Data Driven Network Analysis and Applications

Dissertation

zur Erlangung des Doktorgrades
Dr. rer. nat.
der Mathematisch-Naturwissenschaftlichen Fakultäten
der Georg-August-Universität zu Göttingen

im PhD Programme in Computer Science (PCS)
der Georg-August University School of Science (GAUSS)

vorgelegt von

Narisu Tao

aus Inner Mongolia, China

Göttingen
im August 2015

Betreuungsausschuss:	Prof. Dr. Xiaoming Fu, Georg-August-Universität Göttingen Dr. Jan Nagler, Eidgenössische Technische Hochschule Zürich
Prüfungskommission:	
Referent:	Prof. Dr. Xiaoming Fu, Georg-August-Universität Göttingen
Korreferenten:	Prof. Dr. Dieter Hogrefe, Georg-August-Universität Göttingen
Weitere Mitglieder der Prüfungskommission:	Prof. Dr. K.K. Ramakrishnan, University of California, Riverside, USA Prof. Dr. Carsten Damm, Georg-August-Universität Göttingen Prof. Dr. Winfried Kurth, Georg-August-Universität Göttingen Prof. Dr. Burkhard Morgenstern, Georg-August-Universität Göttingen

Tag der mündlichen Prüfung: 14. September 2015

Abstract

Data is critical for scientific research and engineering systems. However, data collection procedures are often subject to high cost or heavy loss rate. It is challenging to accurately estimate missing or unobserved data points through the available ones. To cope with this challenge, data interpolation methods have been utilized to approximate the missing data with lower price.

In this thesis, we study three specific problems on missing data interpolation in computer networking. They are 1) Autonomous System (AS) level path inference problem, 2) environment reconstruction problem in Wireless Sensor Networks (WSNs) and 3) rating of network paths inference problem.

For the first problem, we bring a new angle to the AS path inference by exploiting the metrical tree-likeness or the low hyperbolicity of the Internet, part of the complex network properties of the Internet. We show that such property can generate a new constraint that narrows down the searching space of possible AS paths to a much smaller size. Based on this observation, we propose two new algorithms, namely HyperPath and Valley-free HyperPath. With intensive evaluations on AS paths from real-world BGP Routing Information Bases, we show that the proposed algorithms can achieve better performance. We demonstrate that our algorithms can significantly reduce inter-AS traffic for P2P applications with an improved AS path prediction accuracy.

For the second problem, we propose a new approach, namely Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing (PMEST-CS), to boost the performance of CS-based methods for environment reconstruction in WSNs. During algorithm design, we consider two new perspectives, which are exploiting the sparsity in spatio-temporal difference in environment and using probabilistic model and inference to enrich the available dataset. Experimental results utilizing the two real-world datasets show that significant performance gains, in terms of reconstruction quality, can be obtained in comparison with the state of the art CS-based methods.

For the third problem, we investigate the rating of network paths, which is not only informative but also cheap to obtain. We firstly address the scalable acquisition of path ratings by statistical inference. By observing similarities to recommender systems, we examine the applicability of solutions to recommender system and show that our inference problem can be solved by a class of matrix factorization techniques. Then, we investigate the usability of rating-based network measurement and inference in applications. A case study is performed on whether locality awareness can be achieved for overlay networks of Pastry and BitTorrent using inferred ratings. We show that such coarse-grained knowledge can improve the performance of peer selection and that finer granularities do not always lead to larger improvements.

Acknowledgements

I have been fortunate to work with many people. Without their kind help, this thesis would never have been possible.

My deep appreciation goes to my advisor Prof. Dr. Xiaoming Fu. It was his constant guidance, support, and encouragement that allow me to pursue my diverse research interests. His valuable assistance, suggestions, and feedback made this thesis possible.

I would like to sincerely thank Dr. Wei Du. I have learned and benefited hugely from our collaboration. His deep knowledge and insight on research have strongly shaped the way I work.

I am greatly indebted to Dr. Xu Chen, with whom we had so many fruitful discussion on research ideas. He had spent a lot of time revising, polishing, and improving almost every single paper of mine. Without his patience and efforts, this thesis would not have been what it is today.

Last but definitely not least I am deeply grateful to my former and current colleagues at the Computer Networks Group at the University of Göttingen, especially Konglin Zhu, Florian Tegeler, Mayutan Arumaithurai, Jiachen Chen, Hong Huang and David Koll. The whole lab helped me to continuously improve through constructive criticism and reviews, hours over hours of discussions, collaboration, and the enjoyable time in the lab.

My thanks in addition go to Dr. Jan Nagler for being a member of my thesis committee; I also thank him, Prof. Dr. K.K. Ramakrishnan, Prof. Dr. Dieter Hogrefe, Prof. Dr. Carsten Damm, Prof. Dr. Burkhard Morgenstern and Prof. Dr. Winfried Kurth for serving as the exam board for my thesis.

I owe a great deal to my parents. Their unconditional and endless love and support is always my motivation to go forward.

I would like to thank my wife, Tselmeg, who is always my constant source of strength. With her company, the four years PhD life at Göttingen was full of memorable moments. I want to thank my daughter, Dulaan. She has been bringing us so many joy and happiness since her birth. To them I dedicate this thesis.

Contents

Abstract III

Acknowledgements V

Table of Contents VII

List of Figures XI

List of Tables XIII

1 Introduction 1

1.1 The Problem 1

1.1.1 AS Path Inference 2

1.1.2 Environment Reconstruction in WSNs 3

1.1.3 Rating of Network Paths Inference 3

1.2 Dissertation Contributions 4

1.2.1 Improving AS Path Inference Accuracy 4

1.2.2 Improving Environment Reconstruction Accuracy in Sensor Network 5

1.2.3 Improving Locality-Awareness in Overlay Network Construction and Routing 6

1.3 Dissertation Overview 7

2 AS Path Inference from Complex Network Perspective 9

2.1 Introduction 11

2.2 Related Work 13

2.3 δ-hyperbolicity: Tree-likeness from Metric Point of View 14

2.3.1 Definition 15

2.3.2 Low Hyperbolicity of Scale-free Networks 15

2.4 HyperPath Method for AS Path Inference 17

2.4.1 Data Collection and Analysis 17

2.4.2 Algorithms 18

2.4.3 Discussion 20

2.5 Evaluation . . . 21
2.5.1 Benchmark Methods . . . 22
2.5.2 Experiment Set-up . . . 23
2.5.3 Estimation Accuracy . . . 24
2.5.4 Application: Inter-domain Traffic Reduction for BitTorrent P2P System . . . 27
2.6 Chapter Summary . . . 28

3 Probabilistic Model Enhanced Compressive Sensing for Environment Reconstruction in Sensor Networks 31
3.1 Introduction . . . 33
3.2 Related Work . . . 34
3.3 Problem Formulation . . . 35
3.3.1 Environmental Data Reconstruction . . . 35
3.3.2 Problem Statement . . . 35
3.4 Two Real World WSN Datasets . . . 36
3.4.1 Intel Lab Dataset . . . 36
3.4.2 Uppsala Dataset . . . 37
3.5 Probabilistic Model for WSN . . . 37
3.5.1 Preprocessing: Data Quantization . . . 38
3.5.2 Pairwise MRF for WSN . . . 38
3.6 Model Learning in WSN MRF . . . 40
3.6.1 The Log-linear Representation . . . 40
3.6.2 Spatio-Temporal Local Distribution Learning Algorithm from Complete Data . . . 43
3.6.3 Spatio-Temporal Local Distribution Learning Algorithm from Incomplete Data . . . 44
3.7 Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing . . . 45
3.7.1 Compressive Sensing Based Approach Design . . . 46
3.7.2 ESTI-CS Approach . . . 47
3.7.3 Our Approach: PMEST-CS . . . 48
3.8 Evaluation . . . 51
3.8.1 Incomplete Data Driven Model Training . . . 52
3.8.2 Environment Reconstruction . . . 53
3.9 Chapter Summary . . . 53

4 Rating Network Paths for Locality-Aware Overlay Construction and Routing 57
4.1 Introduction . . . 59
4.2 Related Work . . . 61
4.2.1 Inference of Network Path Properties . . . 61
4.2.2 Locality-Aware Overlay Networks . . . 61

4.3 Properties and Rating of Network Paths 62
4.3.1 Properties of Network Paths 62
4.3.2 Rating of Network Paths 63
4.4 Network Inference of Ratings 64
4.4.1 Problem Statement 64
4.4.2 Connections to Recommender Systems 65
4.4.3 Matrix Factorization 67
4.4.4 MF for Network Inference 70
4.4.5 Comparison of Different MF Models 71
4.5 Case Study: Locality-Aware Overlay Construction and Routing 76
4.5.1 Pastry 77
4.5.2 BitTorrent 78
4.5.3 Remarks 80
4.6 Chapter Summary 80

5 Conclusion 81

Bibliography 83

Curriculum Vitae 93

List of Figures

2.1 Power law distribution of node degrees in the AS topology 18

2.2 Two paths example 20

2.3 inter-domain traffic reduction on unstructured overlay networks 27

3.1 Two real world WSN deployments 37

3.2 The graphical model of Intel lab WSN 39

3.3 The spatial correlation knowledge learned from Intel lab temperature dataset with increasing loss rate. The bars on the right hand side represent the correlation factors. A larger value means a stronger correlation 41

3.4 Spatio-temporal difference analysis in selected real world datasets 49

3.5 Normalized entropy obtained by probabilistic model versus normalized error of expected value when 50% data is missing 50

3.6 Loss data prediction application based on learned model from incomplete data with random loss rate from 10% to 90% 53

3.7 Performance of different CS-based algorithms on environment reconstruction 54

3.8 Uppsala light intensity data and its restoration with 60% data loss rate by different CS-based environment reconstruction methods 55

4.1 A matrix completion view of network inference. In (b), the blue entries are measured path properties and the green ones are missing. Note that the diagonal entries are empty as they represent the performance of the path from each node to itself which carries no information 64

4.2 The singular values of a 2255×2255 RTT matrix, extracted from the Meridian dataset [1], and a 201×201 ABW matrix, extracted from the HP-S3 dataset [2], and of their corresponding rating matrices. The rating matrices are obtained by thresholding the measurement matrices with $\tau = \{20\%, 40\%, 60\%, 80\%\}$ percentiles of each dataset. The singular values are normalized so that the largest ones equal 1 66

4.3 Matrix factorization 68

4.4 Stretch of the routing performance of Pastry, defined as the ratio between the routing metric of Pastry using inferred ratings and of Pastry using true measurements. Note that $R = 0$ means that no proximity knowledge is used, and $R = \infty$ means that inferred values are used 78

4.5 Performance of peer selection for BitTorrent, calculated as the average link performance between each pair of selected peers. Note that "*true*" means that true measurements are used 79

List of Tables

2.1 Synthetic scale-free networks 16

2.2 Distribution of the δ-hyperbolicity value of quadruplets in different graphs . 16

2.3 Comparisons between the different AS path inference methods 23

2.4 Confusion matrices of the prediction performance of the baseline method . 25

2.5 Confusion matrices of the prediction performance of the HyperPath method 25

2.6 Confusion matrices of the prediction performance of the AS relationships based method 26

2.7 Confusion matrices of the prediction performance of the KnownPath method 26

2.8 Confusion matrices of the prediction performance of the Valley-free HyperPath method 26

3.1 Selected datasets for spatio-temporal difference analysis 48

3.2 Selected datasets for probabilistic model learning form incomplete data . . 52

3.3 Selected datasets for environment reconstruction 52

4.1 An example of a recommender system 66

4.2 RMSE on different datasets 73

4.3 Confusion matrices 74

Chapter 1

Introduction

1.1 The Problem

In computer networking, we measure and collect data for the desirable information behind. In this thesis, the information can be about the characteristics of surrounding environment, such as temperature or light intensity, obtained through Wireless Sensor Networks (WSNs), or about the characteristics of the communication system, such as hop count distance, latency or bandwidth between end-systems in the Internet. Provided the data collected by WSNs, scientist have conducted a variety of insightful research work [3–7]. Provided the data about the end-to-end connection quality in the Internet, Peer to Peer (P2P) system or Content Delivery Network (CDN) can perform more efficient server selection and, therefore, reduce download time and inter-domain traffic volume.

However, data collection of above mentioned characteristics can be costly and, sometimes even, impossible. During the data collection in WSNs, it cost energy for sensors to do measurement and data delivery. If congestions or packet loss happened or sensor hardware was damaged, the measurement would fail to reach the sink node and lost. To gather the latency or bandwidth data of certain end-to-end connection, active measurements, such as traceroute, has to be initiated in the source node. Active probing always introduces extra overhead to the infrastructure. When there are a large number of clients in the system, it become formidable to collect the data on the connection quality between all pairs of end system with active measurements.

To cope with the high cost for data collection in computer networking, data interpolation methods has been utilized to approximate the missing data with a much cheaper price. Given a number of data point obtained by sampling or experimentation, *data interpolation* methods estimate the value for an intermediate value of the independent variable. Specific techniques include curve fitting and regression analysis. Traditional interpolation methods is based on the assumption that the data is generated by a hidden function, whose close approximations can be obtained by data fitting. Then the approximation function is used to interpolate the missing value given a input of the value of the function variable. However, for the above mentioned data collection problem in computer networking, it is not straightforward to find such a hidden function to approximate. The measurements of a single sensor in a WSNs can be generated by a hidden function where the variable is the time when measurements take place. But this kind of interpolation works poorly, especially when a great percentage of readings are missing.

In this thesis, we propose novel data interpolation methods for estimating the sensor readings data in WSNs, Autonomous System Level hop counts distance data and latency and bandwidth data in the Internet.

1.1.1 AS Path Inference

The Internet is actually a network of Autonomous Systems (ASes). Each AS is owned and administered by the same organization and adheres to a single and clearly defined routing policy. AS Number (ASN) is a globally unique identifier for every AS [8]. As a result, one possible way to describe the path taken by data packets delivered in the Internet would be a series of ASNs, which is referred as AS path.

The knowledge of the actual inter-domain routing path or AS path between arbitrary pairs of end hosts is essential for network operators and researchers to detect and diagnose problems, study routing protocol behavior, characterize end-to-end paths through the Internet and optimize network performance [9]. Moreover, being aware of AS paths is beneficial for numerous network applications [10–16]

Although AS paths are of great value for many network applications, there is no oracle that can tell the AS paths between arbitrary pairs of end systems. BGP routing tables collected from vantage ASes can reveal a small portion of actual AS paths. But the number of ASes that support publicly direct access is very limited. To the best of our knowledge, only hundreds (out of totally around 47,000) ASes on the Internet can support remote access and routing information viewing [17–20]. Another way to obtain AS paths is active probing (e.g.,traceroute, iPlane [21] and iPlane Nano [22]). However, besides the direct access requirement, these active probing approaches have to deal with other issues, such as mapping between IP address to ASN, blocking from ISPs and additional overload to the infrastructure.

In Chapter 2, we introduce two new data interpolation methods for AS path inference by exploiting the underlying geometry of the Internet.

1.1.2 Environment Reconstruction in WSNs

Wireless Sensor Networks (WSNs) are able to monitor the environment of interest in much higher frequency and resolution. WSNs [3–7] have been used to collect various kinds of data, ranging from the temperature in forest to the marine pollution level in ocean. However, due to hardware damage, low battery level and/or poor condition in WSN communication, data collected by WSN often contains considerable percentage of missing readings. To interpolate the original measurements from raw (incomplete) data in WSNs, environment reconstruction [23] methods have been proposed.

In Chapter 3, we extends the state of the art environment reconstruction method by exploiting the spatio-temporal feature in WSNs and additional information obtained from probabilistic model of WSNs.

1.1.3 Rating of Network Paths Inference

Network measurement is a fundamental problem in the heart of the networking research. Over the years, various tools have been developed to acquire path properties such as round-trip time (RTT), available bandwidth (ABW) and packet loss rate, etc [24].

A practical issue of network measurement is the efficient acquisition on large networks. While cheap for a single path, it is still infeasible to rate all paths in a network by active probing due to the quadratic complexity. The scalability issue has been successfully addressed by statistical inference that measures a few paths and predicts the properties of the other paths where no direct measurements are made [25–33]. Inspired by these studies, a particular focus of this chapter is **network inference of ratings**: how ratings of network paths can be accurately predicted. Although coarse-grained, ordinal ratings are appealing for the following reasons:

- Ratings carry sufficient information that already fulfills the requirements of many applications.
- Ratings are rough measures that are cheaper to obtain than exact property values.
- Ratings can be encoded in a few bits, saving storage and transmission costs.

In Chapter 4, we investigate the rating of network paths and answer the following two questions: 1) whether the inference of ratings is accurate enough to be exploited by applications and 2) how to determine a proper granularity.

1.2 Dissertation Contributions

1.2.1 Improving AS Path Inference Accuracy

In Chapter 2, we study the AS path inference problem from a complex network's point of view.

In particular, we focus on exploring a key and intrinsic geometrical characteristic of complex networks, namely hyperbolicity or metrical tree-likeness. Roughly speaking, hyperbolicity measures the extent to which a graph resembles a tree from the metric's point of view. The key rationale for considering hyperbolicity for the AS path inference problem is that an AS system can be regarded as a complex network (i.e., a network of networks) and many complex networks (e.g., web graphs, collaboration networks, social networks and biological networks) have been empirically shown to have a low hyperbolicity or be metrically tree-like. By exploiting the property of hyperbolicity, we design an efficient AS path inference scheme.

Specifically, we make the following contributions:

- We conduct intensive empirical study with AS paths extracted from BGP control plane data to understand the extent to which actual AS paths exhibit metrical tree-likeness.
- We propose HyperPath and Valley-free HyperPath, two novel AS path inference algorithms which consider the impact of underlying geometric structure on the actual AS paths. To show the performance of the new methods, we implement two state-of-the-art benchmark methods, namely AS relationships based inference method [9] and KnownPath method [34], and compare them with the new algorithms.
- Experiments with ground truth AS paths show that our methods can be highly competitive when AS path is short and achieve significant performance gain when AS path is long with much less computation time and information. Moreover, while the benchmark techniques based on valley-free property frequently fail to work when actual AS paths are with 6 hops or more, the new inference algorithms can still achieve impressive prediction accuracy.
- We show that the improvement of AS path prediction accuracy by our methods can reduce inter-AS traffic on BitTorrent network [35].

1.2.2 Improving Environment Reconstruction Accuracy in Sensor Network

In Chapter 3, we extends the state of the art environment reconstruction method by exploiting the spatio-temporal feature in WSNs and additional information obtained from probabilistic model of WSNs.

Different kinds of prior knowledge have been exploited in existing solutions to optimize the speed and accuracy of signal reconstruction. A recent proposal — Compressive Sensing (CS [36]) exploits sparsity for efficient reconstruction and has become a key technique in today's signal processing systems. It can be adapted in WSN since the measurement matrices also have sparse structure in their singular values, but a straightforward adaptation is not enough. Studies have observed that many natural signals have features in addition to sparsity, e.g., structure [37], clustering property in image [38], etc. A recent research in WSN (ESTI-CS [39]) exploits strong time stability and spatial correlation together with sparsity to improve the accuracy of reconstruction. However, we show that by exploiting more features we can further improve the performance of reconstruction in WSNs.

In Chapter 3, we propose Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing (PMEST-CS) that extends ESTI-CS by utilizing two kinds of prior knowledge: 1) with analysis on real datasets, we show that the spatio-temporal feature of WSN data *is* sparse and can be exploited further, and 2) we find that statistical inference on the probabilistic model can provide us a rough guess with a confidence level on the missing readings which can enhance the overall accuracy.

We also realized that the probabilistic model is a critical component in our solution. Therefore, we design a tree based Markov Random Field (MRF) that takes both temporal and spatial correlation of environment into consideration. Furthermore, we train the MRF from WSN data to improve model quality. One challenge raised here is that standard learning scheme cannot scale when the feeding WSN data is incomplete. Therefore, we also propose a new algorithm that can build a qualified MRF out of highly incomplete data.

Specifically, we make the following contributions:

- We propose a new compressive sensing optimization problem which exploits the sparsity in the spatio-temporal difference and leverages prior knowledge from a probabilistic model.
- To overcome the limitations of existing probabilistic models, we design an MRF model which incorporates both spatial and temporal correlation in the environment. To cope with the highly incomplete data in WSNs, we propose a new learning algorithm for MRF. Our evaluation results show that the proposed learning algorithm can generate highly effective MRF models from data even with 60% of missing readings.

- We perform intensive quantitative analysis to show that our solution can outperform the state of the art approach (ESTI-CS) by 30% in terms of accuracy.

1.2.3 Improving Locality-Awareness in Overlay Network Construction and Routing

In Chapter 4, we investigate the rating of network paths and answer the following two questions: 1) whether the inference of ratings is accurate enough to be exploited by applications and 2) how to determine a proper granularity.

An interesting observation is that the inference problem resembles the problem of *recommender systems* which studies the prediction of preferences of users to items [40]. If we consider a path property as a “friendship” measure between end nodes, then intelligent peer selection can be viewed as a “friend” recommendation task. This seemingly trivial connection has the great benefit to leverage the rapid progresses in machine learning and investigate the applicability of various solutions to recommender systems for network inference.

Another practical issue on rating-based network measurement is **the usability in applications**. Two questions need to be answered, the first of which is whether the inference of ratings is accurate enough to be exploited by applications and the second of which is how to determine a proper granularity. While a coarser granularity means rougher and thus cheaper measurement, it also means more information losses which may hurt the performance of applications. Answers to these questions are critical in the design of system architecture, particularly for P2P applications where the knowledge of locality plays an important role [35,41,42].

Thus, we answer these two questions by investigating quantitatively the impacts of both the inaccuracy of the inference and the granularity. For the case study, we consider locality-aware overlay construction and routing where locality refers to the proximity between network nodes according to some path property such as RTT or ABW. More specifically, we performed the study on Pastry [42] and BitTorrent [35], which are typical structured and unstructured overlay networks and are known to enjoy the property of locality awareness, and evaluated the performance of overlay construction and routing, with the knowledge of locality obtained via network inference of ratings. Our studies show that while the knowledge of inferred ratings can improve the performance of peer selection, finer granularities do not always lead to larger improvements. For example, our simulations on various datasets show that the performance of peer selection improves very little when the rating level reaches 2^4.

Specifically, we make the following contributions:

- We investigate the rating-based network measurement that acquires quantized path properties represented by ordinal numbers. Such representation not only is informative but also reduces measurement, storage and transmission costs.

- We investigate the scalable acquisition of ratings by network inference. We highlight similarities between network inference and recommender systems and examine the applicability of solutions from this latter domain to network inference. In particular, we show that our inference problem can be solved by a class of matrix factorization techniques.
- We perform a case study on locality-aware overlay construction and routing to demonstrate the usability of rating-based network measurement and inference in P2P applications.

1.3 Dissertation Overview

This thesis contains part of the content of the following published and submitted papers.

- Narisu Tao, Xu Chen, Xiaoming Fu, AS Path Inference from Complex Network Perspective. IFIP Networking 2015, May 2015.
- Narisu Tao, Xu Chen, Farshid Hassani Bijarbooneh, Wei Du, Edith Ngai, Xiaoming Fu, Probabilistic Model Enhanced Compressive Sensing for Environment Reconstruction in Sensor Networks. INFOCOM 2016, April 2016. (under submission)
- Wei Du, Yongjun Liao, Narisu Tao, Pierre Geurts, Xiaoming Fu, Guy Leduc, Rating Network Paths for Locality-Aware Overlay Construction and Routing. IEEE/ACM Transactions on Networking, July 2014.

