Information Theory

## Education

2008-2012 **Ph.D. in Computer and Communication Sciences**
Ecole Polytechnique Fédérale de Lausanne, Switzerland.
Dissertation Title: *Graph-Based Information Processing: Scaling Laws and Applications.*
Supervisors: *Prof. Rudiger Urbanke, Prof. Martin Vetterli*

2004-2007 **M.Sc. in Computer and Communication Sciences**
Ecole Polytechnique Fédérale de Lausanne, Switzerland.
Dissertation Title: *Video Compression over Wireless Channels.*
Supervisor: *Prof. Pascal Frossard.*

2003-2004 **B.Sc. in Electrical Engineering**
Ecole Polytechnique Fédérale de Lausanne, Switzerland.

2000-2002 **B.Sc. in Electrical Engineering (Transferred to EPFL)**
Tehran Polytechnic University of Technology, Iran.

## Honors, Awards and Distinctions

April 2011 Recipient of the **ICASSP Best Student Paper Award** for *"Calibration in Circular Ultrasound Tomography Devices"* (with Reza Parhizkar).

| | |
|---|---|
| June 2010 | Recipient of the **ACM/SIGMETRICS Best Student Paper Award** for *"Distributed Sensor Network Localization from Local Connectivity* (with Sewoong Oh ). |
| May 2010 | **ISIT Best Student Paper Award Nominee** for *"Graph-Constrained Group Testing"*. |
| Jan. 2008 | Recipient of the **Departmental Fellowship**, Lausanne, Switzerland. |
| July 2002 | Team leader of the **World Champion** rescue robot team in the 6th International RoboCup competition, Japan. |

## Research Experience

| | |
|---|---|
| Mar-Sep '12 | **Nokia Research Center, Lausanne, Switzerland.**<br>Work specifically on interactive learning algorithms. |
| Oct-Dec '10 | **Microsoft Research Center, Cambridge, UK.**<br>Work specifically on job scheduling for cloud computing.<br>*Supervisor*: Dr. Milan Vojnovic. |
| Feb-Jul '10 | **Technicolor Research Center, Paris, France.**<br>Work specifically on comparison-based content search.<br>*Supervisor*: Dr. Laurent Massoulié. |
| Mar-Dec '07 | **Logitech Research Center, Lausanne, Switzerland.**<br>Work specifically on video compression over wireless channels. |
| Apr-Oct '05 | **NEC Media & Information Research Lab, Tokyo, Japan.**<br>Work specifically on Direction of Arrival (DOA) estimation. *Supervisor*: Dr. Akihiko Sugiyama. |
| Mar-Sep '04 | **IBM Research Center, Zurich, Switzerland.**<br>Work specifically on error correction codes suitable for large storage systems.<br>*Supervisor*: Prof. Rudiger Urbanke. |

## Publications

(All publications are available online at `http://people.epfl.ch/amin.karbasi`)

– **Submitted Journal Papers:**

1. A . Karbasi, S. Ioannidis and L. Massoulié. *"Robust Comparison-Based Search"*. Under submission to IEEE Transactions on Information Theory, 2012.
2. A . Karbasi, S. Ioannidis and L. Massoulié. *"Adaptive Content Search Through Comparisons"*. Submitted to the Journal of Machine Learning Research (JMLR), 2011. (arXiv:1107.3059)
3. R . Parhizkar, A. Karbasi, S. Oh and M. Vetterli. *"Calibration for Ultrasound Breast Tomography Using Matrix Completion"*. Submitted to IEEE Transactions on Signal Processing, 2011. (arXiv:1012.4928)

4. A . Hormati, A. Karbasi, S. Mohajer and M. Vetterli. *"An Estimation Theoretic Approach for Sparsity Pattern Recovery in the Noisy Setting"*. Submitted to IEEE Transactions on Signal Processing, 2011. (arXiv:0911.4880)

- **Published Journal Papers:**

5. A . Karbasi and S. Oh. *"Robust Localization from Incomplete Local Information"*. To appear in IEEE/ACM Transactions on Networking, 2011. (arXiv:1110.3018)
6. A . Amini, A. Karbasi and F. Marvasti. *"Low-rank Matrix Approximation Using Point-wise Operators"*. In IEEE Transactions on Information Theory, vol. 58, p 302-310, Jan. 2012.
7. M . Cheraghchi, A. Karbasi, S. Mohajer and V. Saligrama. *"Graph-Constrained Group Testing"*. In IEEE Transactions on Information Theory, vol. 58, p 248 - 262, Jan. 2012.
8. M . Cheraghchi, A. Hormati, A. Karbasi and M. Vetterli. *"Group Testing with Probabilistic Tests: Theory, Design and Application"*. In IEEE Transactions on Information Theory, vol. 57, p 7057 - 7067, Oct. 2011.

- **Submitted Conference Papers:**

9. A . Karbasi, A. H. Salavati and A. Shokrollahi. *"Coupled Neural Associative Memories"*.
10. A . Karbasi and M. Zadimoghaddam. *"Constrained Binary Identification Problem"*.
11. A . Karbasi, A. H. Salavati and A. Shokrollahi. *"Iterative Learning in Modular Associative Memories"*.