The remainder of this dissertation is organized as follows: Chapter 1 provides an overview of this thesis: introducing the problem and the challenges and stating the contributions and the structure of this thesis. Chapter 2, based on our first publication as mentioned above, describes our work on improving the AS path inference accuracy by exploiting the metric tree-likeness of AS level topology of the Internet. Chapter 3, based on our second submitted paper as mentioned above, describes our work on improving the environment reconstruction in WSN with probabilistic model enhanced compressive sensing approach. Chapter 4, based on our third publication as mentioned above, describes our work on improving the locality awareness of structured and unstructured overlay network with matrix completion approach. Chapter 5 summarizes this thesis.

Chapter 2

AS Path Inference from Complex Network Perspective

AS-level end-to-end paths are of great value for ISPs and a variety of network applications. Although tools like traceroute may reveal AS paths, they require the permission to access source hosts and introduce additional probing traffic, which is not feasible in many applications. In contrast, AS path inference based on BGP control plane data and AS relationship information is a more practical and cost-effective approach. However, this approach suffers from a limited accuracy and high traffic, especially when AS paths are long.

In this chapter, we bring a new angle to the AS path inference problem by exploiting the metrical tree-likeness or low hyperbolicity of the Internet, part of the complex network properties of the Internet. We show that such property can generate a new constraint that narrows down the searching space of possible AS paths to a much smaller size. Based on this observation, we propose two new AS path inference algorithms, namely HyperPath and Valley-free HyperPath. With intensive evaluations on AS paths from real-world BGP Routing Information Bases, we show that the proposed new algorithms can achieve superior performance, in particular, when AS paths are long paths. We demonstrate that our algorithms can significantly reduce inter-AS traffic for P2P applications with an improved AS path prediction accuracy.

Contents

2.1 Introduction **11**

2.2 Related Work **13**

2.3 δ-hyperbolicity: Tree-likeness from Metric Point of View **14**

2.3.1 Definition 15

2.3.2 Low Hyperbolicity of Scale-free Networks 15

2.4 HyperPath Method for AS Path Inference **17**

2.4.1 Data Collection and Analysis 17

2.4.2 Algorithms 18
2.4.3 Discussion 20
2.5 Evaluation **21**
2.5.1 Benchmark Methods 22
2.5.2 Experiment Set-up 23
2.5.3 Estimation Accuracy 24
2.5.4 Application: Inter-domain Traffic Reduction for BitTorrent P2P System 27
2.6 Chapter Summary **28**

2.1 Introduction

As a network of networks, the Internet infrastructure consists of tens of thousands of networks or Autonomous Systems (ASes). Each AS, as a part of the Internet, is owned and administered by the same organization and adheres to a single and clearly defined routing policy. AS Number (ASN) is a globally unique identifier for every AS [8]. AS path is a series of ASNs, representing the route taken by data packets sent from one AS to a certain network and originally exchanged by neighboring ASes to avoid loops in inter-domain routing.

The knowledge of the actual AS path between arbitrary pairs of end hosts directly reflects the topological property of the connection. Therefore it is essential for network operators and researchers to detect and diagnose problems, study routing protocol behavior, characterize end-to-end paths through the Internet and optimize network performance [9]. Moreover, many network applications can benefit from being aware of AS paths. For example, it has been shown that most bottleneck links are more likely to appear in the access network or on the links between ISPs, rather than in the backbones of the ISPs [10]. Therefore, preferring the peers or servers with a shorter AS path can reduce chances of having bottlenecks in the path and, in turn, improve performance of applications (e.g., P2P), reduce the inter-domain traffic and lower cost for ISPs. With this motivation, J. Li and K. Sollins have proposed a structured P2P network, in which AS hop counts are used to filter out unlikely candidates [11]. This proposed system significantly reduces network traffic while maintaining fast lookups. As another example, AS path information has been leveraged for improving QoS of the VoIP service (e.g., Skype) [12]. In addition, AS path information has also been used for network delay estimation [13], cache deployment in Content Delivery Networks (CDNs) [14] and assessment of Internet routing resilience to failures and attacks [15, 16].

Although AS paths are of great value for many network applications, how to obtain such information is still a challenging issue. Collecting the BGP routing tables directly is impractical, since the number of ASes that support public direct access is very limited. To the best of our knowledge, only hundreds (out of totally around 47,000) ASes on the Internet can support remote access and routing information viewing [17–20]. Another way to obtain AS paths is active probing (e.g.,traceroute, iPlane [21] and iPlane Nano [22]). However, besides the direct access requirement, these active probing approaches have to deal with other issues, such as mapping between IP address to ASN, blocking from ISPs and additional overload to the infrastructure. A more practically-relevant and cost-effective approach is to estimate the AS paths by inference techniques based on BGP control plane data and AS relationship information [9, 34]. However, traditional inference-based approaches suffer from limited accuracy, especially when AS paths are long.

In this chapter, we study the AS path inference problem from a complex network's point of view. In particular, we focus on exploring a key and intrinsic geometrical characteristic of complex networks, namely hyperbolicity or metrical tree-likeness. Roughly speaking, hyperbolicity measures the extent to which a graph resembles a tree from the metric's point of view. The key rationale for considering hyperbolicity for the AS path inference problem is that an AS system can be regarded as a complex network (i.e., a network of networks) and many complex networks (e.g., web graphs, collaboration networks, social networks and biological networks) have been empirically shown to have a low hyperbolicity or be metrically tree-like.

In this chapter, we leverage the property of hyperbolicity to design an efficient AS path inference scheme. To this end, we address the following main challenges:

- AS path inference problem is complicated by the fact that information collected from the current routing system is highly incomplete [43].
- Hyperbolicity is only studied under the shortest path distance metric of graph models of communication networks [44–47]. However, due to the policy-based inter-domain routing, actual AS path is not necessarily the shortest path and usually longer than the shortest path [48]. With the actual AS path hop count as the distance function, whether the AS-level Internet still exhibits metrical tree-likeness and to which extent it follows remain open questions.
- If the actual AS paths respect the underlying geometry of the Internet, how can we leverage this fact to improve AS path inference technique?

To tackle the above-mentioned challenges, we first conduct intensive empirical study with AS paths extracted from BGP control plane data to understand the extent to which actual AS paths exhibit metrical tree-likeness. Then we propose HyperPath and Valley-free HyperPath, two novel AS path inference algorithms which consider the impact of underlying geometric structure on the actual AS paths. To show the performance of the new methods, we implement two state-of-the-art benchmark methods, namely AS relationships based inference method [9] and KnownPath method [34], and compare them with the new algorithms. Experiments with ground truth AS paths show that our methods can be highly competitive when AS path is short and achieve significant performance gain when AS path is long with much less computation time and information. Moreover, while the benchmark techniques based on valley-free property frequently fail to work when actual AS paths are with 6 hops or more, the new inference algorithms can still achieve impressive prediction accuracy. We also show that the improvement of AS path prediction accuracy by our methods can reduce inter-AS traffic on BitTorrent network [35].

The remainder of the chapter is organized as follows. In Section 2.2, we introduce related work. In Section 2.3, we introduce the concept of δ-hyperbolicity of graphs and illustrate with synthetic network models. In Section 2.4, we conduct empirical study to understand the impact of underlying geometry of the Internet on AS path and propose new algorithms. In Section 2.5 we evaluate the new methods and conclude the chapter in Section 2.6.

2.2 Related Work

Active probing from an end host in a source AS can reveal the AS path. By running traceroute, a series of IP addresses of router interfaces would be obtained. Mapping these IP addresses into ASNs can give us a raw AS path. After removing the repeatedly occurring ASNs, the AS path can be finally generated. Although active probing can deliver accurate AS path, it can be problematic in practice. Firstly, traceroute can be blocked by ISPs for security consideration. Secondly, the mapping from IP address to AS number is not always accurate. Thirdly, it introduces additional measurement overhead into the infrastructure. Finally, the biggest problem is that it requires direct access to the end host in the source AS, which is usually hard to achieve.

To deal with the lack to direct access of active probing method, a plethora of different techniques have been proposed for inferring AS path. The most straightforward way is to run the shortest path algorithm, such as Dijkstra's algorithm, on the AS topology generated from BGP routing information as an approximation [11]. However, due to the inflation of AS paths, this method cannot provide high accuracy [48].

Later, AS relationships were introduced to design better AS path inference methods [9, 34, 49]. Specifically, AS relationships between two connected ASes can be classified into the following three types: customer to provider (c2p), peer to peer (p2p) or sibling to sibling (s2s). In a c2p relationship, the customer pays the provider to obtain transit service through the provider's network, while, in a p2p relationship, it is assumed that two peering ASes share the deployment and maintenance cost for the connecting link. Siblings are peering ASes that generally have a mutual transit agreement, i.e., merging ISPs.

Gao et al. [50] pointed out that patterns of AS path should follow, the so called, valley-free property. The valley-free property stems from the fact that ASes don't want to be used as a transit. Gao et al. characterize a path as downhill (uphill) if it only contains p2c or s2s links (c2p or s2s links) and any valid (valley-free) path must match one of the following patterns [50]:

- An uphill path;
- A downhill path;
- An uphill segment followed by a downhill segment;
- An uphill segment followed by a p2p link;

- A p2p link followed by a downhill segment;
- An uphill segment followed by a p2p link, followed by a downhill segment.

Mao et al. proposed one of the first methods to infer arbitrary AS path from the BGP routing tables [9]. Their method filter out the AS paths violating the valley-free property and choose the shortest AS path from the remaining AS paths. Later, Qiu and Gao [34] proposed the KnownPath algorithm for AS path inference. The key idea of the KnownPath method is to exploit AS paths that have appeared in BGP routing tables. This method has been cited by many recent research papers as one of the state-of-the-art method for AS path inference based on BGP control plane data [13, 51, 52]. One weak point of inference based on AS relationships information is that AS relationships can contain errors. In fact, inference about AS relationships itself is an active research problem [50, 53, 54].

In this chapter, we enrich the AS path inference techniques by introducing a new kind of constraint or filtering mechanism, which is similar to the role the valley-free property plays. The new constraint mainly narrows down the candidate AS path set to a much smaller size. Without this filtering of possible AS path sets, originally, we have to go through every possible path connecting source AS and destination prefix, which can be $O(|V|^2)$ in AS relationships based inference method, where V is the number of ASes in an AS topology. Therefore, the new constraint can enable speed-up in inference time. In addition, even though the new methods infer with much less information input, they can be highly competitive and even outperform the benchmark methods. Finally new inference methods can be implemented in a distributed manner, which is not easy for the state-of-the-art methods.

The hyperbolic space has been used for distance embedding and greedy routing for communication networks [44–47]. These methods are based on a given topology and a shortest path length distance function. The main difference between our methods and the existing studies is that we investigate and leverage the hyperbolicity of a metric space where the distance function is the actual AS hop count, rather than the shortest path distance.

2.3 δ-hyperbolicity: Tree-likeness from Metric Point of View

To facilitate discussions, in this part, we first give a brief introduction to the definition of hyperbolicity.

2.3.1 Definition

The notion of δ-hyperbolicity comes from the field of geometric group theory and the geometry of negatively curved metric spaces [55, 56]. Intuitively speaking, hyperbolicity of a graph/network can be viewed as a measure of how close a graph is to a tree from a metric point of view.

There are two definitions of hyperbolicity, which are equivalent to each other up to a multiplicative constant. In this chapter, we use the 4-point δ-hyperbolicity definition by Gromov [55].

Definition 2.1 (). *[55] Let $\delta \geq 0$. A metric space (X,d), where X is the set of points and d is the distance measure, is called δ-hyperbolic if and only if given quadruplet x,y,u,v $\in X$ satisfying that $d(x,y)+d(u,v) \geq d(x,u)+d(y,v) \geq d(x,v)+d(y,u)$, the following condition holds:*

$$(d(x,y)+d(u,v))-(d(x,u)+d(y,v)) \leq 2 * \delta \tag{2.3.1}$$

For a graph $G=(V,E)$, we can regard it as a metric space where $X=V$ and d is the graph distance (e.g., shortest path distance) between two vertices u and v in the graph G. Then, the hyperbolicity δ of the graph G is typically defined as the minimum value of δ and the metric space (V,d) based on graph G is δ-hyperbolic.

A key property of hyperbolicity is that it can characterize the metrical tree-likeness of a graph. Generally, the lower the hyperbolicity of a graph is, the more likely it is metrically to a tree. For example, trees are exactly 0-hyperbolic. A cycle of length $2k$ is $\frac{k}{2}$-hyperbolic, which is the largest hyperbolicity a finite graph with $2k$ vertexes can have. It has been empirically shown that many real-world graphs/networks, such as collaborative graphs, email networks, biological networks, web graphs, p2p networks and social networks, have low hyperbolicity [57, 58].

2.3.2 Low Hyperbolicity of Scale-free Networks

In this section, we will demonstrate the metrical tree-likeness of scale-free networks by using the measure of hyperbolicity through numerical evaluation. Note that this is useful for our study later since the AS topology is also scale-free as shown in Section 2.4. The scale-free networks are generated according to the H^2 model [59], which is one of the latest scale-free network generation models. The generated networks by the H^2 model exhibit many similar properties of real-world complex networks. The H^2 model requires input parameters, such as the node number (N), the average node degree (d), the exponent of the power law distribution of the node degrees (γ) and the temperature (T).

We generate three synthetic scale-free networks (i.e., S_1, S_2, S_3) according to the parameters in Table I. Note that we choose γ as 2.1, which is the exponent of the power distribution of node degrees in the AS topology observed in our numerical study.

To gain more useful insight of the network structures, we compute the δ-hyperbolicity value distribution of the Largest Connected Component (LCC) of each network. $O(N^4)$ number of quadruplets has to be exhaustively iterated to obtain a complete δ-hyperbolicity value distribution. It is computationally prohibitive to obtain such a complete distribution when networks are of tens of thousands of nodes. Therefore, we randomly sample 100 million quadruplets to approximate the distribution when a network has more than one thousand nodes.

Table 2.1: Synthetic scale-free networks.

ID	$\|V\|$	$E(d)$	γ	T	$\|V\|$ in LCC	$\|E\|$ in LCC
S_1	100	60	2.1	0	100	293
S_2	1,000	25	2.1	0	875	3,435
S_3	10,000	30	2.1	0.8	8,952	31,058

Table 2.2: Distribution of the δ-hyperbolicity value of quadruplets in different graphs.

ID δ	(S_1,d)	(S_2,d)	(S_3,d)	(T,d_2)
0	0.838	0.932	0.724	0.460
0.5	0.162	0.068	0.275	0.430
1	1.64E-06	1.88E-06	0.002	0.093
1.5	-	-	2.20E-07	0.015
2	-	-	-	0.002
2.5	-	-	-	1.41E-04
3	-	-	-	1.75E-05
3.5	-	-	-	6.73E-07
4	-	-	-	5.34E-09
$\% \leq 1$	1.000	1.000	0.999	0.983

Table 2.2 shows that scale-free networks (i.e., (S_1,d), (S_2,d), (S_3,d) in Table 2.2) are metrical tree-like and almost every quadruplet has a δ-hyperbolicity smaller than or equal to one.

2.4 HyperPath Method for AS Path Inference

As mentioned in Section 2.3.1, the graph hyperbolicity is typically defined under the shortest path distance metric. But due to AS path inflation [48], the actual AS path is usually not the shortest one. In this case, whether the space (T, d_2), where T is ASes set and d_2 is actual AS hop count distance, is hyperbolic or not is not explored yet.

To understand to which extent (T, d_2) exhibits metrical tree-likeness, in this section, we conduct a data driven analysis on AS paths obtained from real-world BGP control plane data.

2.4.1 Data Collection and Analysis

To facilitate the data analysis, we need a large survey of ground truth AS paths set. To obtain this set, we use a collection of BGP tables (collected on 08:00 AM UTC on August 29, 2013) obtained from the RouteViews [18] and RIPE [19] repositories. Although we only consider one snapshot data in this study, a brand new snapshot on BGP tables is available in every two hours and an additional update is available in every fifteen minutes [18, 19]. From the BGP routing tables, we can extract AS paths. Each AS path is a path from a source AS, via a set of intermediate ASes, to a destination IP prefix. For example, the AS path from the AS680 (German National Research and Education Network) to the IP prefix of 65.169.169.0/24 in U.S. is $AS680 \rightarrow AS6939 \rightarrow AS6598 \rightarrow AS25612$. Note that the IP prefixes 65.169.169.0/24 belongs to the $AS25612$.

The full dataset is collected from 389 unique monitors; it consists of over 60 million AS paths and contains at least $646,567$ unique destination prefixes. The AS topology obtained from the AS paths data includes $48,133$ ASes and $164,883$ links. The degree distribution of the AS topology is given in figure 2.1, which is scale-free and follows a power law distribution.

Since part of the monitor-to-prefix paths is missing, we hence filter out the monitors with few known AS paths to IP prefixes, leading to 70 out of 389 monitors selected. All of these 70 monitors can simultaneously reach $30,000$ distinctive IP prefixes, which are from more than $7,000$ different ASes.

Moreover, by accounting the paths originated from one of the vantage ASes and ending with prefixes only appeared in one individual AS, the final ground truth AS paths set contains 2,446,644 AS paths.

Note that, to get the AS hop count, we don't treat the multiple occurrences of the same AS

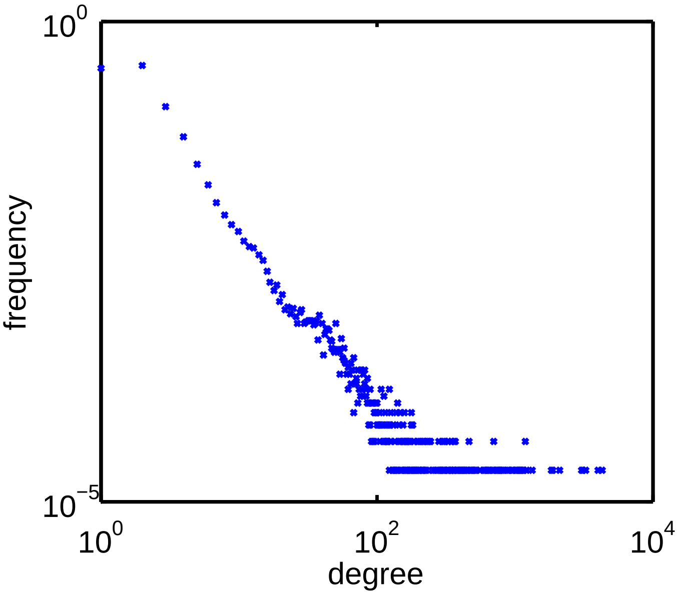


Figure 2.1: Power law distribution of node degrees in the AS topology.

as multiple hops. In other words, the AS hop count is equal to the number of the distinctive ASes in the AS path minus one.

Using the dataset above, we compute the δ-hyperbolicity value distribution based on a sample set of hundreds of millions of quadruplets in the largest connected component of AS topology graph (T, d_2). The result is given in the last column of table 2.2. We can see that (T, d_2) is indeed metrically tree-like with most quadruplets having δ value smaller than or equal to one.

2.4.2 Algorithms

Motivated by the observation that AS topology (T, d_2) is metrically tree-like (i.e., low hyperbolicity), we then propose AS path inference algorithms accordingly. To proceed, we first introduce the following definitions.

Definition 2.2 (). *We denote the shortest distance between two points $x, y \in X$ by $|x-y|$. If $x \in X$ and $A \subseteq X$ then*

$$dist(x,A) = \inf\{|x-y| : y \in A\}. \tag{2.4.1}$$

Definition 2.3 (). *For $\varepsilon > 0$ the open ε-neighborhood $N_\varepsilon(A)$ of a set $A \subseteq X$ is*

$$N_\varepsilon(A) = \{x \in X : dist(x,A) < \varepsilon\} \tag{2.4.2}$$

According to the property of the δ-hyperbolicity [56], all triangles in the space are δ-thin, i.e. for all $x,y,z \in X$ and segments $[x,y]$,$[x,z]$ and $[y,z]$, we have

$$[x,y] \subseteq N_\delta([x,z]) \cup N_\delta([y,z]). \tag{2.4.3}$$

For the AS topology space (T, d_2), this property implies that, given two AS paths rooted from the same origin to two different destinations ASes, the ground truth AS path between two destinations ASes should be in δ-neighborhood of these two paths. Based on the property above, we then propose an AS path inference algorithm. The key idea is to construct an AS path that is within the δ-neighborhood of the AS path we want to know.

To construct such an AS path, let's first look at AS paths obtained from BGP control plane data. There are hundreds of vantage ASes and each has AS paths from itself to hundreds of thousands of IP prefixes. The entire AS paths originated from every vantage AS can make up a sub-graph of the AS topology. This sub-graph can include loops, so it is not a spanning tree of the original graph. But, still, every pair of AS paths from the same vantage AS n_v to two different IP prefixes prefix_1 and prefix_2 always split at a certain node which we call a branching point, denoted by n_b. Note that, while the two paths may have several branching points, we only consider the first one. Assuming that two paths are $p = n_v \rightarrow \cdots \rightarrow n_b \cdots \rightarrow n_1 \rightarrow \text{prefix}_1$ and $q = n_v \rightarrow \cdots \rightarrow n_b \cdots \rightarrow n_2 \rightarrow \text{prefix}_2$, we define the following function to construct a path to approximate the ground truth AS path:

$$\phi_{n_v}(p,q) = n_1 \rightarrow \cdots \rightarrow n_b \rightarrow \cdots \rightarrow n_2. \tag{2.4.4}$$

Figure 2.2 shows a simple example, where the vantage AS is AS10026 and the paths to two different IP prefixes are $p = \text{AS10026} \rightarrow \text{AS174} \rightarrow \text{AS39792} \rightarrow 37.140.192.0/22$ and $q = \text{AS10026} \rightarrow \text{AS174} \rightarrow \text{AS2914} \rightarrow \text{AS8151} \rightarrow 189.245.128.0/19$. AS174 is the branching point and $\phi_{\text{AS10026}}(p,q) = \text{AS39792} \rightarrow \text{AS174} \rightarrow \text{AS2914} \rightarrow \text{AS8151}$.

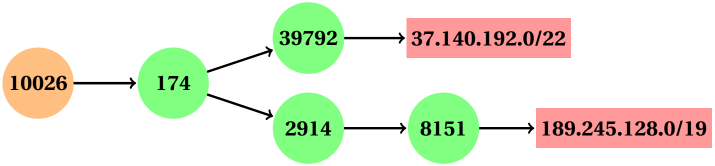


Figure 2.2: Two paths example.

In practice, we can have k pairs of AS paths(p_i,q_i) that are originated from multiple vantage ASes $n_{v_i}, i = 1,\ldots,k$ to IP prefix$_1$ and IP prefix$_2$. In this case, suppose that each $\phi_{n_{v_i}}(p_i,q_i), i = 1,\ldots,k$ hits ground truth AS path with a probability P_k independently, the probability that all $\phi_{n_{v_i}}, i = 1,\ldots,k$ fail to hit the AS path would be as the following:

$$P_{\emptyset} = \prod_{i=1}^{k}(1-P_k) \tag{2.4.5}$$

$P_{\emptyset}$ decreases exponentially as the number of vantage ASes increase. One straightforward way to incorporate estimation from multiple vantage ASes would be to choose the $\arg\min_{\phi_{n_{v_i}}} |\phi_{n_{v_i}}|, i \in [1,\ldots,k]$ as the estimation. The match rate of this method is equal to the probability that at least one of the $\phi_{n_{v_i}}$ hits the AS path, which is $1-P_{\emptyset}$. Based on this simple idea, HyperPath algorithm is given in Algorithm 1.

For the HyperPath algorithm, we do not require AS relationship information. When AS relationship information is taken into account, we develop the Valley-free HyperPath algorithm. It is an extension of the HyperPath algorithm by integrating the valley-free property and is given in Algorithm 2. The idea is to consider two constraints (i.e., valley-free property and low hyperbolicity of the Internet) together to filter possible AS paths. When the valley-free property fails to work, we return the AS path that only considers low hyperbolicity in the inference process.

2.4.3 Discussion

Comparisons between the different AS path inference methods from complexity and information requirement aspects are given in Table 2.3. It shows that our proposed algorithms require less information and demand lower computation complexity. Note that, in Table 2.3, $|V|$ and $|E|$ are the total numbers of nodes and links in the AS topology respectively.

Because the HyperPath algorithm and the Valley-free HyperPath algorithms only consider dozens of constructed paths recorded by the vantage ASes, the computational complexity of both algorithms are $O(K)$. Here K is the number of vantage ASes (around a few

Algorithm 1 HyperPath Algorithm

```
INPUT: $k$ pairs of AS paths $(p_i, q_i), i = 1, \ldots, k$. $p_i$ can reach prefix$_1$ and $q_i$ can reach
    prefix$_2$. Both paths are originated from vantage AS $n_i$.
OUTPUT: Inferred AS path $\hat{p}$ between prefix$_1$ and prefix$_2$.
$\hat{b} = +\infty$; $\hat{p} = \emptyset$
for $i = 1$ to $k$ do
    path= $\phi_{n_i}(p_i, q_i)$
    if $\hat{b} \geq HopCount(\text{path})$ then
        $\hat{b} = HopCount(\text{path})$
        $\hat{p} = \text{path}$
    end if
end for
return $\hat{p}$
```

hundreds), which is much smaller than the number of all ASes in the AS topology. If two end hosts store the AS paths set from the vantage ASes to the networks they are sitting in locally, our methods make it possible for them to infer the AS path connecting them by exchanging the AS paths sets. However, the benchmark methods need to build entire AS topology locally to do inference and, therefore, have to iterate through a much bigger search space. As a result, these methods require higher computational complexity and demand more information. Moreover, our methods are able to infer certain individual AS path between two end hosts at a time. In contrary, to infer certain individual AS path, one of the benchmark methods, KnownPath method, has to infer all AS paths from one node to all other nodes in the graph, even when they are not required.

2.5 Evaluation

In this section, we evaluate the performance of the two proposed methods with realistic AS paths data. We will use two state-of-the-art methods (i.e., AS relationships based inference algorithm and KnownPath algorithm) as the benchmark. In addition, we also implement the no policy method (shortest path heuristic) as the baseline method. Experiment set-up details and evaluation result will be discussed after the introduction of the benchmark methods.

Algorithm 2 Valley-free HyperPath Algorithm

```
INPUT: $k$ pairs of AS paths $(p_i, q_i), i = 1, \dots, k$. $p_i$ can reach prefix$_1$ and $q_i$ can reach
    prefix$_2$. Both paths are originated from vantage AS $n_i$; AS relationships information on
    each edge appeared in the AS paths.
OUTPUT: Inferred AS path $\hat{p}$ between prefix$_1$ and prefix$_2$.
$\hat{b_1} = +\infty$ ; $\hat{b_2} = +\infty$; $\hat{p_1} = \emptyset$; $\hat{p_2} = \emptyset$
for $i = 1$ to $k$ do
    path $= \phi_{n_i}(p_i, q_i)$
    if $isValidPath$(path) and $\hat{b_1} \geq HopCount$(path) then
        $\hat{b_1} = HopCount$(path)
        $\hat{p_1} =$ path
    end if
    if $\hat{b_2} \geq HopCount$(path) then
        $\hat{b_2} = HopCount$(path)
        $\hat{p_2} =$ path
    end if
end for
if $\hat{b_1} \neq +\infty$ then
    $\hat{p} = \hat{p_1}$
else
    $\hat{p} = \hat{p_2}$
end if
return $\hat{p}$
```

2.5.1 Benchmark Methods

2.5.1.1 AS Relationships based Inference Algorithm

This algorithm is one of the pioneers and the most cited work on AS path inference algorithm [9]. The key idea of this method is to filter out the paths that don't satisfy the valley-free property and to find the shortest AS path from the remaining valid AS paths set. The algorithm is given in Algorithm 3.

2.5.1.2 KnownPath Algorithm

As an extension of the AS relationships based inference algorithm, besides using the valley-free property, this algorithm improves the inference accuracy by further integrating the AS paths that are already observed from the vantage ASes. The algorithm is detailed in Algo-

Table 2.3: Comparisons between the different AS path inference methods.

	Baseline method	HyperPath method	AS relationships based method	KnownPath method	Valley-free HyperPath method
Time complexity from one node to all nodes	$O(\|E\|log(\|V\|))$	$O(\|V\|K)$	$O(\|V\|^3)$	$O(\|V\|\|E\|)$	$O(\|V\|K)$
Time complexity from one node to another node	$O(\|E\|log(\|V\|))$	$O(K)$	$O(\|V\|^2)$	$O(\|V\|\|E\|)$	$O(K)$
AS topology information required	global	local	global	global	local
AS relationships information required	no	no	yes	yes	yes

rithm 4, in which rib_in(u)[p] is a path set that contains all the feasible paths from AS u to a specific IP prefix p learned from u's neighbors. The baseASset contains the ASes that have the assured paths from themselves to the prefix p.

2.5.1.3 No Policy Baseline Method

We also implement the no policy method as the baseline method. In no policy method, the actual AS path is approximated by the shortest AS path in the AS topology obtained from BGP control plane data.

2.5.2 Experiment Set-up

Algorithm Input:

- AS paths: we use the 2.4 million ground truth AS paths to do the evaluation, which has been introduced in Section 2.4. Besides that, we have also AS paths from 70 vantage ASes to feed our proposed algorithms.
- AS topology: To mimic the case where we don't have access to the routing table of the ASes, we build the AS topology out of the BGP control plane data of the 69 ASes, excluding the one we are interested in. These AS topologies are the only

input required by the no policy method. In comparison, our methods don't need to construct the entire topology locally. For the AS relationships based inference method and knownPath method, the AS relationships information on links in the AS topology is also required.

- AS relationship: We use the AS relationships data from Caida's Inferred AS Relationships Dataset [60]. This data is of as high as 97% accuracy [53]. Although it is possible to generate the AS relationships data of the same day when we collected BGP data, we use the AS relationships data on 1st of September 2013 in our study, simply because Caida's AS relationships data is only available on the first day of each month. The date on which AS relationships data is generated is two days later than the date on which BGP control plane data is collected. But we still use this AS relationships data in our study, assuming that most of the AS relationships would not change dramatically and remain almost the same within several days.

To achieve a fair comparison, we organize the experiments in two categories by considering the cases with and without AS relationships information.

- Comparisons without considering AS relationship: In this part, we compare against HyperPath with no policy baseline method. Both methods don't require AS relationship information to do estimation.
- Comparisons with considering AS relationship: In this part, we compare against Valley-free HyperPath with the benchmark methods. In addition to AS path information, all of them take the AS relationships information into account.

2.5.3 Estimation Accuracy

Similar to many studies in the literature [9, 34], we evaluate the methods' performance based on the hop count number of AS paths and present the prediction accuracy in the form of confusion matrices.

- Comparisons without considering AS relationships: Table 2.4 and 2.5 show the prediction accuracy of both the HyperPath method and the no policy baseline method. We observe that the HyperPath method achieves similar performance as the baseline method when AS paths are short (e.g., hop counts are smaller than or equal to 2). HyperPath method achieves significant performance improvement over the baseline method when AS paths are long (e.g., hop counts are greater than 2). For example, when AS path hop counts are greater than 3, the HyperPath method possesses more than 50% prediction accuracy, while the baseline method only has an accuracy of 27%.

Table 2.4: Confusion matrices of the prediction performance of the baseline method.

%		Predicted hop count							
		1	2	3	4	5	6	7	8
Actual hop count	1	**50.7**	49.1	0	0	0	0	0	0
	2	2.2	**83.4**	14.4	0	0	0	0	0
	3	0.9	30.2	**67.1**	1.9	0	0	0	0
	4	0.5	13.2	58.3	**26.9**	1.2	0	0	0
	5	1.1	7.9	27.0	38.9	**24.8**	0.3	0	0
	6	0	2.7	16.3	17.9	31.9	**26.1**	5.2	0
	7	0	1.2	1.2	3.5	1.5	67.3	**25.4**	0
	8	0	0	30.7	11.4	17.6	1.1	39.2	**0**

Table 2.5: Confusion matrices of the prediction performance of the HyperPath method.

%		Predicted hop count							
		1	2	3	4	5	6	7	8
Actual hop count	1	**48.1**	51.6	0.3	0	0	0	0	0
	2	0.1	**78.0**	21.8	0	0	0	0	0
	3	0	10.3	**84.0**	5.7	0	0	0	0
	4	0	4.0	32.7	**60.1**	3.2	0	0	0
	5	0	0.7	10.2	30.0	**57.5**	1.6	0	0
	6	0	0.1	3.1	7.4	25.2	**58.6**	5.6	0
	7	0	0	1.3	1.5	1.1	2.3	**92.3**	1.4
	8	0	0	1.1	1.1	12.0	6.9	0.6	**78.3**

- Comparisons with considering AS relationships: Table 2.6,2.7 and 2.8 show the prediction accuracy of AS relationships based method, KnownPath method and Valley-free HyperPath method. Similar to the case where AS relationship information is not considered, we also observe similar performance among the different methods when AS paths are short (e.g., hop counts are smaller than or equal to 3). The valley-free HyperPath method, however, achieves significant performance improvement over the benchmark methods when AS paths are long (e.g., hop counts are greater than 3). For example, when AS path hop counts are greater than 6, the Valley-free HyperPath method achieves more than 70% prediction accuracy, while the accuracy of the benchmark methods drops significantly, with the accuracy of less than 17%. Moreover, as reported in [9,34], we also observe (see the columns of "N/A" in Table 2.6,2.7 and 2.8) that benchmark methods could fail to return estimation values, in particular, when AS paths are long. By contrast, the Valley-free HyperPath method is more robust and doesn't suffer from this issue.

Table 2.6: Confusion matrices of the prediction performance of the AS relationships based method.

%		Predicted hop count								
		1	2	3	4	5	6	7	8	N/A
Actual hop count	1	**50.7**	43.7	5.4	0.1	0	0	0	0	0
	2	1.8	**81.6**	15.8	0.7	0	0	0	0	0.1
	3	0.9	18.8	**77.1**	3.0	0.1	0	0	0	0.2
	4	0.5	8.9	41.3	**47.6**	1.5	0	0	0	0.2
	5	1.1	6.1	19.5	38.8	**31.6**	0.3	0	0	2.7
	6	0	2.5	14.2	17.5	16.5	**2.9**	0	0	46.5
	7	0	1.2	0.7	3.6	1.1	0.1	**0**	0	93.4
	8	0	0	2.3	0.6	17.6	1.1	0	**0**	78.4

Table 2.7: Confusion matrices of the prediction performance of the KnownPath method.

%		Predicted hop count								
		1	2	3	4	5	6	7	8	N/A
Actual hop count	1	**50.4**	41.3	7.9	0.1	0	0	0	0	0.3
	2	0.3	**78.5**	19.0	1.9	0.1	0	0	0	0.3
	3	0.3	3.7	**88.6**	5.8	0.3	0	0	0	1.3
	4	0.1	2.4	14.6	**74.9**	3.4	0.5	0	0	4.2
	5	0.4	0.2	6.0	16.6	**67.2**	3.2	0.2	0	5.8
	6	0	0.4	3.2	2.8	10.4	**55.2**	0.6	0	27.4
	7	0	0.2	1.1	2.3	0	0.9	**16.4**	0	79.1
	8	0	0	1.1	1.1	12.5	0	4.6	**8.0**	72.7

Table 2.8: Confusion matrices of the prediction performance of the Valley-free HyperPath method.

%		Predicted hop count								
		1	2	3	4	5	6	7	8	N/A
Actual hop count	1	**48.1**	42.5	9.2	0.2	0	0	0	0	0
	2	0.1	**75.1**	22.6	2.2	0	0	0	0	0
	3	0	3.2	**88.8**	7.7	0.3	0	0	0	0
	4	0	2.4	14.9	**77.7**	4.4	0.5	0	0	0
	5	0	0.3	6.4	18.4	**70.2**	4.3	0.3	0	0
	6	0	0.1	2.3	2.9	17.0	**70.9**	6.7	0	0
	7	0	0	1.3	1.5	0.3	2.3	**93.2**	1.5	0
	8	0	0	1.1	1.1	11.9	0	5.7	**79.6**	0

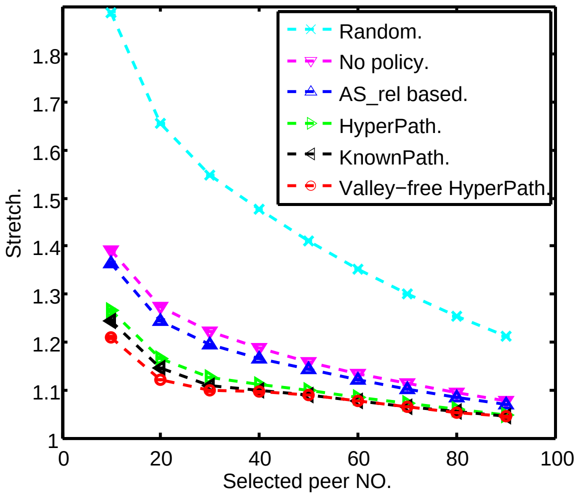


Figure 2.3: inter-domain traffic reduction on unstructured overlay networks.

2.5.4 Application: Inter-domain Traffic Reduction for BitTorrent P2P System

To demonstrate the usefulness of our AS path inference methods for the inter-domain traffic reduction, we simulate the BitTorrent, which is one of the most popular unstructured peer to peer overlay network applications.

We use the same dataset in the previous section for the underlying network condition. Specifically, we assume that a certain BitTorrent client is located in one of the 93 different IP prefixes. Then the client has to select k number of peers out of a pool of 100 peers. The 100 peers are randomly located in $26,308$ different IP prefixes. We change the selected peers number k from 10, 20 to 90. To evaluate how much inter-domain traffic is reduced by each peer selection method, we measure the performance on traffic reduction with the stretch metric, which is defined as the following:

$$\text{stretch} = \frac{\sum_{i=1}^{k} d_{x'_i}}{\sum_{i=1}^{k} d_{x_i}} \tag{2.5.1}$$

where $x'_i, i = 1, \ldots, k$ is the IDs of the selected peers based on the inferred information and $x_i, i = 1, \ldots, k$ is the IDs of the true best-performing peers in the pool. d denotes the actual AS hop count from the node to the peer. The stretch metric can reflects the ratio of the inter-ASes traffic introduced by the peer selection strategy based on estimation to the inter-ASes traffic introduced by idealized peer selection strategy. We simulate 50 times. For each time, we conduct more than 10,000 rounds of peer selection.

Figure 2.3 shows the stretch metric with confidence interval with different k for all methods. We can see that Valley-free HyperPath method outperforms all the other methods. For instance, when $k = 10$, the random peer selection method introduces 89% additional inter-domain traffic, compared with the ideal case. If we do peer selection with the help of inference methods without considering AS relationships information, no policy method introduces 39% additional traffic and HyperPath introduces 27% additional traffic. If we take AS relationships information into account, AS relationships based method, KnownPath method and Valley-free HyperPath method introduce 36%, 25% and 21% additional traffic respectively.

This experiment shows that, even though without considering AS relationships information, traffic reduction by HyperPath method is competitive with that by KnownPath method. When we further take into account AS relationships information in our algorithm design, the proposed Valley-free HyperPath method can achieve better performance than the KnownPath method. Please note that, as shown in Table 2.3, KnownPath method also requires additional information on AS topology and higher computational complexity.

2.6 Chapter Summary

In this chapter, we revisit the AS path inference problem from the complex network perspective. A brand new constraint is proposed based on the fact the AS paths respect the underlying geometric structure of the Internet. Resulting two new AS path inference algorithms, HyperPath and Valley-free HyperPath, have $O(K)$ complexity to infer certain end-to-end AS path and can run locally. Intensive evaluation on the ground truth AS paths shows that HyperPath method can not only outperform no policy method, it can be superior to AS relationships based method by being blind to the AS relationships information. The Valley-free HyperPath method outperforms both AS relationships based method and KnownPath method. Moreover, two new algorithms are immune to the fail-to-detect problem, by which the benchmark methods are always haunted. We also simulate BitTorrent P2P applications to show the potential of our methods on inter-domain Internet traffic reduction.

Algorithm 3 AS relationships based inference algorithm [9]

```
INPUT: A pair of source and destination ASes$(s,d)$, AS topology $T$ and AS relationships
    information $R$ of each link in the AS topology.
OUTPUT: An inference path $\hat{p}$ between source and destination ASes $(s,d)$
find all shortest uphill paths rooted from source node $s$ as $S = \{s \rightarrow \cdots \rightarrow p_i | i \in \{1,\ldots,K\}\}$; and find all shortest uphill paths rooted from destination node $d$ as $D = \{d \rightarrow \cdots \rightarrow q_i | i \in \{1,\ldots,L\}\}$. We denote the end points of the uphill paths rooted from $s$ as $P = \{p_i | i \in \{1,\ldots,K\}\}$ and the end points of the uphill paths rooted from $d$ as $Q = \{q_i | i \in \{1,\ldots,L\}\}$

//cost without peer to peer link
if $P \cap Q \neq \emptyset$ then
    $\text{cost}_0 = min_k(dist(s,k) + dist(d,k))$
else
    $\text{cost}_0 = -1$
end if

//cost with peer to peer link
if true $\in \{rp(p,q) | p \in P, q \in Q, p \neq q\}$ then
    $\text{cost}_1 = min_{p,q}(dist(s,p) + dist(d,q) + 1)$
else
    $\text{cost}_1 = -1$
end if
Here we assume for two ASes $x$ and $y$, function
$rp(x,y) = \begin{cases} \text{true} & \text{if } (x,y) \text{ is a p2p link;} \\ \text{false} & \text{otherwise.} \end{cases}$
if $\text{cost}_0 == -1$ and $\text{cost}_1 == -1$ then
    return null.
else if $\text{cost}_0 == -1$ then
    return uphill($s$,$p$), ($p$,$q$), reverse(uphill($d$,$q$)).
else if $\text{cost}_1 == -1$ then
    return uphill($s$,$k$), reverse(uphill($d$,$k$)).
else if $\text{cost}_0 <= \text{cost}_1$ then
    return uphill($s$,$k$), reverse(uphill($d$,$k$)).
else
    return uphill($s$,$p$), ($p$,$q$), reverse(uphill($d$,$q$)).
end if
```

Algorithm 4 KnownPath algorithm [34]

INPUT: A destination IP prefix p, baseASset, AS topology T and AS relationships information R of each link in the AS topology.

OUTPUT: Inferred paths from all ASes to the IP prefix p.

```
queue $\leftarrow \emptyset$
for $v \in$ baseASset do
    append $v$ to queue
    path($v$)[$p$] $\leftarrow$sure path of $v$
    SORT(rib_in($v$)[$p$])
end for
while queue.length $> 0$ do
    $u \leftarrow$ POP(queue, 0)
    for $w \in$ peers($u$) do
        $P_u \leftarrow$ rib_in($u$)[$p$][0]
        if $w \notin$ baseASset and $(w) + P_u$ is a valid path then
            tmppath $\leftarrow$ rib_in($w$)[$p$][0]
            rib_in($w$)[$p$] $\leftarrow$ rib_in($w$)[$p$] $\cup$ $\{(w) + P_u\}$
            SORT(rib_in($w$)[$p$])
            if tmppath == path($w$)[$p$][0] and $w \notin$ queue then
                append $w$ to queue
            end if
        end if
    end for
end while
return $\{$rib_in($v$)$|\forall v \in V\}$
```

Chapter 3

Probabilistic Model Enhanced Compressive Sensing for Environment Reconstruction in Sensor Networks

Recent studies on environment reconstruction in Wireless Sensor Networks (WSNs) suggest that compressive sensing (CS) is a promising approach to restore all sensor readings from small percentage of observations. In this chapter, we propose a new approach, namely Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing (PMEST-CS), to boost the performance of CS-based methods from two new perspectives. Firstly, we explicitly exploit the sparsity in spatio-temporal difference in environment. Secondly, we propose a pair-wise Markov Random Field (MRF) based probabilistic model to integrate spatio-temporal correlation structure into CS-based algorithm design. Experimental results utilizing the Intel lab dataset and the Uppsala dataset show that significant performance gains, in terms of reconstruction quality, can be obtained compared to the state of the art CS-based methods. For instance, PMEST-CS can achieve 30% accuracy improvement when 90% and even more data is missing for Intel lab temperature data.

Contents

3.1 Introduction **33**

3.2 Related Work **34**

3.3 Problem Formulation **35**

3.3.1 Environmental Data Reconstruction 35

3.3.2 Problem Statement 35

3.4 Two Real World WSN Datasets **36**

3.4.1 Intel Lab Dataset 36

3.4.2 Uppsala Dataset 37

3.5 Probabilistic Model for WSN **37**

3.5.1 Preprocessing: Data Quantization 38

3.5.2 Pairwise MRF for WSN 38

3.6 Model Learning in WSN MRF **40**

3.6.1 The Log-linear Representation 40

3.6.2 Spatio-Temporal Local Distribution Learning Algorithm from Complete Data 43

3.6.3 Spatio-Temporal Local Distribution Learning Algorithm from Incomplete Data 44

3.7 Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing . **45**

3.7.1 Compressive Sensing Based Approach Design 46

3.7.2 ESTI-CS Approach 47

3.7.3 Our Approach: PMEST-CS 48

3.8 Evaluation **51**

3.8.1 Incomplete Data Driven Model Training 52

3.8.2 Environment Reconstruction 53

3.9 Chapter Summary **53**

3.1 Introduction

Compared with traditional data collection methods, Wireless Sensor Networks (WSNs) are able to monitor the environment of interest in much higher frequency and resolution. WSNs [3–7] have been used to collect various kinds of data such as temperature and humidity level in forest, seismic and acoustic data of active volcano, marine pollution level in ocean etc. In reality, due to hardware damage, low battery level and/or poor condition in WSN communication data collected by WSN often contains considerable percentage of missing readings. Thus environment reconstruction [23] methods have been proposed to restore the measurements from raw (incomplete) data. The accuracy of the environment reconstruction is critical for many basic scientific research.