- **Published Conference Papers:**

12. A . Karbasi and M. Zadimoghaddam. *"Sequential Group Testing with Graph Constraints"*. To appear in IEEE Information Theory Workshop (ITW), 2012.
13. A . Karbasi, S. Ioannidis and L. Massoulie. *"Comparison-Based Learning with Rank Nets"*. In Proceedings of the 29th International Conference on Machine Learning (ICML), 2012.
14. A . H. Salavati and A. Karbasi. *"Multi-Level Error-Resilient Neural Networks"*. In Proceedings of IEEE International Symposium on Information Theory (ISIT), 2012.
15. A . Karbasi, S. Ioannidis and L. Massoulie. *"Hot or Not: Interactive Content Search Using Comparisons"*. In Information Theory and Applications Workshop (ITA), 2012. (Invited paper)
16. A . Karbasi and M. Zadimoghaddam. *"Compression with Graphical Constraints: An Interactive Browser"*. In Proceedings of IEEE International Symposium on Information Theory (ISIT), 2011.
17. A . Karbasi, S. Ioannidis and L. Massoulie. *"Content Search Through Comparisons"*. In Proceedings of the 38th International Colloquium on Automata, Languages and Programming (ICALP), 2011.
18. R . Parhizkar, A. Karbasi and M. Vetterli. *"Calibration in Circular Ultrasound Tomography Devices"*. In Proceedings of 36th International Conference on Acoustics, Speech and Signal Processing, 2011. **(Best Student Paper Award)**
19. A . Karbasi. *"From Centralized to Distributed Sensor Localization"*. In Proceedings of ACM MobiHoc-S3, 2010. (Invited Paper)

20. A . Karbasi, S. Oh, R. Parhizkar and M. Vetterli. *"Ultrasound Tomography Calibration using Structured Matrix Completion"*. In Proceedings of 20th International Congress on Acoustics, 2010.
21. A . Karbasi and S. Oh. *"Distributed Sensor Network Localization from Local Connectivity: Performance Analysis for the HOP-TERRAIN Algorithm"*. In Proceedings of ACM SIGMETRICS, 2010. (**Best Student Paper Award**)
22. M . Cheraghchi, A. Karbasi, S. Mohajer and V. Saligrama. *"Graph-Constrained Group Testing"*. In Proceedings of IEEE International Symposium on Information Theory (ISIT), 2010. (**Nominated for Best Student Paper Award**)
23. S . Oh, A. Karbasi and A. Montanari. *"Sensor Network Localization from Local Connectivity : Performance Analysis for the MDS-MAP Algorithm"*. In Proceedings of IEEE Information Theory Workshop (ITW), 2010.
24. M . Cheraghchi, A. Hormati, A. Karbasi and M. Vetterli. *"Compressed Sensing with Probabilistic Measurements: A Group Testing Solution"*. In Proceedings of 47th Allerton Conference on Communication, Control and Computing, 2009.
25. A . Karbasi, A. Hormati, S. Mohajer and M. Vetterli. *"Support Recovery in Compressed Sensing: An Estimation Theoretic Approach"*. In Proceedings of IEEE International Symposium on Information Theory (ISIT), 2009.
26. A . Karbasi and A. Sugiyama. *"A New DOA Estimation Method Using a Circular Microphone Array"*. In Proceedings of 15th European Signal Processing Conference (EUSIPCO), 2007.
27. A . Karbasi and A. Sugiyama. *"A DOA Estimation Method For an Arbitrary Triangular Microphone Arrangement"*. In Proceedings of 14th European Signal Processing Conference (EUSIPCO), 2006.

– **Patents:**

28. A . Karbasi, S. Ioannidis and L. Massoulié. *"Comparison-Based Learning with Rank Nets"*, Technicolor, Paris, 2012.
29. A . Karbasi and M. Vojnovic, *"Greedy Scheduling for Distributed Computing Clusters"*, Microsoft Research, Cambridge, 2011.
30. A . Karbasi, S. Ioannidis and L. Massoulié. *"Content Search Through Comparisons"*, Technicolor, Paris, 2011.

# Teaching Experience

Fall 2011 **Dynamical Systems Theory, EPFL**
Teaching Assistant.

Spring 2011 **Advanced Theoretical Computer Science, EPFL**
Teaching Assistant.

Spring 2009 **Advanced Signal Processing: Wavelets and Applications, EPFL**
Teaching Assistant.

Spring 2008 **Modern Coding Theory, EPFL**
Teaching Assistant.

Spring 2006 **Introduction to Communication Systems, EPFL**
Teaching Assistant.

| | |
|---|---|
| Spring 2006 | **Advanced Digital Communications, EPFL**<br>Teaching Assistant. |
| Fall 2006 | **Principles of Digital Communications, EPFL**<br>Teaching Assistant. |
| Spring 2005 | **Information Theory and Coding, EPFL**<br>Teaching Assistant. |