Different kinds of prior knowledge have been exploited in existing solutions to optimize the speed and accuracy of signal reconstruction. A recent proposal — Compressive Sensing (CS [36]) exploits sparsity for efficient reconstruction and has become a key technique in today's signal processing systems. It can be adapted in WSN since the measurement matrices also have sparse structure in their singular values, but a straightforward adaptation is not enough. Studies have observed that many natural signals have features in addition to sparsity, e.g., structure [37], clustering property in image [38], etc. A recent research in WSN (ESTI-CS [39]) exploits strong time stability and spatial correlation together with sparsity to improve the accuracy of reconstruction. However, we show that by exploiting more features we can further improve the performance of reconstruction in WSNs.

In this chapter, we propose Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing (PMEST-CS) that extends ESTI-CS by utilizing two kinds of prior knowledge: 1) with analysis on real datasets, we show that the spatio-temporal feature of WSN data *is* sparse and can be exploited further, and 2) we find that statistical inference on the probabilistic model can provide us a rough guess with a confidence level on the missing readings which can enhance the overall accuracy.

We also realized that the probabilistic model is a critical component in our solution. Therefore, we design a tree based Markov Random Field (MRF) that takes both temporal and spatial correlation of environment into consideration. Furthermore, we train the MRF from WSN data to improve model quality. One challenge raised here is that standard learning scheme cannot scale when the feeding WSN data is incomplete. Therefore, we also propose a new algorithm that can build a qualified MRF out of highly incomplete data.

Specifically, we make the following contributions:

- We propose a new compressive sensing optimization problem which exploits the sparsity in the spatio-temporal difference and leverages prior knowledge from a probabilistic model.

- To overcome the limitations of existing probabilistic models, we design an MRF model which incorporates both spatial and temporal correlation in the environment. To cope with the highly incomplete data in WSNs, we propose a new learning algorithm for MRF. Our evaluation results show that the proposed learning algorithm can generate highly effective MRF models from data even with 60% of missing readings.
- We perform intensive quantitative analysis to show that our solution can outperform the state of the art approach (ESTI-CS) by 30% in terms of accuracy.

The remainder of the chapter is organized as follows. In Section 3.2, we provide the related work on environment reconstruction in WSN. The problem is modeled in Section 3.3 which is followed by the WSN data set description in Section 3.4. We introduce the probabilistic graphical model in Section 3.5 and describe the learning scheme in Section 3.6. Section 3.7 details the overall PMEST-CS approach and Section 3.8 reports the evaluation results. We conclude this chapter in Section 3.9.

3.2 Related Work

A new idea of signal processing is of compressive sensing. For many real world signal exhibits structure and redundancy, it is possible to reconstruct original signal from partial or incomplete one. The structure or redundancy in real world signal are also called sparsity. It means there are a great fraction of zeros and a tiny fraction of nonzero values in the signal vector. In the context of matrices, sparsity means low rank of the matrix. Because the vector made up by the singular values of such a matrix is a sparse vector. CS-based methods have been utilized in different use cases, including network traffic estimation [61] and localization in mobile network [62].

ESTI-CS [39] is the state of the art CS-based method for environment reconstruction in WSNs. It leverages the benefits of both compressive sensing and environmental space-time features. PMEST-CS is similar to ESTI-CS considering that both of them explore prior knowledge or structure in WSN data to improve CS method. Different from ESTI-CS, PMEST-CS considers not only spatio-temporal feature in WSNs, but also information from additional probabilistic model.

3.3 Problem Formulation

In this section, we first introduce the problem formulation for environment reconstruction, by using the part of the notations stated in [39].

3.3.1 Environmental Data Reconstruction

Suppose for a given WSN, n sensor nodes are deployed in an environment of interest. The sensors sense and forward readings to the sink node periodically for m time instances.

Definition 3.1 (). *Environment Matrix (EM): We use X to denote the sensor readings of a WSN in an ideal case. The ideal case here means all readings can be successfully transferred to the sink node. X is with the form of a $n \times m$ matrix. Row $X_{i,:}$ records all the readings from the i^{th} sensor during the whole m time instances. Column $X_{:,t}$ records the readings from all sensors at time instance t.*

Definition 3.2 (). *Binary Indicator Matrix (BIM): We use B to denote all the possible missing readings in the WSN. B is a matrix and shares the same size with X. Specifically, each element in B is defined as the following:*

$$B_{i,t} = \begin{cases} 0 & \textit{if sensor reading } X_{i,t} \textit{ is lost} \\ 1 & \textit{otherwise} \end{cases}, \tag{3.3.1}$$

Definition 3.3 (). *Sensory Matrix (SM): We use S to denote the actual sensor readings collected at the sink node. S is a matrix and share the same size with X and B. Let's assume $X_{i,t} > 0$ and denote lost readings in S as 0. Then the actual readings S can be presented by the element-wise production of X and B as follows:*

$$S = X \circ B, \tag{3.3.2}$$

3.3.2 Problem Statement

Environment reconstruction aims to restore the EM X given available SM S.

Definition 3.4 (). *Reconstructed Matrix (RM): We use $\hat{X}$ to denote the reconstruction of the EM X.*

Problem 3.3.1. *Environment Reconstruction in Sensor Network (ERSN): Given an SM S, the ERSN problem is to find an optimal RM $\hat{X}$ that approximates the original EM X as closely as possible. i.e.,*

$$\begin{aligned} &\textit{minimize} \quad ||X - \hat{X}||_F \\ &\textit{subject to} \quad \hat{X} \circ B = S. \end{aligned} \tag{3.3.3}$$

Here $|| \bullet ||_F$ is the Frobenius norm and used to measure the error between matrix X and $\hat{X}$. In ERSN problem, $||X - \hat{X}||_F = \sqrt{\sum_{i,t}(X_{i,t} - \hat{X}_{i,t})^2}$.

In order to measure the performance of environment reconstruction in different scenarios among different methods, we define the following metric:

Definition 3.5 (). *Error Ratio(ER): is a metric for measuring the reconstruction error after space interpolation.*

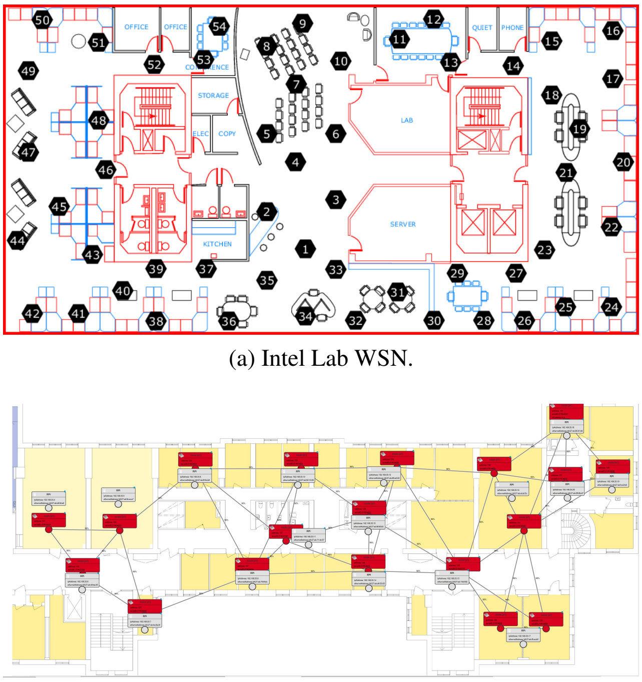
$$\varepsilon = \frac{\sqrt{\sum_{i,t:B(i,t)=0}(\hat{X}_{i,t} - X_{i,t})^2}}{\sqrt{\sum_{i,t:B(i,t)=0} X_{i,t}^2}} \tag{3.3.4}$$

3.4 Two Real World WSN Datasets

We then introduce the two real world datasets, based on which analysis and evaluation are carried out in the following sections.

3.4.1 Intel Lab Dataset

This dataset contains information about data collected from 54 sensors deployed in the Intel Berkeley Research lab between February 28^{th} and April 5^{th}, 2004. Mica2Dot sensors with weather boards collected time stamped topology information, along with humidity, temperature, light and voltage values once every 31 seconds. Data was collected using the TinyDB in-network query processing system, built on the TinyOS platform [63]. The sensors were arranged in the lab according to the diagram shown in Figure 3.1a. We will carry out analysis and experiments by using the temperature and light intensity measurement from this dataset.



(a) Intel Lab WSN.

(b) Uppsala WSN.

Figure 3.1: Two real world WSN deployments.

3.4.2 Uppsala Dataset

This dataset is collected by the WSN deployed using ProFuN TG tool [64] in Polacksbacken campus, department of information technology of Uppsala University during April 2015. The network includes 17 sensor nodes, measuring temperature, light intensity, and humidity. The sensors were arranged in the lab according to the diagram shown in Figure 3.1b. We will carry out analysis and experiments using the temperature and light readings from this dataset.

3.5 Probabilistic Model for WSN

In this section we propose a probabilistic model based on pairwise Markov Random Fields (MRF) to learn both temporal and spatial correlation of WSN sensory readings. By leveraging the learned temporal and spatial correlation knowledge, we will enhance the CS method for environment reconstruction by using the posterior distribution of missing readings derived from pairwise MRF and the given observed readings to increase the prediction confidence.

3.5.1 Preprocessing: Data Quantization

To proceed, we first briefly introduce a key data preprocessing technique — data quantization.

Quantization is a classic technique in signal processing and has been widely used for compression [65]. In general, quantized measures offer a couple of advantages as follows:

- A quantized measure is tunable and can be informative enough for describing the correlation between the data.
- A quantized measure can be encoded into a few bits, saving storage and transmission costs.
- A quantized measure is highly adjustable to match the needs of WSN application.

Let the metric to be quantized take on values in the range $[r_{min}, r_{max}]$, and values outside this interval are mapped either to r_{min} or r_{max}. The quantization is done by partitioning the interval into d bins using $d-1$ thresholds, denoted by $\tau = \{\tau_1, \ldots, \tau_{d-1}\}$. Let the value l_i represent the i bin. A table look-up is used to map a metric value to l_i according to the bin threshold:

$$Quant(x) = l_j, \text{ if } \tau_{j-1} < x \leq \tau_j, j = 1, \ldots, d, \tag{3.5.1}$$

where $\tau_0 = r_{min}$ and $\tau_d = r_{max}$. The bin index values are the followings:

$$L = \{l_1, \ldots, l_d\}, \tag{3.5.2}$$

and L is stored in a codebook. A sensor reading is represented by a bin index in L, that is encoded into few bits.

Given a SM S, usually we have $S_{i,t} \in \mathbb{R}^+$. Then we quantize the valid reading in S and obtain quantized $n \times m$ sensory matrix Q, where

$$Q_{i,t} = \begin{cases} 0 & \text{if } S_{i,t} = 0, \\ Quant(S_{i,t}) & \text{otherwise.} \end{cases} \tag{3.5.3}$$

3.5.2 Pairwise MRF for WSN

In a graph of a pairwise MRF, nodes represent the random variables and edges represent a joint probability distribution between pairs of variables. When it comes to MRF in WSN,

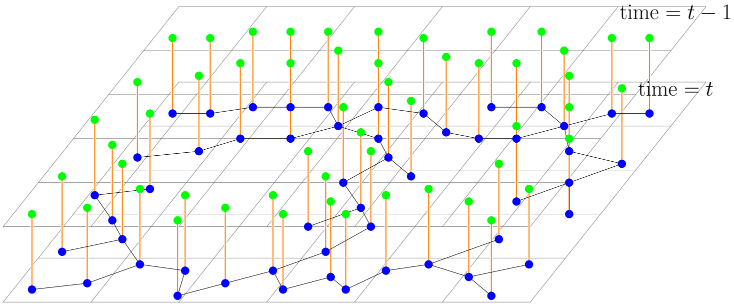


Figure 3.2: The graphical model of Intel lab WSN.

each node represents one individual sensor reading and edge corresponds to the spatio-temporal correlation between pairs of sensor readings.

To parametrize MRF, we associate with each single variable (i.e., node) or pairwise connected variables (i.e., edge) a general-purpose function, also called a factor. Specifically, we have the following three factors. In Figure 3.2, the MRF model for Intel lab WSN of 54 sensors is shown for illustration of these factors.

- **Singleton factors:** Factor $\phi^S_{i,t}(Q_{i,t})$ is used to encode how likely the i^{th} sensor at time instance t is a particular quantized value. For example, in Figure 3.2, green dots represent the singleton factor of sensor readings at the time instance $t-1$ and blue dots represent all singleton factors of sensor readings at time instance t, where $Scope[\phi^S_{i,t}(Q_{i,t})] = L$, where L is given in Equation (3.5.2);
- **Spatial pairwise factors:** Factor $\phi^{SP}_{i,j}(Q_{i,t}, Q_{j,t})$ is used to encode the joint probability distribution between the reading of the i^{th} sensor and the reading of the j^{th} sensor at time instance t. For example, in Figure 3.2, blue edges represent all spatial pairwise factor in current time instance, where $Scope[\phi^{SP}_{i,j}(Q_{i,t}, Q_{j,t})] = L \times L$;
- **Temporal pairwise factors:** Factor $\phi^{TP}_i(Q_{i,t-1}, Q_{i,t})$ is used to encode the i^{th} sensor's joint probability distribution between the reading at the time instance t and the reading at the time instance $t-1$. For example, in Figure 3.2, orange edges represent all temporal pairwise factor, where $Scope[\phi^{TP}_i(Q_{i,t-1}, Q_{i,t})] = L \times L$.

By combining these local interactions or factors, we can define a global model to approximate the spatio-temporal joint probability distribution of all sensor readings in the WSN. We denote $Q_{:,t}$ as the quantized sensor reading of all sensors at time instance t. Then the approximated joint probability distribution of the quantized sensor readings $Q_{:,t-1}$ and $Q_{:,t}$

is the following:

$$\begin{aligned}P(Q_{:,t-1},Q_{:,t}) = \frac{1}{Z}\prod_{i=1}^{n}\phi_i^S(Q_{i,t-1})\prod_{i=1}^{n}\phi_i^S(Q_{i,t})\\ \prod_{(i,j)\in \mathrm{E}}\phi_{i,j}^{SP}(Q_{i,t},Q_{j,t})\prod_{i=1}^{n}\phi_i^{TP}(Q_{i,t-1},Q_{i,t}),\end{aligned} \tag{3.5.4}$$

where E is the edge set of all spatial pairwise factors and Z is the partition function given as

$$\begin{aligned}Z = \sum_{Q'_{:,t-1},Q'_{:,t}}\prod_{i=1}^{n}\phi_i^S(Q'_{i,t-1})\prod_{i=1}^{n}\phi_i^S(Q'_{i,t})\\ \prod_{(i,j)\in \mathrm{E}}\phi_{i,j}^{SP}(Q'_{i,t},Q'_{j,t})\prod_{i=1}^{n}\phi_i^{TP}(Q'_{i,t-1},Q'_{i,t}).\end{aligned} \tag{3.5.5}$$

3.6 Model Learning in WSN MRF

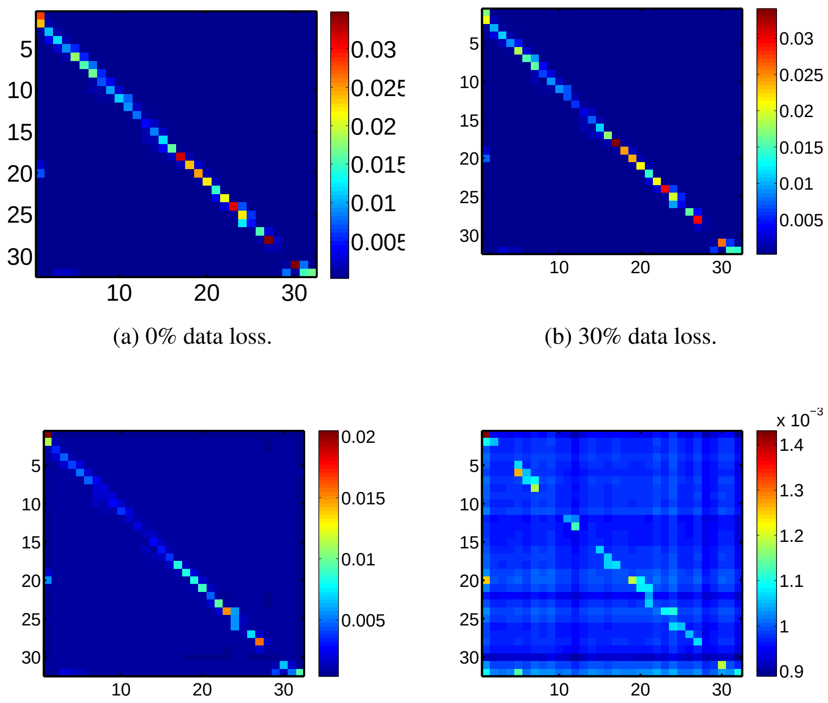
To make the joint probability in Equation (3.5.4) capture the real statistical properties of WSN, we still need to specify local factors carefully. We adapt learning scheme to extract the local factor parameters instead of using a hand crafted model. The benefits are many folds. Learning scheme is free from involving expert knowledge of environment and it always delivers optimal model parameters. In this section, we will first introduce the MRF learning algorithm based on the complete sensory data. Building on this, we further propose an advanced MRF learning algorithm for the incomplete data case.

3.6.1 The Log-linear Representation

Due to the factor product form of the distribution in (3.5.4), it is very difficult to do the learning. Thus we consider to use log linear representation to approximate the distribution in Equation (3.5.4). We first define the following notations.

3.6.1.1 Feature Functions

A feature is a function $f_i(D_i)$:Val$(D_i) \to \mathbb{R}$, where D_i is the set of variables in the scope of the i^{th} feature. Features are binary indicators features and each of them has an associated parameter θ_i. We define the following three types of indicator features for the three types of factors in our pairwise MRF above:



(a) 0% data loss.

(b) 30% data loss.

(c) 60% data loss.

(d) 90% data loss.

Figure 3.3: The spatial correlation knowledge learned from Intel lab temperature dataset with increasing loss rate. The bars on the right hand side represent the correlation factors. A larger value means a stronger correlation.

- $f^{S}_{i,u}(Q_{i,t}) = \mathbf{1}(Q_{i,t} = l_u)$ for singleton factor;
- $f^{SP}_{i,j,u,v}(Q_{i,t}, Q_{j,t}) = \mathbf{1}(Q_{i,t} = l_u$ and $Q_{j,t} = l_v)$ for spatial pairwise factor;
- $f^{TP}_{i,u,v}(Q_{i,t-1}, Q_{i,t}) = \mathbf{1}(Q_{i,t-1} = l_u$ and $Q_{i,t} = l_v)$ for temporal pairwise factor.

Here $i, j \in \{1, \ldots, n\}, t = 2, \ldots, m$ and $u, v \in \{1, \ldots, d\}$.

3.6.1.2 Parameters Sharing

To reduce the total number of parameters we need to learn, we share parameters across multiple features. The parameters that we use in our model are the following:

- θ^S_u, shared by $f^S_{i,u}(Q_{i,t})$ where $i = 1,\dots,n$, i.e., all n number of singleton factors share the same set of parameters;
- $\theta^{SP}_{u,v}$, shared by $f^{SP}_{i,j,u,v}(Q_{i,t}, Q_{j,t})$ where $(i, j) \in \mathrm{E}$, i.e., all $|\mathrm{E}|$ number of spatial pairwise factors share the same set of parameters;
- $\theta^{TP}_{u,v}$, shared by $f^{TP}_{i,u,v}(Q_{i,t-1}, Q_{i,t})$ where $i = 1,\dots,n$, i.e., all n number of temporal factors share the same set of parameters.

We denote $\Theta = \{\theta^S_1,\dots,\theta^S_d,\theta^{SP}_{1,1},\dots,\theta^{SP}_{d,d},\theta^{TP}_{1,1},\dots,\theta^{TP}_{d,d}\}$. There are $\hat{d} = d+2\times d^2$ parameters in total, where d is quantized bin number.

Given the above features functions and parameters, the distribution in Equation (3.5.4) can be approximated as the following:

$$P(Q_{:,t-1}, Q_{:,t}|\Theta) = \frac{1}{Z(\Theta)} exp(\mathscr{S}(Q_{:,t-1}, Q_{:,t}, \Theta)), \tag{3.6.1}$$

where $Z(\Theta)$ is the partition function as

$$Z(\Theta) = \sum_{Q'_{:,t-1}, Q'_{:,t}} exp(\mathscr{S}(Q'_{:,t-1}, Q'_{:,t}, \Theta)), \tag{3.6.2}$$

and $\mathscr{S}(Q_{:,t-1}, Q_{:,t}, \Theta)$ is given as

$$\begin{aligned}\mathscr{S}(Q_{:,t-1}, Q_{:,t}, \Theta) = &\sum_{i=1}^{n}\sum_{u=1}^{d} \theta^S_u f^S_{i,u}(Q_{i,t-1})\\ &+\sum_{i=1}^{n}\sum_{u=1}^{d} \theta^S_u f^S_{i,u}(Q_{i,t})\\ &+\sum_{i,j\in \mathrm{E}}\sum_{u=1}^{d}\sum_{v=1}^{d} \theta^{SP}_{u,v} f^{SP}_{i,j,u,v}(Q_{i,t}, Q_{j,t})\\ &+\sum_{i=1}^{n}\sum_{u=1}^{d}\sum_{v=1}^{d} \theta^{TP}_{u,v} f^{TP}_{i,u,v}(Q_{i,t-1}, Q_{i,t}).\end{aligned} \tag{3.6.3}$$

Intuitively, here we assign different weights (i.e.,parameters) for different features to capture the correlation dependencies cross different factors including singleton/spatial pairwise/temporal pairwise factors. Accordingly, we can use the aggregate weights to measure the likelihood of the future sensory readings. The key issue is how to derive or learn the optimal weights to well characterize the behavior or pattern of WSN sensory readings, and this will be addressed in the coming section.

3.6.2 Spatio-Temporal Local Distribution Learning Algorithm from Complete Data

We now consider to how to learn the optimal weights to train the pair-wise MRF model for WSN. Based on Bayesian rule, we have the following conditional distribution of the learning parameters given the observed sensory readings:

$$P(\Theta|Q_{:,t-1},Q_{:,t}) = \frac{P(Q_{:,t-1},Q_{:,t}|\Theta)P(\Theta)}{P(Q_{:,t-1},Q_{:,t})}, \tag{3.6.4}$$

where $P(Q_{:,t-1},Q_{:,t})$ is a normalization constant and without prior information $P(\Theta)$ is assumed to be uniformly distributed. Then to find the optimal learning parameters, we want to maximize $P(\Theta|Q_{:,t-1},Q_{:,t})$. In this case, we only need to maximize the likelihood function $P(Q_{:,t-1},Q_{:,t}|\Theta)$, or equivalently minimize the following negative log-likelihood function (due to (3.6.1)):

$$\mathscr{N}(Q_{:,t-1},Q_{:,t}|\Theta) = log(Z(\Theta)) - \mathscr{S}(Q_{:,t-1},Q_{:,t},\Theta). \tag{3.6.5}$$

To prevent over-fitting, we also introduce a L_2-regularization penalty on the parameter values and, as a result, we have the following negative log-likelihood function:

$$\begin{aligned}\mathscr{N}(Q_{:,t-1},Q_{:,t}|\Theta) &= log(Z(\Theta)) \\ &\quad - \mathscr{S}(Q_{:,t-1},Q_{:,t},\Theta) + \frac{\lambda}{2}\sum_{i=1}^{u}\theta_i^2.\end{aligned} \tag{3.6.6}$$

It can be proved that the resulting objective function is convex and the global minimum and the corresponding variable can be found by using standard optimization methods, such as gradient descent. The partial derivatives for this function have the following elegant forms:

$$\frac{\partial}{\partial\theta_i}\mathscr{N}(Q_{:,t-1},Q_{:,t}|\Theta) = E_\Theta[f_i] - E_D[f_i] + \lambda\theta_i, \tag{3.6.7}$$

where $E_\theta[f_i]$ is the expectation of feature values with respect to the model parameters, and $E_D[f_i]$ is the expectation of the feature values with respect to the sensor readings from $(Q_{:,t-1},Q_{:,t})$. Using the definition of expectation, we have:

$$E_\Theta[f_i] = \sum_{Q'_{:,t-1},Q'_{:,t}} P(Q'_{:,t-1},Q'_{:,t}|\Theta)f_i(Q'_{:,t-1},Q'_{:,t}), \tag{3.6.8}$$

$$E_D[f_i] = f_i(Q_{:,t-1},Q_{:,t}). \tag{3.6.9}$$

Algorithm 5 Learning with complete data

```
INPUT: Sensor nodes number $n$, the number of times for measurement $m$, $n \times m$ raw sensor readings matrix $S$, number of bits for quantization $b$, minimum spanning tree $T$ regularization parameter $\lambda$, learning rate $\alpha$, maximum iteration number $IterMax$
OUTPUT: Learned parameter set $\Theta$.
Initialize $\Theta = \mathbf{0}$.
$Quant(S) = Q$.
for $\theta_i \in \Theta$ do
    Compute $\bar{E}_D[f_i]$ according to (3.6.10).
end for
repeat
    for $\theta_i \in \Theta$ do
        Compute $E_\Theta[f_i]$ according to (3.6.8).
        Compute regularization value $r = \lambda * \theta_i$
        Compute $grad_{\theta_i} = E_\Theta[f_i] - \bar{E}_D[f_i] + r$.
        $\theta_i = \theta_i - \alpha * grad_{\theta_i}$
    end for
until $\Theta$ is converged or the iteration number exceeds $IterMax$.
```

To speed up the learning procedure, we compute $E_D[f_i]$ of every $(Q_{:,t-1}, Q_{:,t}), t = 2, \ldots, m$ and get the average feature counts $\bar{E}_D[f_i]$ as follows:

$$\bar{E}_D[f_i] = \frac{1}{m-1}\left(\sum_{t=2}^{m} f_i(Q_{:,t-1}, Q_{:,t})\right) \tag{3.6.10}$$

To compute $E_\Theta[f_i]$, we have to do inference with the MRF with current parameters Θ. To this end, we use the efficient Belief Propagation (BP) algorithm [66]. The detailed algorithm for the spatio-temporal correlation learning with complete data algorithm is given in Algorithm 5.

3.6.3 Spatio-Temporal Local Distribution Learning Algorithm from Incomplete Data

When WSN data includes missing readings, Algorithm 5 simply does not work, due to that feature functions become undefined when their input variables include no value.

To cope with this challenge, a simple approach is to “fill in” the missing sensor readings arbitrarily first, then call Algorithm 5 as usual. The problem with such an approach is that the filled in default values introduces a bias that will be reflected in the spatio-temporal distribution we learn.

A better approach is to first run probabilistic inference given a choice of parameters to get the distribution of variables whose readings are missing and then use Algorithm 5 to update the parameters.

Based on this simple idea, we propose a new algorithm that is capable for extracting spatio-temporal distribution from highly incomplete WSN data. The new algorithm is an instantiations of Expectation-Maximization algorithm. The detailed algorithm is given in Algorithm 6.

Specifically, in the E-step, the algorithm use the current model Θ to determine the expected feature count with respect to the incomplete data, which has the similar rule of $E_D[f_i]$. For each pair of continues sensor readings $(Q_{:,t-1}, Q_{:,t}), t = 2, \ldots, m$, we first build a model in which observed sensor readings are regarded as evidence. Then we compute the expected feature count $E_{P(h(i)|o(i),\Theta)}[f_i]$ and treat it as $E_D[f_i]$, where $P(h(i)|o(i),\Theta)$ is the probability distribution of missing readings $h(i)$ given observed readings $o(i)$ for sensor readings $(Q_{:,t-1}, Q_{:,t})$. Then, just like Equation (3.6.10), we can compute the average feature count as follows:

$$\bar{M}_\Theta[f_i] = \frac{1}{m-1}\left[\sum_{i=2}^{m} E_{P(h(i)|o(i),\Theta)}[f_i]\right]. \tag{3.6.11}$$

With the average expected feature counts $\bar{M}_\Theta[f_i]$, we can perform an M-step by doing maximum likelihood parameter estimation. As an instantiations of EM algorithm, our algorithm converges to an (at least local) optimum in the parameter space.

To get a better understanding of the performance on Algorithm 6, we extract the spatial correlation knowledge (i.e. $\{\theta_{1,1}^{SP}, \ldots, \theta_{d,d}^{SP}\} \in \Theta$) from data with different loss rates. Figure 3.3 shows that Algorithm 6 can learn the knowledge well from data even with 60% of missing readings.

3.7 Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing

In this section, we will introduce the Probabilistic Model Enhanced Spatio-Temporal Compressive Sensing (PMEST-CS) approach for environment reconstruction, by leveraging the spatial and temporal distributional knowledge learned by the pair-wise MRF. To make the discussion easier, we first briefly review the basic CS and ESTI-CS approaches in literature.

Algorithm 6 Learning with incomplete data

```
INPUT: Sensor nodes number $n$, the number of times for measurement $m$, $n \times m$ raw sensor readings matrix $S$, number of bits for quantization $b$, minimum spanning tree $T$, regularization parameter $\lambda$, learning rate $\alpha$, maximum iteration number $IterMax_1$ for gradient descent, maximum iteration number $IterMax_2$ for E-M steps,
OUTPUT: Learned parameter set $\Theta$.
Initialize $\Theta = \mathbf{0}$.
$Quant(S) = Q$.
repeat
    Expectation-Step
    for $\theta_i \in \Theta$ do
        Compute $\bar{M}_\theta[f_i]$ according to Equation (3.6.11).
    end for
    Maximization-Step
    repeat
        for $\theta_i \in \Theta$ do
            Compute $E_\Theta[f_i]$ according to Equation (3.6.8).
            Compute regularization value $r = \lambda * \theta_i$
            Compute $grad_{\theta_i} = E_\Theta[f_i] - \bar{M}_\theta[f_i] + r$.
            $\theta_i = \theta_i - \alpha * grad_{\theta_i}$
        end for
    until $\Theta$ is converged or the iteration number exceeds $IterMax_1$.
until $\Theta$ is converged or the iteration number exceeds $IterMax_2$.
```

3.7.1 Compressive Sensing Based Approach Design

The standard compressive sensing estimation on EM X based SM S can be obtained by solving following optimization problem:

$$\begin{aligned} \text{minimize} \quad & rank(\hat{X}) \\ \text{subject to} \quad & \hat{X} \circ B = S. \end{aligned} \tag{3.7.1}$$

Suppose $\hat{X}$ has low rank $k << min(n,m)$, we can factorize $\hat{X}$ into the matrix product of a tall thin $n \times k$ matrix L and a short thick $k \times m$ matrix R as follows:

$$\hat{X} = LR. \tag{3.7.2}$$

If we substitute Equation (3.7.2) into optimization problem (3.7.1), we have the following new optimization problem:

$$\begin{aligned} \text{minimize} \quad & rank(LR) \\ \text{subject to} \quad & B \circ (LR) = S. \end{aligned} \tag{3.7.3}$$

Unfortunately, the objective function $rank(LR)$ in the above optimization problem is not convex and we can not solve it efficiently. But if the BIM B satisfies the, so called, restricted isometry property [67] and the rank of $\hat{X}$ is less then the rank of LR then Equation (3.7.3) is equivalent to the following convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & ||L||_F^2 + ||R||_F^2 \\ \text{subject to} \quad & B \circ (LR) = S. \end{aligned} \tag{3.7.4}$$

To avoid over-fitting, the following problem with regularization is solved instead:

$$\text{minimize} \quad ||B \circ (LR) - S||_F^2 + \lambda * (||L||_F^2 + ||R||_F^2). \tag{3.7.5}$$

The parameter λ can be used to trade off between precise matching to the observation and achieving low rank matrix.

3.7.2 ESTI-CS Approach

ESTI-CS method extends the standard CS by taking into account the idea of “spatial and temporal stability”, i.e., the differences between the readings from two neighboring sensors at the same time instance or from the same sensor between two time consecutive instances should be very small [39].

To compute the spatial and temporal stability, two matrices H and T are introduced. For a WSN, a graph G with sensors as nodes and Euclidean distance between sensors as the edge weights is constructed. In the graph G, one of minimum spanning trees is selected and its adjacent matrix is denoted as A. Then H is the $n \times n$ random walk normalized Laplacian matrix of A and defined as follows:

$$H = I - D^{-1}A, \tag{3.7.6}$$

where I is identical matrix and$D$ is the degree matrix of A. T is a $m \times m$ Toeplitz matrix with the central diagonal given by ones and the first upper diagonal given by negative ones. For a given EM X, matrix operation HX captures spatial difference between nearby sensors and matrix operation XT captures the temporal difference of consequent readings of same sensors.

Then the resulting ESTI-CS optimization problem is the following:

$$\begin{aligned} \text{minimize} \quad & ||B \circ (LR) - S||_F^2 + \lambda * (||L||_F^2 + ||R||_F^2) \\ & + ||H(LR)||_F^2 + ||(LR)T||_F^2. \end{aligned} \tag{3.7.7}$$

Table 3.1: Selected datasets for spatio-temporal difference analysis.

Data Name	Matrix Size	Time Interval
Intel lab temperature	54 nodes × 1000 intervals	31 seconds
Intel lab light	54 nodes × 1000 intervals	31 seconds
Uppsala temperature	17 nodes × 1000 intervals	15 seconds
Uppsala light	13 nodes × 1000 intervals	15 seconds

By incorporating the term $||H(LR)||_F^2 + ||(LR)T||_F^2$ in the objective function, ESTI-CS is trying to minimize the spatio-temporal difference in RM $\hat{X}$ while it is reaching a low rank approximation.

3.7.3 Our Approach: PMEST-CS

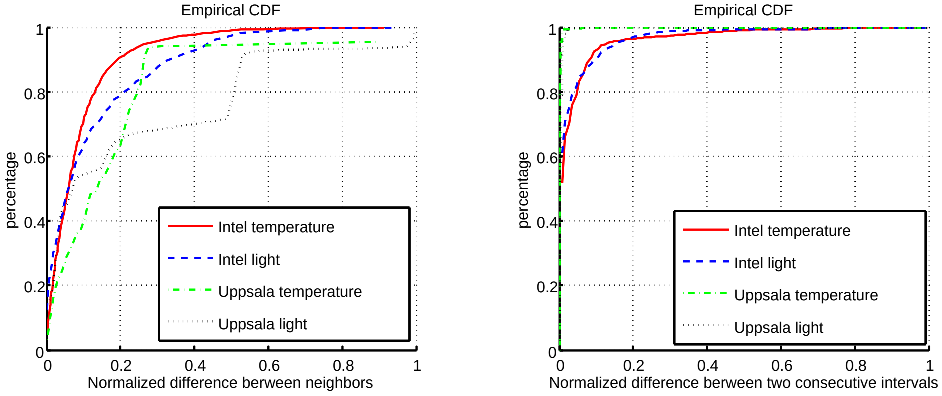
We now introduce our proposed probabilistic model enhanced compressive sensing method. The main enhancement stems from the following two aspects.

3.7.3.1 Exploiting Sparsity in Spatio-temporal Difference

Many environment properties, such as temperature, light intensity and humidity, usually remain stable in a short period of time. As a result, two measurements of a certain environment property at consequent two time instances are often close to each other. In the same way, two places close to each other also share similar environment properties. Kong et al. had already shown this phenomenon in their paper [39] for the ESTI-CS method. However, they didn't fully exploit the spatio-temporal difference features in their approach.

In order to better understand this problem, we conduct similar analysis on the selected datasets as shown in Table 3.1. For each EM X, we again use matrix operation HX and XT to capture the spatio-temporal difference of data. We normalize the result and summarize their CDF in Figure 3.4. In Figure 3.4a, The X-axis presents the normalized difference between values of one sensor and its direct neighbors on the minimum spanning tree. In Figure 3.4b, The X-axis presents the normalized difference between values of one sensor at two consequent time instances. Y-axis presents the cumulative probability. We observe that, for any data set, at least 60% of spatial difference and 95% of temporal difference are less than 0.2. From the analysis result, it is safe to argue that temporal differences in WSN are sparse and spatial differences are close to sparse.

However, the penalty function on individual spatio-temporal difference introduced by the square of Frobenius norm in ESTI-CS is actually L_2 norm. It is well known that L_2 norm



(a) Spatical difference.

(b) Temporal difference.

Figure 3.4: Spatio-temporal difference analysis in selected real world datasets.

is not the best penalty function for usage when the signal is sparse or close to sparse. To truly fully exploit the sparsity in spatio-temporal difference in WSN, the Smoothly Clipped Absolute Deviation (SCAD) [68] penalty function or similar ones should be considered. Yet it would leave the objective function non convex and make the problem hard to solve. In this chapter, instead, we introduce L_1 norm as the replacement of L_2 norm. L_1 norm is the tightest convex surrogate of L_0 norm which leads to sparse solution [69]. .

3.7.3.2 Enriching Observation Set by Probabilistic Model

Given available sensory data, statistical inference based on our probabilistic model (i.e., inference based on the learned spatial and temporal distributional knowledge) is able to generate posterior probability distribution function of the missing readings. We denote the posterior probability distribution function of i^{th} sensor at t^{th} time instance as $p_{X_{i,t}}$ and compute its expected sensor reading as $\mathbf{E}(p_{X_{i,t}}) = \sum_{x_{i,t}} x_{i,t} p_{X_{i,t}}(x_{i,t})$.

Intuitively, if we regard the expected sensor reading as the predicted value $\mathbf{E}(p_{X_{i,t}})$, then we can then use the entropy $\mathbf{H}(p_{X_{i,t}})$ of the posterior probability distribution function $p_{X_{i,t}}$ to capture the uncertainty of the predicted reading by i^{th} sensor at t^{th} time instance. Moreover, in Figure 3.5, we show the scatter plot between prediction error (i.e., $|X_{i,t} - \mathbf{E}(p_{X_{i,t}})|$) and the entropy $\mathbf{H}(p_{X_{i,t}})$. We can see that prediction with a concentrated probability distribution function (i.e., lower entropy) often corresponds to accurate prediction to $X_{i,t}$. The extreme case would be those observed readings whose $\mathbf{H}(p_{X_{i,t}}) = 0$ and $\mathbf{E}(p_{X_{i,t}}) = X_{i,t}$. On the other hand, predictions with a scatted probability distribution function or higher entropy are usually inaccurate.

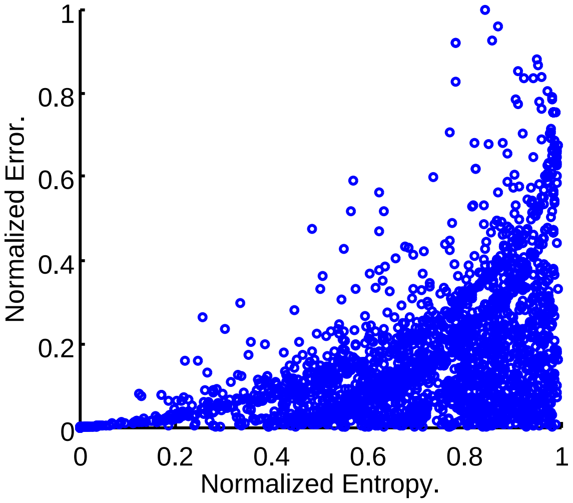


Figure 3.5: Normalized entropy obtained by probabilistic model versus normalized error of expected value when 50% data is missing.

After observing the connection between the prediction error and the entropy from the posterior probability distribution function, the next question would be how to leverage this connection to boost compressive sensing approach? The following compressive sensing problem can make it happen:

$$\text{minimize} \quad ||W \circ (LR - E)||_F^2 + \lambda * (||L||_F^2 + ||R||_F^2), \tag{3.7.8}$$

where W and E are matrices with the same size of the EM X. Specifically, for the matrix E we set $E_{i,t} = \mathbf{E}(p_{X_{i,t}})$, and for the matrix W we have

$$W_{i,t} = \left(1 - \frac{\mathbf{H}(p_{X_{i,t}})}{\mathbf{H}(p_U)}\right)^k, \tag{3.7.9}$$

where p_U is the uniform distribution such that $\mathbf{H}(p_U)$ represents the maximum entropy $\mathbf{H}(p_{X_{i,t}})$. Intuitively, for a pair of sensor readings $X_{1,t}$ and $X_{2,t}$, let's assume $\mathbf{H}(p_{X_{1,t}}) > \mathbf{H}(p_{X_{2,t}})$, i.e., $\mathbf{E}(p_{X_{1,t}})$ tends to have a higher chance of being erroneous (i.e., having higher entropy). In this case, according the Equation (3.7.9), the matrix weights satisfy that $W_{1,t} < W_{2,t}$. Since $|\hat{X}_{i,t} - \mathbf{E}(p_{X_{i,t}})|$ measures the deviation of the restored reading $\hat{X}_{i,t}$ from the expected reading $\mathbf{E}(p_{X_{i,t}})$, in this case Problem (3.7.8) will impose more weight on the deviation term $|\hat{X}_{2,t} - \mathbf{E}(p_{X_{2,t}})|$ than $|\hat{X}_{1,t} - \mathbf{E}(p_{X_{1,t}})|$, in order to reduce the deviation from the expected readings for those sensor readings (e.g., $X_{2,t}$) of higher confident level (i.e., lower entropy).

Moreover, the parameter k is used to tune the degree to which the posterior probability distribution function based entropy measure should be integrated into compressive sensing. A larger k means less importance. When the parameter $k \to +\infty$, $W_{i,t} = 1$ when $X_{i,t}$ is observed and $W_{i,t} = 0$ otherwise. In this case, Equation (3.7.8) degenerates to a standard compressive sensing method without using the entropy measure. In this chapter, through numerical evaluation we set $k = 3$ to achieve best performance.

Together with exploiting sparsity discussed in Section 3.7.3.1, we finally have our PMEST-CS problem as follows:

$$\begin{aligned} \text{minimize} \quad & ||W \circ (LR - E)||_F^2 + \lambda * (||L||_F^2 + ||R||_F^2) \\ & + ||H(LR)||_1 + ||(LR)T||_1, \end{aligned} \tag{3.7.10}$$

where $|| \bullet ||_1$ is the matrix element wise norm with $p = 1$ (i.e., L_1 norm). For example, $||X||_1 = \sum_{i,t} |X_{i,t}|$.

3.7.3.3 Solving PMEST-CS Problem

Similar to the standard CS approach, we derive L and R using an alternative least square procedure. Specifically, L and R are initialized as random matrix. Then we first treat one of them as given constant and get another one by solving a convex (quadratic) programming problem. Afterwards we swap their roles and solve the problem again. We continue this procedure until L and R converge or the iteration time reach the threshold value. To solve each sub problem, we used CVX, a package for specifying and solving convex programs [70, 71]. From the evaluation experience in Section 3.8, L and R converge after 5 iterations.

3.8 Evaluation

We next evaluate the performance of the proposed PMEST-CS method through numerical studies. The evaluation contains two parts. First part evaluates the incomplete data learning algorithm on generating effective probabilistic model. Second part evaluates the PMEST-CS approach in environment reconstruction.

Table 3.2: Selected datasets for probabilistic model learning form incomplete data.

Data Name	Matrix Size	Time Interval
Intel lab temperature	54 nodes × 120 intervals	31 seconds
Intel lab light	54 nodes × 120 intervals	31 seconds
Uppsala temperature	17 nodes × 400 intervals	15 seconds

Table 3.3: Selected datasets for environment reconstruction.

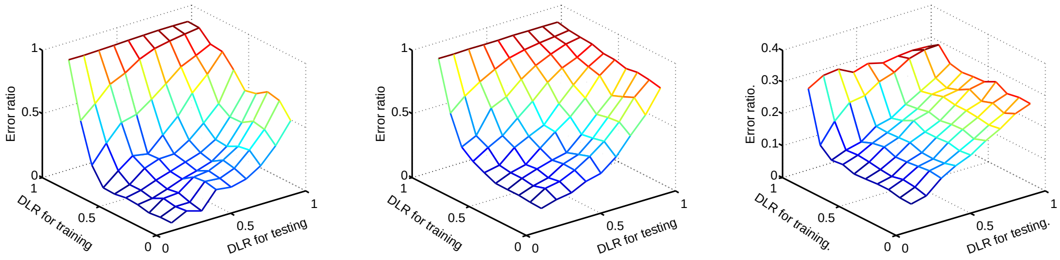
Data Name	Matrix Size	Time Interval
Intel lab temperature	54 nodes × 100 intervals	31 seconds
Intel lab light	54 nodes × 100 intervals	31 seconds
Uppsala temperature	17 nodes × 100 intervals	15 seconds
Uppsala light	13 nodes × 100 intervals	15 seconds

3.8.1 Incomplete Data Driven Model Training

To understand the relationship between the quality of learned probabilistic model and the severity of the missing data, we build nine different models based on incomplete data with loss rate ranging from 10% to 90%. Then we use these models for environment reconstruction, where the percentages of missing readings also range from 10% to 90%. We use the dataset in Table 3.2. For Intel lab data, we use first 100 time instance data for training and use the remaining ones for testing. For Intel Uppsala data, we use first 300 time instance data for training and use the remaining ones for testing. Other settings on the Algorithm 6 are given as the following:

- $IterMax = 200$, $IterMax_1 = 20$, $IterMax_2 = 10$;
- $\lambda = 0.03$;
- $\alpha = \frac{0.05}{1+0.05*iter}$.

The result is given in Figure 3.6. We observe that the learned models from incomplete data can achieve comparable performance with those model learned from complete data until 50% of data loss for Intel temperature, 30% of data loss for Intel light and 70% of data loss for Uppsala temperature. The result is showing that even with a considerable percentage of data loss, the incomplete data learning algorithm can generate high quality models which is comparable with the models trained on complete data.



(a) Intel lab temperature. (b) Intel lab light intensity. (c) Uppsala temperature.

Figure 3.6: Loss data prediction application based on learned model from incomplete data with random loss rate from 10% to 90%.

3.8.2 Environment Reconstruction

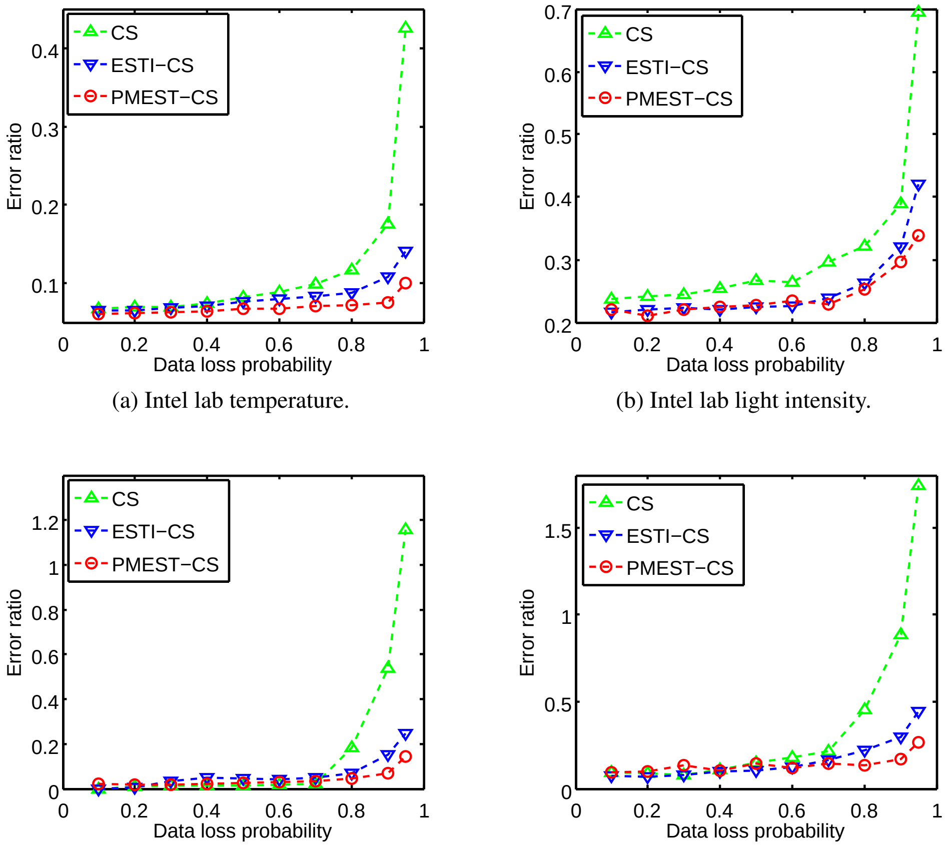
In this section, we run simulation on environment reconstruction with different compressive sensing algorithms. Detailed information on the environment matrix has to be restored are given in Table 3.3 and simulation results are given in Figure 3.7. The percentages of missing readings again range from 10% to 90%.

Figure 3.7 shows that PMEST-CS outperforms the ESTI-CS and standard CS approaches in every setting of simulation. Especially when the data loss rate is extremely high, such as 95%, the PMEST-CS degenerate slower than other two methods. Specifically, for Intel lab temperature dataset, PMEST-CS can achieve 30% accuracy improvement compared to ESTI-CS approach when data loss rate is equal or higher than 90%.

To visualize the restoration effect obtained by different CS methods, we reconstruction the Uppsala light intensity EM from data with 60% of random loss. Figure 3.8 demonstrates that PMEST-CS is able to restore the missing data much better than CS and ESTI-CS method can.

3.9 Chapter Summary

In this chapter, we proposed a new CS-based approach for environment reconstruction problem in WSN. By exploiting the sparsity in spatio-temporal difference and introducing additional informative "observations" with probabilistic model, the proposed PMEST-CS method can achieve superior performance than the state of the art methods. We also proposed a new learning algorithm which can efficiently learn probabilistic model from highly



(a) Intel lab temperature.

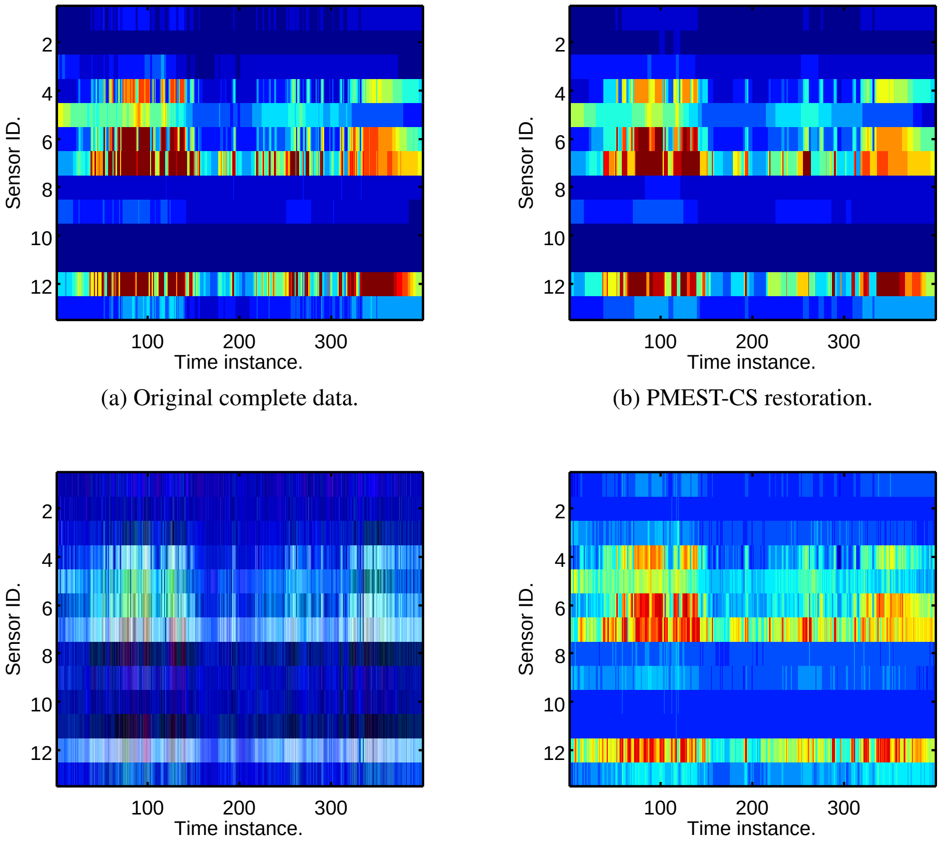
(b) Intel lab light intensity.

(c) Uppsala temperature.

(d) Uppsala light intensity.

Figure 3.7: Performance of different CS-based algorithms on environment reconstruction.

incomplete data. Experimental results corroborate PMEST-CS method is very efficient and can achieve significant reconstruction quality gain over the state of the art CS-based methods.



(a) Original complete data.

(b) PMEST-CS restoration.

(c) Standard CS restoration.

(d) ESTI-CS restoration.

Figure 3.8: Uppsala light intensity data and its restoration with 60% data loss rate by different CS-based environment reconstruction methods.

Chapter 4

Rating Network Paths for Locality-Aware Overlay Construction and Routing

This chapter investigates the rating of network paths, i.e. acquiring quantized measures of path properties such as round-trip time and available bandwidth. Comparing to fine-grained measurements, coarse-grained ratings are appealing in that they are not only informative but also cheap to obtain.

Motivated by this insight, we firstly address the scalable acquisition of path ratings by statistical inference. By observing similarities to recommender systems, we examine the applicability of solutions to recommender system and show that our inference problem can be solved by a class of matrix factorization techniques.

Then, we investigate the usability of rating-based network measurement and inference in applications. A case study is performed on whether locality awareness can be achieved for overlay networks of Pastry and BitTorrent using inferred ratings. We show that such coarse-grained knowledge can improve the performance of peer selection and that finer granularities do not always lead to larger improvements.

Contents

4.1 Introduction **59**

4.2 Related Work **61**

4.2.1 Inference of Network Path Properties 61

4.2.2 Locality-Aware Overlay Networks 61

4.3 Properties and Rating of Network Paths **62**

4.3.1 Properties of Network Paths 62

4.3.2 Rating of Network Paths 63

4.4 Network Inference of Ratings **64**

4.4.1 Problem Statement 64

4.4.2 Connections to Recommender Systems 65

4.4.3 Matrix Factorization 67

4.4.4 MF for Network Inference 70

4.4.5 Comparison of Different MF Models 71

4.5 Case Study: Locality-Aware Overlay Construction and Routing **76**

4.5.1 Pastry 77

4.5.2 BitTorrent 78

4.5.3 Remarks 80

4.6 Chapter Summary **80**

4.1 Introduction

Network measurement is a fundamental problem in the heart of the networking research. Over the years, various tools have been developed to acquire path properties such as round-trip time (RTT), available bandwidth (ABW) and packet loss rate, etc [24]. As in most scientific disciplines, the common sense in the field is that a measurement should be made as fine-grained and accurate as possible. This is considered necessary to enable quantitative analysis of network performance.

However, recent advances in emerging Internet services have created numerous situations where coarse-grained measurements can be leveraged and may even be preferred. A typical example is Peer-to-Peer (P2P) Overlay Networks [72] where a common design is to use *Intelligent Peer Selection* to improve traffic locality [41], i.e. encourage more communications between "nearby" nodes, with connections of small RTT or high ABW. The objective is thus to find "good-enough" paths for overlay construction and routing, which can be well served by using coarse-grained network measurement.

Motivated by this insight, this chapter investigates the rating of network paths, i.e. acquiring quantized path properties represented by ordinal numbers of $1,2,3,\ldots$, with larger value indicating better performance. Although coarse-grained, ordinal ratings are appealing for the following reasons:

- Ratings carry sufficient information that already fulfills the requirements of many applications.
- Ratings are rough measures that are cheaper to obtain than exact property values.
- Ratings can be encoded in a few bits, saving storage and transmission costs.

A practical issue of rating-based network measurement is the efficient acquisition on large networks. While cheap for a single path, it is still infeasible to rate all paths in a network by active probing due to the quadratic complexity. The scalability issue has been successfully addressed by statistical inference that measures a few paths and predicts the properties of the other paths where no direct measurements are made [25–33]. Inspired by these studies, a particular focus of this chapter is **network inference of ratings**: how ratings of network paths can be accurately predicted. An interesting observation is that the inference problem resembles the problem of *recommender systems* which studies the prediction of preferences of users to items [40]. If we consider a path property as a "friendship" measure between end nodes, then intelligent peer selection can be viewed as a "friend" recommendation task. This seemingly trivial connection has the great benefit to leverage the rapid progresses in machine learning and investigate the applicability of various solutions to recommender systems for network inference. Our studies show that a class of matrix factorization techniques are suitable for network inference and achieved good results that are known to be acceptable for recommendation tasks.

Another practical issue on rating-based network measurement is **the usability in applications**. Two questions need to be answered, the first of which is whether the inference of ratings is accurate enough to be exploited by applications and the second of which is how to determine a proper granularity. While a coarser granularity means rougher and thus cheaper measurement, it also means more information losses which may hurt the performance of applications. Answers to these questions are critical in the design of system architecture, particularly for P2P applications where the knowledge of locality plays an important role [35, 41, 42].

Thus, we answer these two questions by investigating quantitatively the impacts of both the inaccuracy of the inference and the granularity. For the case study, we consider locality-aware overlay construction and routing where locality refers to the proximity between network nodes according to some path property such as RTT or ABW. More specifically, we performed the study on Pastry [42] and BitTorrent [35], which are typical structured and unstructured overlay networks and are known to enjoy the property of locality awareness, and evaluated the performance of overlay construction and routing, with the knowledge of locality obtained via network inference of ratings. Our studies show that while the knowledge of inferred ratings can improve the performance of peer selection, finer granularities do not always lead to larger improvements. For example, our simulations on various datasets show that the performance of peer selection improves very little when the rating level reaches 2^4.

Thus, this chapter makes the following contributions:

- We investigate the rating-based network measurement that acquires quantized path properties represented by ordinal numbers. Such representation not only is informative but also reduces measurement, storage and transmission costs.
- We investigate the scalable acquisition of ratings by network inference. We highlight similarities between network inference and recommender systems and examine the applicability of solutions from this latter domain to network inference. In particular, we show that our inference problem can be solved by a class of matrix factorization techniques.
- We perform a case study on locality-aware overlay construction and routing to demonstrate the usability of rating-based network measurement and inference in P2P applications.

The rest of the chapter is organized as follows. Section 4.2 introduces related work. Section 4.3 describes the properties and the rating of network paths. Section 4.4 introduces network inference by a class of matrix factorization techniques. Section 4.5 introduces the case study on locality-aware overlay construction and routing. Section 4.6 gives conclusions and future work.

4.2 Related Work

4.2.1 Inference of Network Path Properties

Network inference has been studied for over a decade, and numerous approaches have been developed. For example, GNP [25] and Vivaldi [26] solved the inference of RTT by Euclidean embedding, while tomography-based approaches such as TOM [27], Network Kriging [28] and NetQuest [29] used linear algebraic methods and the routing tables to infer additive properties such as RTT and packet loss rate. Although interesting in different aspects, these approaches are limited only to certain path properties and/or rely on the routing information of the network which is expensive to obtain for applications.

To overcome these constraints, a recent advance was made in [32, 33] that formulated network inference as a matrix completion problem. A class of algorithms called DMFSGD was proposed and showed improved accuracy and flexibility in dealing with various properties including the non-additive ABW. Matrix completion is suitable for network inference because it exploits the correlations between measurements of different paths [27,28,33,73] and has also been used for other problems such as network traffic estimation [74–76].

All above-mentioned work focused on the prediction of exact measurement values, except [32] which classified path properties into binary classes of either “good” or “bad”. This chapter goes further and is the first to investigate rating-based network measurement and inference. Ratings are measures in between property values and binary classes: on the one hand, they are more informative than binary classes; on the other hand, they are advantageous over property values due to the coarse-grained measurement.

4.2.2 Locality-Aware Overlay Networks

There are two classes of overlay networks: structured and unstructured [72]. The former places content deterministically at specified locations according to the Distributed Hash Table (DHT) which makes subsequent queries efficient. The latter organizes peers more randomly and uses flooding or random walks to query content. Examples of structured systems are Kademlia [77], Chord [78] and Pastry [42], and those of unstructured are Freenet [79], Gnutella [80] and BitTorrent [35].

For both structured and unstructured overlay networks, a desired property is locality-awareness which makes communications as local as possible. Such a property is important for P2P applications because it can reduce cross-domain traffic, avoid congestions and improve service performance [41]. It is recognized that locality-awareness can be achieved via intelligent peer selection whereby dense connections are enforced between nodes that are well connected to each other, i.e., short in terms of RTT or wide in terms of ABW. While the effectiveness of intelligent peer selection has been repeatedly justified [41, 81, 82], this chapter is the first to study whether locality-awareness can be achieved, in both structured and unstructured overlay networks, using rough and inaccurate knowledge of network proximity obtained via network inference of ratings.

4.3 Properties and Rating of Network Paths

4.3.1 Properties of Network Paths

While ultimately the performance of a network path should be judged by the quality of service (QoS) perceived by end users, it is important to define an objective metric that is directly measurable without user interventions. Among the commonly-used path properties mentioned earlier, this chapter only discusses round-trip time (RTT) and available bandwidth (ABW) due to the availability of such data and the relatively rare occurrence of packets losses in the Internet [24].

4.3.1.1 Round-Trip Time (RTT)

The RTT is one of the oldest and most commonly-used network measurements due partially to its ease of acquisition by using ping with little measurement overhead, i.e., few ICMP echo request and response packets between the sender and the target node. Although the one-way forward and reverse delays are not exactly the same due e.g. to the routing policies, the RTTs between two network nodes can be treated as symmetric.

4.3.1.2 Available Bandwidth (ABW)

The ABW is the remaining capacity of the bottleneck link on a path. Many comparative studies have shown that tools based on self-induced congestion are generally more accurate [24, 83]. The principle states that if the probing rate exceeds the available bandwidth over the path, then the probe packets become queued at some router, resulting in an increased transfer time. The ABW can then be estimated as the minimum probing rate that creates congestions or queuing delays. To this end, pathload [84] sends trains of UDP packets at a constant rate and adjusts the rate from train to train until congestions are observed at the target node, while pathchirp [85] reduces the probe traffic by varying the probe rate within a train exponentially.

Comparing to the RTT, the ABW measurement is much more costly and less accurate. For example, many studies revealed the pitfalls and the uncertainties of the ABW measurement [83, 86, 87]. In this chapter, we ignore these measurement issues and investigate how network inference can be performed[1] for ABW using a publicly available dataset [2] in which the measurements were acquired by pathchirp, a fairly accurate and widely-used tool [83]. In contrast to the RTT, the ABW is asymmetric and its measurement is inferred at the target node.

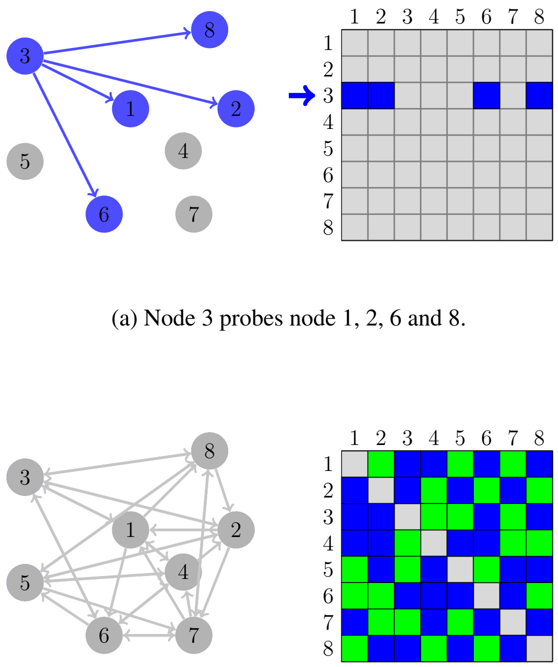
4.3.2 Rating of Network Paths

Generally, ratings can be acquired by vector quantization that partitions the range of the path property into R bins using $R-1$ thresholds, denoted by $\tau = \{\tau_1, \ldots, \tau_{R-1}\}$, and determines to which bins property values belong. The thresholds can be chosen evenly or unevenly according to the data distribution or the requirements of applications. For example, Google TV requires a broadband speed of 2.5Mbps or more for streaming movies and 10Mbps for High Definition contents [88]. Accordingly, $\tau = \{2.5\text{Mbps}, 10\text{Mbps}\}$ can be defined for ABW to separate paths of rating 1, 2 and 3.

Rating by quantization is directly applicable to RTTs whose measurements are cheap. For the ABW, it is preferable to directly obtain the rating measures without the explicit acquisition of the values in order to reduce measurement overhead. In general, as data acquisition undergoes the accuracy-versus-cost dilemma, stating that accuracy always comes at a cost, coarse-grained measurements are always cheaper to obtain than fine-grained ones. The advantage of the low measurement cost holds particularly for ABW whose measurement is costly.

Note that when $R = 2$, rating degenerates to binary classification of network paths which was studied in [32]. When R is very large, rating approaches the measurement of property values, which was studied in [33]. Thus, R results from a trade-off between the granularity and the measurement costs. A larger R means finer granularity but at the cost of more measurement overheads, and vice versa. Nevertheless, we will fix $R = 5$ in Section 4.4 where we examine the applicability of solutions to recommender systems. Such a choice is natural because it corresponds to the commonly-used 5-star rating system which categorizes performance into 5 discrete levels of “very poor”, “poor”, “ordinary”, “good” and “very good” quality . We will study the impact of different choices of R in Section 4.5.

[1] Strictly speaking, we do not infer the property of a path. Instead, we infer the property value that should have been obtained by some measurement tool.



(a) Node 3 probes node 1, 2, 6 and 8.

(b) A matrix with some missing entries is formed.

Figure 4.1: A matrix completion view of network inference. In (b), the blue entries are measured path properties and the green ones are missing. Note that the diagonal entries are empty as they represent the performance of the path from each node to itself which carries no information.

4.4 Network Inference of Ratings

This section describes the scalable acquisition of ratings on large networks by statistical inference.

4.4.1 Problem Statement

Formally, the inference problem is stated as follows. Consider a network of n end nodes, and define a path to be a routing path between a pair of end nodes. There are $O(n^2)$ paths in the network, and we wish to monitor a small subset of paths so that the performance of all other paths can be estimated. Note that here we seek to infer ratings, instead of values, of path properties. Besides reducing the measurement overheads, another motivation of network inference is that we can estimate the properties of those paths which are not measurable due e.g. to the lack of experimental controls.

This inference problem has a matrix completion form where a partially observed matrix is to be completed [89]. In this context, a matrix, denoted by X, is constructed, with its entry x_{ij} being the rating of the path from node i to node j. Each node randomly probes a few other nodes and measures the ratings of the paths to them. The measurements are put in the corresponding entries of X, and the missing entries are the ratings of those unmeasured paths and need to be estimated, illustrated in Figure 4.1. Note that X is a square matrix when it contains the performance of all pairwise paths in a network. However, for situations where one set of nodes in a network probe another set of nodes in the same or a different network, X becomes non-square. Such situations arise in content distribution networks (CDNs) where a set of users probe a set of servers [90].

In order for network inference to be feasible, network measurements on different paths have to be correlated with or dependent on each other. Otherwise, the inference would be impossible or suffer from large errors, regardless the inference schemes used. The correlations between network measurements exist, because Internet paths with nearby end nodes often overlap or share bottleneck links due to the simple topology of the Internet core. The redundancy of link usage across paths causes the performance of many paths to be correlated [27–29]. For example, the congestion at a certain link causes the delays of all paths that traverse this link to increase jointly. A direct consequence of such correlations is that the related performance matrix occurs to have a low-rank characteristic [33, 73] which further enables the inference problem to be solved by matrix factorization techniques, shown in the following sections. The low-rank characteristic of performance matrices of RTT and ABW is illustrated by the spectral plots in Figure 4.2. It can be seen that the singular values of these matrices decrease fast, which is an empirical justification of the low-rank phenomenon.

4.4.2 Connections to Recommender Systems

The problem of network inference bears strong similarities to the problem of recommender systems which predicts the preference of users to items such as music, books, or movies [40, 91]. Table 4.1 shows an example of a recommender system. Consider that network nodes are users and that each node treats other nodes as items or “friends”. The performance of a network path is then actually a “friendship” measure of how one end node would like to contact the other end node of the path. In particular, the task of intelligent peer selection can be viewed as a “friend” recommendation task. Guided by this insight the rapid advances

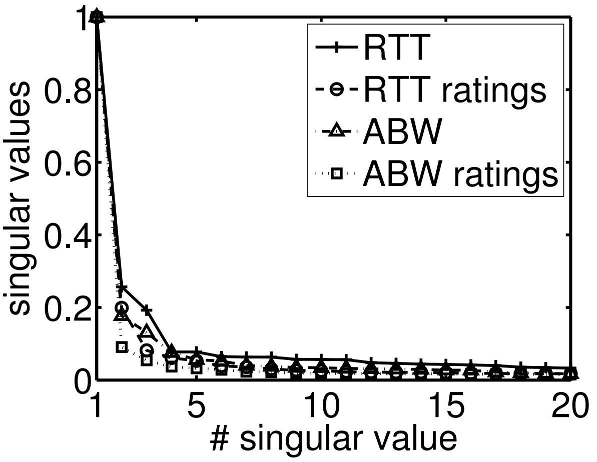


Figure 4.2: The singular values of a 2255 × 2255 RTT matrix, extracted from the Meridian dataset [1], and a 201 × 201 ABW matrix, extracted from the HP-S3 dataset [2], and of their corresponding rating matrices. The rating matrices are obtained by thresholding the measurement matrices with $\tau = \{20\%, 40\%, 60\%, 80\%\}$ percentiles of each dataset. The singular values are normalized so that the largest ones equal 1.

made for recommender systems can be applied to our inference problem.

Table 4.1: An example of a recommender system.

	item1	item2	item3	item4	item5	item6
user1	5	3	4	1	?	?
user2	5	3	4	1	5	2
user3	5	?	4	1	5	3
user4	1	3	2	5	1	4
user5	4	?	4	4	4	?

In particular, research on recommender systems was largely motivated by the *Netflix* prize which was an open competition initiated in 2006 for algorithms that predict the preference, quantized by the 5-star rating system, of users to movies [92]. A grand prize of one million dollar was to be given to the first algorithm that improves the accuracy of Netflix's own algorithm *cinematch* by 10%. The prize had attracted a lot of efforts and attempts before it was given to the BellKor's Pragmatic Chaos team in 2009 which achieved an improvement of 10.21%. In what follows, the prize-winning solution is called *BPC*. Specifically, BPC integrated two classes of techniques based on neighborhood models and on matrix factorization. Neighborhood models exploit the similarities between users and between items. For example, two users are considered similar if they rate a set of items similarly. Meanwhile, two items are considered similar if they are given similar ratings by a set of users. Although interesting, neighborhood models are infeasible for network inference, because, to compute the similarities between network nodes, they must probe a number of common nodes, which is impossible when the nodes being probed are randomly selected[2]. Thus, we focus below on the matrix factorization models in BPC.

4.4.3 Matrix Factorization

Matrix factorization (MF) exploits the low-rank nature of matrices of real-world data. Mathematically, a n by n matrix of rank r, where $r \ll n$, has only r non-zero singular values and it can be factorized as

$$X = UV^T, \tag{4.4.1}$$

where U and V are matrices of size $n \times r$. In practice, due to data noise, X is often full-rank but with a rank r dominant component. That is, X has only r significant singular values and the other ones are negligible. In this case, a rank-r matrix $\hat{X}$ can be found that approximates X with high accuracy, i.e.,

$$X \approx \hat{X} = UV^T. \tag{4.4.2}$$

MF can be used for solving the problem of matrix completion, which generally minimizes an objective function of the following form:

$$\min \sum_{ij \in \Omega} l(x_{ij}, u_i v_j^T), \tag{4.4.3}$$

where Ω is the set of observed entries, x_{ij} is the ijth entry of X, and u_i and v_j are the ith and jth row of U and of V respectively. l is a loss function that penalizes the difference between the two inputs. In words, we search for (U,V) so that $\hat{X} = UV^T$ best approximates X at the observed entries in Ω. The unknown entries in X are predicted by

$$\hat{x}_{ij} = u_i v_j^T, \text{ for } ij \notin \Omega. \tag{4.4.4}$$

Note that $\hat{x}_{ij}$ is real-valued and has to be rounded to the closest integer in the range of $\{1, R\}$ for ordinal rating. Figure 4.3 illustrates MF for matrix completion.

Below, we introduce various MF models that were integrated in BPC including RMF, MMMF and NMF.

[2] However, if we accept to depart from a fully distributed solution to the problem, it is possible to deploy some landmark nodes in the network for other nodes to probe. Thus, each node knows its performance information to/from the common landmarks. In such situations, the neighborhood models become applicable and are interesting to study, which is left as future work.

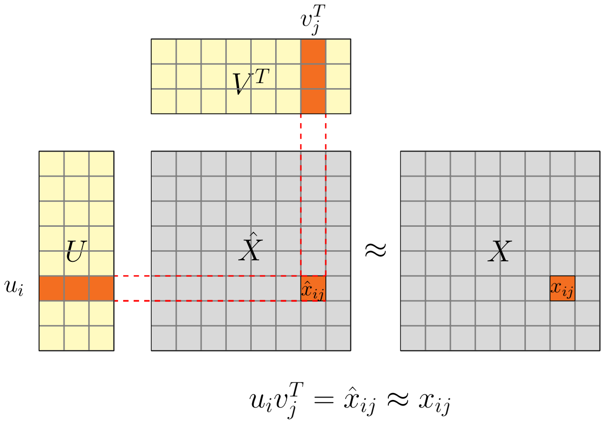


Figure 4.3: Matrix factorization.

4.4.3.1 RMF

Regularized matrix factorization (RMF) [40] adopts the widely-used square loss function and solves

$$\min \sum_{ij \in \Omega} (x_{ij} - u_i v_j^T)^2 + \lambda \sum_{i=1}^{n} (u_i u_i^T + v_i v_i^T). \tag{4.4.5}$$

The second term is the regularization which restricts the norm of U and V so as to prevent overfitting, and λ is the regularization coefficient.

4.4.3.2 MMMF

Max-margin matrix factorization (MMMF) solves the inference problem by ordinal regression [93] which relates the real-valued estimate $\hat{x}_{ij}$ to the ordinal rating x_{ij} by using $R-1$ thresholds $\{\theta_1, \ldots, \theta_{R-1}\}$. More specifically, MMMF requires the following constraint to be satisfied for each x_{ij}, $ij \in \Omega$,

$$\theta_{c-1} < \hat{x}_{ij} = u_i v_j^T \leqslant \theta_c, \ \ \text{for } x_{ij} = c, \ \ 1 \leqslant c \leqslant R. \tag{4.4.6}$$

For simplicity of notation $\theta_0 = -\infty$ and $\theta_R = +\infty$. In words, the value of $\hat{x}_{ij}$ does not matter, as long as it falls in the range of $\{\theta_{c-1}, \theta_c\}$ for $x_{ij} = c$, $1 \leqslant c \leqslant R$. Here we set the thresholds as $\{1.5, 2.5, 3.5, 4.5\}$ for $R = 5$. Thus, the constraint in eq. 4.4.6 means that, for example, if $x_{ij} = 2$, then it is required that $1.5 < \hat{x}_{ij} < 2.5$ so that $\hat{x}_{ij}$ will be rounded to 2. Whether $\hat{x}_{ij}$ is 2, 2.2 or 1.6 makes no difference.

Thus, we penalize the violation of the constraint in eq. 4.4.6 for each x_{ij} and minimize the following objective function

$$\min \sum_{ij\in\Omega} \sum_{c=1}^{R-1} l(T_{ij}^c, \theta_c - u_i v_j^T) + \lambda \sum_{i=1}^{n} (u_i u_i^T + v_i v_i^T), \tag{4.4.7}$$

where $T_{ij}^c = 1$ if $x_{ij} \geqslant c$ and -1 otherwise. Essentially, the objective function in eq. 4.4.7 consists of $R-1$ binary classification losses, each of which compares an estimate $\hat{x}_{ij}$ with a threshold θ_c in $\{\theta_1, \ldots, \theta_{R-1}\}$. For example, for $x_{ij} = 2$, it is required that $\hat{x}_{ij} > 1.5$ and $\hat{x}_{ij} < 2.5$, $\hat{x}_{ij} < 3.5$, $\hat{x}_{ij} < 4.5$. Each violation of these four constraints is penalized by $l(T_{ij}^c, \theta_c - u_i v_j^T)$, with

$$T_{ij}^c = \begin{cases} -1 & \text{for } c = 1 \\ 1 & \text{for } 2 \leqslant c \leqslant 4 \end{cases} \tag{4.4.8}$$

indicating the correct sign of $(\theta_c - u_i v_j^T)$.

Note that l in eq. 4.4.7 can be any classification loss function, among which the smooth hinge loss function is used [93], defined as

$$l(x,\hat{x}) = \begin{cases} 0 & \text{if } x\hat{x} \geqslant 1 \\ \frac{1}{2}(1 - x\hat{x})^2 & \text{if } 0 < x\hat{x} < 1 \\ \frac{1}{2} - x\hat{x} & \text{if } x\hat{x} \leqslant 0 \end{cases} \tag{4.4.9}$$

4.4.3.3 NMF

Non-negative matrix factorization (NMF) [94] incorporates an additional constraint that all entries in U and V have to be non-negative so as to ensure the non-negativity of $\hat{X}$. Besides, NMF uses the divergence as the loss function, defined as

$$D(X||\hat{X}) = \sum_{ij\in\Omega} (x_{ij} \log \frac{x_{ij}}{\hat{x}_{ij}} - x_{ij} + \hat{x}_{ij}). \tag{4.4.10}$$

Thus, NMF solves

$$\min D(X||UV^T) + \lambda \sum_{i=1}^{n} (u_i u_i^T + v_i v_i^T), \text{ s.t. } (U,V) \geqslant 0. \tag{4.4.11}$$

4.4.3.4 MF Ensembles

Instead of using one of these MF models, BPC integrated all of them in an ensemble framework, which is the root of its success. The idea is to learn multiple predictors simultaneously, by using different MF models (RMF, MMMF and NMF) and by setting different parameters for each MF model, and combines their outputs, by voting or averaging, for prediction [95, 96]. Intuitively, the power of ensemble methods comes from the "wisdom of the crowd", which says that a large group's aggregated answer to a question is generally found to be as good as, and often better than, the answer given by any of the individuals within the group [97].

Besides improving accuracy, ensemble models allow to compute the variance of each prediction, by computing the variance of the predictions by different predictors. Variance indicates the uncertainty of the prediction: large variance means that different predictors disagree with each other and we are thus less certain about the combined prediction[3]. Such information can be exploited in e.g. intelligent peer selection so that we can choose, among nodes with high ratings, those with small variance as they are more certain.

4.4.4 MF for Network Inference

4.4.4.1 Inference By Stochastic Gradient Descent

We adopted Stochastic Gradient Descent (SGD) for solving all MF models. In short, we pick x_{ij} in Ω randomly and update u_i and v_j by gradient descent to reduce the difference between x_{ij} and $u_i v_j^T$. SGD is particularly suitable for network inference, because measurements can be acquired on demand and processed locally at each node. We refer the interested readers to [32, 33] for the details of the inference by SGD.

4.4.4.2 Neighbor Selection

We also adopted the common architecture that each node randomly selects k nodes to probe, called neighbors in the sequel. That is, each node measures the properties of the paths from itself to the k neighbors and predicts the other paths by using MF-based inference schemes.

The choice of k is the result of a trade-off between accuracy and measurement overhead. On one hand, increasing k always improves accuracy as we measure more and infer less. On the other hand, the more we measure, the higher the overhead is. Thus, we vary k for networks of different sizes so as to control the number of paths being monitored to be a certain percentage of the total number of paths in a network. In particular, we require k to be no smaller than 10 so that a certain amount of information about the performance

[3] While we build our confidence on the variance of a prediction, it does not mean that smaller variance leads to better accuracy. Instead, it means that the combined prediction makes more sense if the variance is small, i.e. if different predictors agree with each other.

at each node is guaranteed. This leads to less than 5% of available measurements for a network of about two hundred nodes, which is the smallest dataset we used in the chapter. For large networks of a few thousand nodes, we increase k so that about 1% of the paths are monitored. As the largest dataset we used in this chapter has less than 5000 nodes, k is no larger than 50. We consider that such setting of k leads to sparse available measurements that is affordable for large networks.

4.4.4.3 Rank r

The rank r is an important parameter and can only be determined empirically. On the one hand, r has to be large enough so that the dominant components in X are kept. On the other hand, a higher-rank matrix has less redundancies and requires more data to recover, increasing measurement overheads. Our experiments show that empirically, $r = 10$ is a good choice, given the sparse available measurements.

4.4.5 Comparison of Different MF Models

This section compares different MF models on our network inference problem. In the evaluations, we set $R = 5$, which was also used in the Netflix prize.

The comparison was then performed on the following publicly available datasets:

- **Harvard** contains dynamic RTT measurements, with timestamps, between 226 Azureus clients deployed on PlanetLab [98];
- **Meridian** contains static RTT measurements between 2500 network nodes obtained from the Meridian project [1];
- **HP-S3** contains static ABW measurements between 231 PlanetLab nodes [2].
- **YouTube** contains static RTT measurements from 441 PlanetLab nodes to 4816 YouTube servers [90].

In the simulations, the static measurements in Meridian, HP-S3 and YouTube are used in random order, whereas the dynamic measurements in Harvard are used in time order according to the timestamp of each measurement.

We adopted the evaluation criterion, *Root Mean Square Error* (RMSE), given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \hat{x}_i)^2}{n}}. \tag{4.4.12}$$

which was used in the Netflix prize. As RMSE is the average estimation error, the smaller it is, the better.

4.4.5.1 Obtaining Ratings

The first step is to obtain ratings on a scale of $\{1,5\}$ from the raw measurements. To this end, the range of a path property is partitioned by the rating threshold $\tau = \{\tau_1,\dots,\tau_4\}$. τ is set by two strategies:

- Strategy 1: set τ by the $\{20\%,40\%,60\%,80\%\}$ percentiles of each dataset.
  - Harvard: $\tau = \{48.8,92.2,177.2,280.3\}$ms
  - Meridian: $\tau = \{31.6,47.3,68.6,97.9\}$ms
  - HP-S3: $\tau = \{12.7,34.5,48.8,77.9\}$Mbps
  - YouTube: $\tau = \{38.1,91.1,131.3,192.4\}$ms
- Strategy 2: partition evenly the range between 0 and a large value manually selected for each dataset.
  - Harvard: $\tau = \{75,150,225,300\}$ms
  - Meridian: $\tau = \{25,50,75,100\}$ms
  - HP-S3: $\tau = \{20,40,60,80\}$Mbps
  - YouTube: $\tau = \{50,100,150,200\}$ms

4.4.5.2 Results

Throughout the chapter, the MF parameters are set as follows: for RMF, MMMF and NMF, the regularization coefficient $\lambda = 0.1$ and the rank $r = 10$. For MF ensembles, we generated for each MF model (RMF, MMMF and NMF) 6 predictors using different parameters, i.e. r ranges from 10 to 100 and λ ranges from 0.01 to 1, as described in [96]. For the neighbor number, $k = 10$ for Harvard of 226 nodes, leading to about 4.42% available measurements; $k = 32$ for Meridian of 2500 nodes, leading to about 1.28% available measurements; $k = 10$ for HP-S3 of 231 nodes, leading to about 4.33% available measurements; $k = 50$ for YouTube of 4816 servers, leading to about 1.04% available measurements. Thus, we collect k measurements at each node and perform the inference using different MF models. The evaluation was done by comparing the inferred ratings of those unmeasured paths with their true ratings, calculated by RMSE defined above.

Table 4.2 shows the RMSEs achieved by different MF models and on different datasets. We can see that while RMF generally outperforms MMMF and NMF, MF ensembles perform the best at the cost of more computational overheads due to the maintenance of multiple MF predictors. Note that all MF models achieved fairly accurate results with the RMSE less than 1. In comparison, for the dataset in the Netflix prize, the RMSE achieved by the Netflix's cinematch algorithm is 0.9525 and that by BPC is 0.8567 [92]. While the RMSEs on different datasets are not comparable, it shows that in practice, the prediction with an accuracy of the RMSE less than 1 for ratings on a scale of $\{1,5\}$ is already accurate enough to be used for recommendation tasks. Note that from Table 4.2, it appears that Strategy 2 which partitions the range of the property evenly produced smaller RMSEs than Strategy 1 which set τ by certain percentiles of the data. However, the RMSEs by different strategies are not comparable, because the evaluations were performed on different rating data generated by different strategies. Nevertheless, Strategy 2 may create unbalanced portions of ratings. For example, we may have no path of rating 1 but a lot of paths of rating 2, which will never occur for Strategy 1. For this reason, Strategy 1 is used by default in the rest of the chapter.

Table 4.2: RMSE on different datasets.

τ: Strategy 1	Harvard	Meridian	HP-S3	YouTube
RMF	0.934	0.831	0.675	0.923
MMMF	0.969	0.863	0.686	0.957
NMF	0.977	0.904	0.682	0.969
MF Ensembles	0.920	0.821	0.661	0.901
τ: Strategy 2	Harvard	Meridian	HP-S3	YouTube
RMF	0.920	0.776	0.669	0.910
MMMF	0.919	0.810	0.670	0.944
NMF	0.932	0.829	0.674	0.961
MF Ensembles	0.904	0.766	0.653	0.873

Overall, RMF is lightweight and suits well for online deployment in P2P applications, and is thus used in Section 4.5 for the case study on overlay construction and routing. Table 4.3 shows the confusion matrices achieved by RMF on the four datasets. In these matrices, each column represents the predicted ratings, while each row represents the actual ratings. Thus, the diagonal entries represent the percentage of the correct prediction, and the off-diagonal entries represent the percentage of "confusions" or mis-ratings. For example, the entry at $(2,2)$ represents the percentage of the rating-2 paths which are correctly predicted as rating-2, and the entry at $(2,3)$ represents the percentage of the rating-2 paths which are wrongly predicted as rating-3, i.e. the confusions from rating-2 to rating-3. It can be seen that while there are mis-ratings, most of them have a small error of $|x_{ij} - \hat{x}_{ij}| = 1$, marked as shaded entries in the confusion matrices in Table 4.3. This means that the mis-ratings are under control. For example, a rating-5 path may be wrongly predicted as 4, but seldom as 3, 2 or

1, since the entries at $(5,3)$, $(5,2)$ and $(5,1)$ in all confusion matrices are small.

Table 4.3: Confusion matrices.

Harvard		Predicted				
		1	2	3	4	5
Actual	1	**68%**	28%	2%	1%	1%
	2	18%	**60%**	20%	1%	1%
	3	3%	13%	**66%**	17%	1%
	4	3%	4%	16%	**67%**	10%
	5	3%	3%	5%	43%	**46%**

Meridian		Predicted				
		1	2	3	4	5
Actual	1	**78%**	18%	3%	1%	0%
	2	8%	**59%**	30%	3%	0%
	3	1%	18%	**60%**	20%	1%
	4	1%	3%	33%	**59%**	4%
	5	1%	1%	12%	59%	**27%**

HP-S3		Predicted				
		1	2	3	4	5
Actual	1	**92%**	6%	1%	1%	0%
	2	11%	**65%**	22%	2%	0%
	3	1%	20%	**68%**	11%	0%
	4	0%	3%	33%	**58%**	6%
	5	0%	1%	10%	55%	**34%**

YouTube		Predicted				
		1	2	3	4	5
Actual	1	**80%**	17%	1%	1%	1%
	2	13%	**66%**	20%	1%	0%
	3	2%	13%	**69%**	15%	0%
	4	3%	8%	33%	**52%**	4%
	5	2%	7%	12%	46%	**32%**

Note that we also evaluated another matrix completion method, namely LMaFit[4], which was used in [74] for traffic matrix completion and found that it performed much worse than the MFs used in this chapter. For example, the RMSE by LMaFit on Meridian, HP-S3 and YouTube are 1.357, 1.139 and 1.422 respectively. Note also that many general matrix completion methods including LMaFit take as input an incomplete matrix and make updates

[4] The source code is downloaded from `http://lmafit.blogs.rice.edu/`.

on the entire matrix. This means that they can not process the dynamic measurements in the Harvard dataset, nor can they be decentralized. It is however worth mentioning that LMaFit runs much faster than the MFs based on SGD.

4.4.5.3 Observations

Our experiments reveal a useful effect of ordinal rating. In practice, MF is known to be sensitive to outliers such as those unusually large and small values in the datasets. However, as ratings are quantized measures, measurement outliers are naturally handled by truncating the large and small property values to 1 and R. A direct consequence is that MF becomes insensitive to parameter settings, as the inputs are always in a relatively small and fixed range.

In the experiments, we observed that there exist nodes which have a poor rating with all their k neighbors. The likely reason is that those nodes have a poor access link, i.e. the link by which a node is connected to the Internet core is the bottleneck, which causes all connected paths to perform poorly. Thus, we calculate the mean rating and the standard deviation of the measured paths associated with each node, denoted by μ_i and σ_i. If σ_i is small, we do not consider that node i provides useful information about the performance of the network, and will simply predict the ratings of all paths of node i as μ_i. In our datasets, the non-informative nodes are rare (i.e. no more than 10 in each dataset), and thus the “mean rating” trick improved the accuracy only slightly. However, we incorporate it to detect non-informative nodes so that they won’t pose a problem in any case.

4.4.5.4 Discussions on Scalability

The MF models have proved to work well for recommender systems on extremely large matrices such as the one in the Netflix prize. Thus, there is no scalability issue when running MFs on performance matrices constructed on networks even with millions of nodes. However, two practical questions need to be answered when deploying MFs on real, large networks:

- How many measurements are required to make predictions with a decent accuracy, i.e., an RMSE at least smaller than 1?
- How fast do MFs run on such a large matrix?

Regarding the first question, a theory in matrix completion states that a $n \times n$ matrix of rank r can be exactly or accurately recovered, with high probability, from as few as $O(nr\log(n))$ randomly sampled entries [89]. This means that each node would have to probe $O(r\log(n))$

neighbors, which scales fairly well in theory. Nevertheless, we are interested in evaluating whether such a bound holds or is required for MF-based network inference on large networks. Regarding the second question, it is known that MFs based on SGD are computationally efficient, as SGD involves only vector operations [32,33], and empirically converge fast for large matrices, as demonstrated in BPC. We leave the study of these issues as future work, because it would require really large-scale network measurement data.

4.5 Case Study: Locality-Aware Overlay Construction and Routing

With the techniques of rating-based network measurement and inference presented above, the remaining issue is the usability in Internet applications. There are two questions to be answered:

- While Section 4.4.5 shows that the inference by various MF models can achieve at least an accuracy of RMSE less than 1, it is natural to ask whether such an accuracy is acceptable for applications.
- It is critical to choose a proper granularity for rating network paths, because although a finer granularity leads to more informative measurement, it also means more measurement overheads which may overweigh the benefit of exploiting the knowledge of network proximity. Thus, it is also natural to ask whether more fine-grained ratings always improve the performance of applications.

To answer these questions, we perform a case study on locality-aware overlay construction and routing and investigate whether locality-awareness can be achieved by using inferred ratings of path properties such as RTT and ABW. More specifically, we consider Pastry [42] and BitTorrent [35] which are typical structured and unstructured overlay networks and are known to enjoy the property of locality-awareness. Both Pastry and BitTorrent rely on an outside program that acquires network path properties. For example, Pastry uses measurement tools such as traceroute for hop count or ping for RTT [42], and BitTorrent uses Vivaldi to infer RTTs [98]. Here, we are interested in knowing whether our MF-based inference schemes can serve as the outside program in Pastry and BitTorrent[5] and the impact of the rating granularity on their performance. To simplify the evaluation, we only employ

[5] Note that investigating how MF-based inference can actually be incorporated in the Pastry and BitTorrent protocols is beyond the scope of this chapter. However, [32, 33] showed that MF can be implemented in a fully decentralized manner, with the same architecture as Vivaldi [26] which has been incorporated in BitTorrent [98]. This indicates that our MF-based inference schemes can be seamlessly used in BitTorrent, with no extra overhead required. We refer the interested readers to [32,33] for the details of the decentralized architecture and implementations of our MF-based inference schemes.

RMF in this section due to its accuracy and simplicity which facilitates its deployment in P2P applications.

4.5.1 Pastry

Pastry is a classic structured overlay network for the implementation of a DHT where the key-value pairs are stored in a P2P network in an organized manner so that queries can be achieved within $O(\log n)$ hops of routing, where n is the number of nodes. Here, we drop the description but refer the interested readers to [42] for the construction and routing algorithms of Pastry. We mention that Pastry determines the best routes by using the proximity knowledge.

In the simulations, we predicted ratings using RMF on the same datasets and with the same configuration as described in Section 4.4.5, except that we varied the rating levels from $R = 2$, $R = 2^2$,..., to $R = 2^8$, instead of $R = 5$ in the previous sections. For comparison, we also ran RMF to predict values of path properties, which is the most fine-grained measure. We then built the routing tables in Pastry using respectively no proximity knowledge, inferred ratings, inferred values, as well as the true measurements in the original datasets. We refer to Pastry using no proximity knowledge as $R = 0$ and using inferred values as $R = \infty$. Pastry using true measurements is the ideal case where the best routes can be found at the cost of $O(n^2)$ active measurements. In the implementation of Pastry, considering that there are at most 2500 nodes in our datasets, the node Id space is $N = 2^{14}$. Other parameters are: the base is $B = 2$, the leaf set size is $L = 2^B$, the neighbor set size is $M = 2^{B+1}$, and the routing table size is $T = (\log_L N, L-1) = (7,3)$.

After construction, we simulated $100,000$ lookup messages, i.e. queries, which were routed from randomly chosen nodes with randomly chosen keys. Then, we compared the routing performance of Pastry under $R = \{0, 2^1, \dots, 2^8, \infty\}$ respectively with that of using true measurements. We measured the routing performance as the average of a routing metric, RTT for Harvard and Meridian and ABW for HP-S3, of each query and calculated the ratio between the different cases of R and the ideal case of using true measurements, called *stretch*. As Pastry can find the best routes when using true measurements, stretch for the routing performance is greater than 1 for RTT and smaller than 1 for ABW. The closer it is to 1, the better.

Figure 4.4 shows the stretches of both the routing performance and the hop count. We make the following observations. First, the stretch of the hop count is very close to 1, which is expected because Pastry always produces routes of $O(\log n)$ hops, with or without using the proximity knowledge. Second, also as expected, the routing performance of using inferred knowledge is worse than that of using true measurements, but is better than using no knowledge at all ($R = 0$), even for the case of $R = 2$ which leads to the most coarse-grained ratings. This shows that network inference of ratings is accurate enough to be used in Pastry. Third, increasing R after R reaches 2^3 or 2^4 cannot further improve the routing performance of Pastry. This shows that rating with finer granularities is not always beneficial for applications, let alone the extra costs due to more fine-grained measurements. Last, the routing performance of using inferred values ($R = \infty$) is a little worse than that of using inferred ratings with $R = 2^4$. This seems to suggest that the inference of property values was affected by measurement outliers, which further degraded the performance of Pastry. In contrast, outliers were naturally filtered out by rating-based measurements.

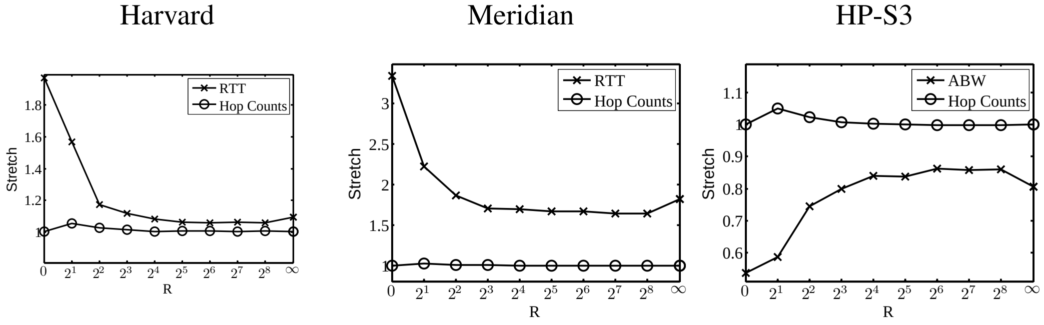


Figure 4.4: Stretch of the routing performance of Pastry, defined as the ratio between the routing metric of Pastry using inferred ratings and of Pastry using true measurements. Note that $R = 0$ means that no proximity knowledge is used, and $R = \infty$ means that inferred values are used.

4.5.2 BitTorrent

BitTorrent is perhaps the most popular unstructured overlay network and is widely used for file sharing [35]. The protocol adopts a random peer selection procedure which causes large inter-domain traffic and degrades application performance. It is well known that these issues can be overcome by intelligent peer selection with the use of the proximity knowledge [41,81,82].

In the simulations, we inferred ratings and values of path properties using RMF as described in the above section. Then, we constructed overlay networks in BitTorrent with a standard scheme whereby a tracker node randomly proposes to each peer 50 peers in the network, among which 32 are selected. Ideally, we would like to connect each peer with those best performing peers, i.e. shortest for RTT and widest for ABW. However, such best peer selection strategy leads to the problem of reachability that some nodes are not reach-

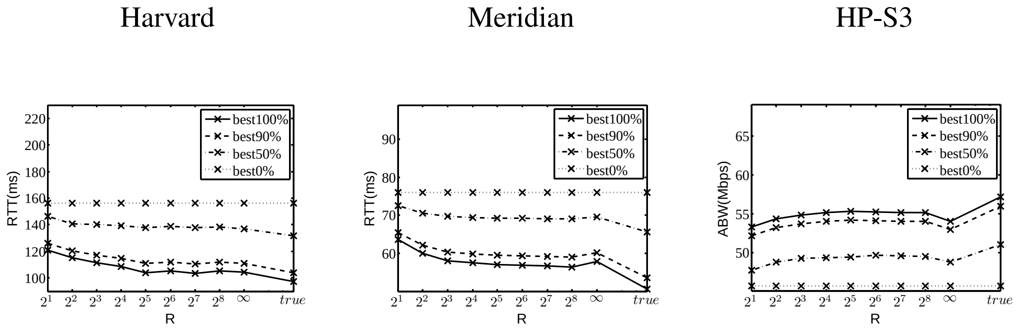


Figure 4.5: Performance of peer selection for BitTorrent, calculated as the average link performance between each pair of selected peers. Note that "*true*" means that true measurements are used.

able by some other nodes[6]. Our simulations showed that there could be 10% ∼ 15% of the end-to-end obstructed paths. To overcome this problem, we came up with a mixed strategy whereby $p\%$ of the peers execute best peer selection and the other ones execute random peer selection, referred to as *best p% peer selection*[7]. We tested $p = \{100, 90, 50, 0\}$, with $p = 100$ being essentially the strategy of best peer selection and $p = 0$ being random peer selection. Note that the reachability problem is already avoided when $p = 90$.

We measured the performance of peer selection by the average performance of the links between each pair of peers selected by the peer selection procedure, shown in Figure 4.5. It can be seen that the performance of peer selection improves as more nodes execute the strategy of best peer selection. This shows that we can reduce download time or avoid congestions in BitTorrent by exploiting inferred ratings of RTT or ABW. It can also be seen that, while $R = 2^4$ or 2^5 appear to be optimal, which is similar to Pastry, the granularity seems to have less impact on BitTorrent. This is probably due to the protocol that requires each peer to select 32 peers out of 50 candidates, thus leaving us with limited choices for peer selection. Note that BitTorrent is particularly concerned with the performance of peer selection, because chunks of files are frequently exchanged between direct neighbors.

[6] In an overlay network, the links between nodes are directed. For example, there may be a link from A to B, because A chooses B as its neighbor. However, there may not be a link from B to A, because B may not choose A as its neighbor. Thus, the problem of reachability doesn't mean that the overlay network is disconnected. Instead, it means that some end-to-end paths are obstructed and that packets can only be routed in one direction between some end nodes.

[7] An alternative strategy for best $p\%$ peer selection is to let each node select $p\%$ best peers and $(100 - p)\%$ random peers. We tested both strategies and found that they achieved statistically similar results.

4.5.3 Remarks

Our case study is encouraging, showing that inferred ratings are accurate enough to be beneficial to applications such as Pastry and BitTorrent. Empirically, a coarse granularity of $R = 2^4$ appears to be satisfactory for our datasets, which is acceptable for RTT whose measurement is cheap. However, for ABW, the improvement by increasing R from 2^3 to 2^4 comes at the cost of more probe flows and thus more overheads, which may not be considered worthy. In practice, the choice of R has to take into account not only the performance of applications but also the measurement budget.

We measured the inference accuracy by RMSE which reflects the overall accuracy of the prediction for each path. Such metric is more meaningful for applications where the inference accuracy of every path is equally important. For the task of intelligent peer selection where it is only important to ensure that the selected paths have a high rating, it may not be worth pushing the metric of RMSE to the limit. This insight shows that a prediction system should be evaluated from the applications' point of view, which highlights the importance of carrying out case studies on real applications.

Note that the case study focused on the impact of the accuracy of rating inference and the impact of the granularity in rating-based network measurement. Thus, we performed no comparison with other inference approaches, as none can deal with ratings of different path properties. For example, Vivaldi [26] infers RTT values by Euclidean embedding and was already shown to be less accurate than our MF-based approach when using the same amount of RTT measurements [33]. Tomography-based approaches [27–29] do not work on non-additive properties such as ABW and require the routing information of the network which is not available in our scenarios. In contrast, our MF-based approach is flexible in dealing with both additive and non-additive properties and in dealing with both property values and ratings of different levels, which is a unique feature that distinguishes it from all previous approaches.

4.6 Chapter Summary

This chapter presents a novel concept of rating network paths, instead of measuring values of path properties, and investigates the inference of ratings by solutions designed for recommender systems, particularly by a class of matrix factorization techniques, which were found to work well. The case study on locality-aware overlay construction and routing highlights the usability of rating-based network measurement and inference by Internet applications. These studies reveal the advantages of ratings: they are informative, have low measurement cost and are easy to process in applications.

Chapter 5

Conclusion

This thesis demonstrate that novel data interpolation methods can make original expensive and, even, unobtainable data collection tasks become affordable and feasible. Three specific problems are considered.

We revisit the **AS path inference problem** from the complex network perspective. A brand new constraint is proposed based on the fact the AS paths respect the underlying geometric structure of the Internet. Resulting two new AS path inference algorithms, HyperPath and Valley-free HyperPath, have $O(K)$ complexity to infer certain end-to-end AS path and can run locally. Intensive evaluation on the ground truth AS paths shows that HyperPath method can not only outperform no policy method, it can be superior to AS relationships based method by being blind to the AS relationships information. The Valley-free HyperPath method outperforms both AS relationships based method and KnownPath method. Moreover, two new algorithms are immune to the fail-to-detect problem, by which the benchmark methods are always haunted. We also simulate BitTorrent P2P applications to show the potential of our methods on inter-domain Internet traffic reduction.

We proposed a new CS-based approach for **environment reconstruction problem** in WSN. By exploiting the sparsity in spatio-temporal difference and introducing additional informative "observations" with probabilistic model, the proposed PMEST-CS method can achieve superior performance than the state of the art methods. We also proposed a new learning algorithm which can efficiently learn probabilistic model from highly incomplete data. Experimental results corroborate PMEST-CS method is very efficient and can achieve significant reconstruction quality gain over the state of the art CS-based methods.

We presents a novel concept of rating network paths, instead of measuring values of path properties, and investigates the inference of ratings by solutions designed for recommender systems, particularly by a class of matrix factorization techniques, which were found to work well. The case study on **locality-aware overlay construction and routing** highlights the usability of rating-based network measurement and inference by Internet applications. These studies reveal the advantages of ratings: they are informative, have low measurement cost and are easy to process in applications.

Bibliography

[1] Bernard Wong, Aleksandrs Slivkins, and Emin Gün Sirer. Meridian: A lightweight network location service without virtual coordinates. In *Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '05, pages 85–96, New York, NY, USA, 2005. ACM.

[2] Praveen Yalagandula, Puneet Sharma, Sujata Banerjee, Sujoy Basu, and Sung-Ju Lee. S3: A scalable sensing service for monitoring large networked systems. In *Proceedings of the 2006 SIGCOMM Workshop on Internet Network Management*, INM '06, pages 71–76, New York, NY, USA, 2006. ACM.

[3] Gilman Tolle, Joseph Polastre, Robert Szewczyk, David Culler, Neil Turner, Kevin Tu, Stephen Burgess, Todd Dawson, Phil Buonadonna, David Gay, and Wei Hong. A macroscope in the redwoods. In *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, SenSys '05, pages 51–63, New York, NY, USA, 2005. ACM.

[4] Geoff Werner-Allen, Konrad Lorincz, Jeff Johnson, Jonathan Lees, and Matt Welsh. Fidelity and yield in a volcano monitoring sensor network. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 381–396, Berkeley, CA, USA, 2006. USENIX Association.

[5] A. Khan and Lawrence Jenkins. Undersea wireless sensor network for ocean pollution prevention. In *Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on*, pages 2–8, Jan 2008.

[6] Mo Li and Yunhao Liu. Underground coal mine monitoring with wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(2):10:1–10:29, April 2009.

[7] Xuefeng Liu, Jiannong Cao, Wen-Zhan Song, and Shaojie Tang. Distributed sensing for high quality structural health monitoring using wireless sensor networks. In *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*, pages 75–84, Dec 2012.

[8] J. Hawkinson and T. Bates. Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930 (Best Current Practice), March 1996. Updated

by RFCs 6996, 7300.

[9] Z. Morley Mao, Lili Qiu, Jia Wang, and Yin Zhang. On as-level path inference. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '05, pages 339–349, New York, NY, USA, 2005. ACM.

[10] Aditya Akella, Srinivasan Seshan, and Anees Shaikh. An empirical evaluation of wide-area internet bottlenecks. *SIGMETRICS Perform. Eval. Rev.*, 31(1):316–317, June 2003.

[11] Ji Li and Karen Sollins. Exploiting autonomous system information in structured peer-to-peer networks. In *In ICCCN*, pages 403–408. IEEE CS Press, 2004.

[12] Shansi Ren, Lei Guo, and Xiaodong Zhang. Asap: An as-aware peer-relay protocol for high quality voip. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems*, ICDCS '06, pages 70–, Washington, DC, USA, 2006. IEEE Computer Society.

[13] D K Lee, Keon Jang, Changhyun Lee, Gianluca Iannaccone, and Sue Moon. Path stitching: Internet-wide path and delay estimation from existing measurements. In *Proceedings of the 29th Conference on Information Communications*, INFOCOM'10, pages 201–205, Piscataway, NJ, USA, 2010. IEEE Press.

[14] S. Hasan, S. Gorinsky, C. Dovrolis, and R.K. Sitaraman. Trade-offs in optimizing the cache deployments of cdns. In *INFOCOM, 2014 Proceedings IEEE*, pages 460–468, April 2014.

[15] Jian Wu, Ying Zhang, Z. Morley Mao, and Kang G. Shin. Internet routing resilience to failures: Analysis and implications. In *Proceedings of the 2007 ACM CoNEXT Conference*, CoNEXT '07, pages 25:1–25:12, New York, NY, USA, 2007. ACM.

[16] Danny Dolev, Sugih Jamin, Osnat Mokryn, and Yuval Shavitt. Internet resiliency to attacks and failures under bgp policy routing. *Comput. Netw.*, 50(16):3183–3196, November 2006.

[17] Bgp ipv4/ipv6 looking glass servers. `http://www.bgp4.as/looking-glasses`.

[18] University of Oregon Route Views Project. `http://www.routeviews.org`.

[19] Ripe's Routing Information Service Raw Data Project. `http://www.ripe.net/data-tools/stats/ris/ris-raw-data`.

[20] Tony Bates, Philip Smith, and Geoff Huston. CIDR Report. `http://www.cidr-`

`report.org/as2.0`.

[21] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iplane: An information plane for distributed services. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 367–380, Berkeley, CA, USA, 2006. USENIX Association.

[22] Harsha V. Madhyastha, Ethan Katz-Bassett, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iplane nano: Path prediction for peer-to-peer applications. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'09, pages 137–152, Berkeley, CA, USA, 2009. USENIX Association.

[23] Linghe Kong, Dawei Jiang, and Min-You Wu. Optimizing the spatio-temporal distribution of cyber-physical systems for environment abstraction. In *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, pages 179–188, June 2010.

[24] Mark Crovella and Balachander Krishnamurthy. *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley & Sons, Inc., New York, NY, USA, 2006.

[25] T. S. Eugene Ng and Hui Zhang. Predicting internet network distance with coordinates-based approaches. In *In INFOCOM*, pages 170–179, 2001.

[26] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. *SIGCOMM Comput. Commun. Rev.*, 34(4):15–26, August 2004.

[27] Yan Chen, David Bindel, Hanhee Song, and Randy H. Katz. An algebraic approach to practical and scalable overlay network monitoring. *SIGCOMM Comput. Commun. Rev.*, 34(4):55–66, August 2004.

[28] D.B. Chua, E.D. Kolaczyk, and M. Crovella. Network kriging. *Selected Areas in Communications, IEEE Journal on*, 24(12):2263–2272, Dec 2006.

[29] Han Hee Song, Lili Qiu, and Yin Zhang. Netquest: A flexible framework for large-scale network measurement. *Networking, IEEE/ACM Transactions on*, 17(1):106–119, Feb 2009.

[30] Yun Mao, L.K. Saul, and J.M. Smith. Ides: An internet distance estimation service for large networks. *Selected Areas in Communications, IEEE Journal on*, 24(12):2273–2284, Dec 2006.

[31] Yongjun Liao, Pierre Geurts, and Guy Leduc. Network distance prediction based on decentralized matrix factorization. In *In Proc. of IFIP Networking*, 2010.

[32] Yongjun Liao, Wei Du, Pierre Geurts, and Guy Leduc. Decentralized prediction of end-to-end network performance classes. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 14:1–14:12, New York, NY, USA, 2011. ACM.

[33] Yongjun Liao, Wei Du, Pierre Geurts, and Guy Leduc. Dmfsgd: A decentralized matrix factorization algorithm for network distance prediction. *IEEE/ACM Trans. Netw.*, 21(5):1511–1524, October 2013.

[34] Jian Qiu and Lixin Gao. As path inference by exploiting known as paths. Technical report, In Proceedings of IEEE GLOBECOM, 2005.

[35] Vuze bittorrent. `http://www.vuze.com/`.

[36] E.J. Candes and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, Dec 2006.

[37] R.G. Baraniuk, V. Cevher, M.F. Duarte, and C. Hegde. Model-based compressive sensing. *Information Theory, IEEE Transactions on*, 56(4):1982–2001, April 2010.

[38] Volkan Cevher, Marco F. Duarte, Chinmay Hegde, and Richard Baraniuk. Sparse signal recovery using markov random fields. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 257–264. Curran Associates, Inc., 2009.

[39] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Min-You Wu, and Xue Liu. Data loss and reconstruction in sensor networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 1654–1662, April 2013.

[40] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, August 2009.

[41] I. Rimac E. Marocco, A. Fusco and V. Gurbani. Improving peer selection in peer-to-peer applications: myths vs. reality. *Internet Research Task Force Working Group*, December 2012.

[42] Antony I. T. Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg*, Middleware

'01, pages 329–350, London, UK, UK, 2001. Springer-Verlag.

[43] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush. 10 lessons from 10 years of measuring and modeling the internet's autonomous systems. *Selected Areas in Communications, IEEE Journal on*, 29(9):1810–1821, October 2011.

[44] Yuval Shavitt and Tomer Tankel. On the curvature of the internet and its usage for overlay construction and distance estimation, 2004.

[45] Robert Kleinberg. Geographic routing using hyperbolic space. In *in Proc. of INFOCOM*, pages 1902–1909, 2007.

[46] A. Cvetkovski and M. Crovella. Hyperbolic embedding and routing for dynamic graphs. In *INFOCOM 2009, IEEE*, pages 1647–1655, April 2009.

[47] D. Krioukov, F. Papadopoulos, M. Boguñá, and A. Vahdat. Efficient Navigation in Scale-Free Networks Embedded in Hyperbolic Metric Spaces. Technical report, arXiv cond-mat.stat-mech/0805.1266, May 2008.

[48] Lixin Gao and Feng Wang. The extent of as path inflation by routing policies, 2002.

[49] B. Quoitin and S. Uhlig. Modeling the routing of an autonomous system with c-bgp. *Netwrk. Mag. of Global Internetwkg.*, 19(6):12–19, November 2005.

[50] Lixin Gao. On inferring autonomous system relationships in the internet. *IEEE/ACM Trans. Netw.*, 9(6):733–745, December 2001.

[51] Josh Karlin, Stephanie Forrest, and Jennifer Rexford. Nation-state routing: Censorship, wiretapping, and bgp. `arXiv:0903.3218`, 2009.

[52] Aaron Johnson Nikita Borisov Matthew Caesar Joshua Juen, Anupam Das. Defending tor from network adversaries: A case study of network path prediction, 2014.

[53] Matthew Luckie, Bradley Huffaker, Amogh Dhamdhere, Vasileios Giotsas, and kc claffy. As relationships, customer cones, and validation. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 243–256, New York, NY, USA, 2013. ACM.

[54] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, k. claffy, and G. Riley. As relationships: Inference and validation. *ACM SIGCOMM Computer Communication Review (CCR)*, 37(1):29–40, Jan 2007.

[55] M. Gromov. *Hyperbolic groups*. Springer, 1987.

[56] M. R. Bridson and A. Häfliger. *Metric Spaces of Non-Positive Curvature*. Springer, 2009.

[57] AB. Adcock, B.D. Sullivan, and M.W. Mahoney. Tree-like structure in large social and information networks. In *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pages 1–10, Dec 2013.

[58] Feodor F. Dragan Muad Abu-Ata. Metric tree-like structures in real-life networks: an empirical study. `arXiv:1402.3364`, 2014.

[59] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, Sep 2010.

[60] Inferred as relationships dataset. `http://www.caida.org/data/request_user_info_forms/as_relationships.xml`.

[61] Yin Zhang, Matthew Roughan, Walter Willinger, and Lili Qiu. Spatio-temporal compressive sensing and internet traffic matrices. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, SIGCOMM '09, pages 267–278, New York, NY, USA, 2009. ACM.

[62] Swati Rallapalli, Lili Qiu, Yin Zhang, and Yi-Chao Chen. Exploiting temporal stability and low-rank structure for localization in mobile networks. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 161–172, New York, NY, USA, 2010. ACM.

[63] Intel Lab Data. `http://db.csail.mit.edu/labdata/labdata.html`, 2004.

[64] Atis Elsts, Farshid Hassani Bijarbooneh, Martin Jackobsson, and Konstantinos Sagonas. Demo abstract: ProFuN TG: A tool using abstract task graphs to facilitate the development, deployment and maintenance of wireless sensor network applications. *Proceedings of EWSN'15, the 12th European Conference on Wireless Sensor Networks*, pages 20–21, 2015.

[65] A. Gersho and R. M. Gray. Vector quantization and signal compression. 1992.

[66] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

[67] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(3):471–501, 2010.

[68] Jianqing Fan and Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

[69] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[70] Michael Grant and Stephen Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

[71] Michael Grant and Stephen Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[72] Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys Tutorials, IEEE*, 7(2):72–93, Second 2005.

[73] Liying Tang and Mark Crovella. Virtual landmarks for the internet. In *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, pages 143–152, New York, NY, USA, 2003. ACM.

[74] Gonca Gürsun, Natali Ruchansky, Evimaria Terzi, and Mark Crovella. Inferring visibility: Who's (not) talking to whom? In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 151–162, New York, NY, USA, 2012. ACM.

[75] Gonca Gürsun and Mark Crovella. On traffic matrix completion in the internet. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, pages 399–412, New York, NY, USA, 2012. ACM.

[76] M. Roughan, Yin Zhang, W. Willinger, and Lili Qiu. Spatio-temporal compressive sensing and internet traffic matrices (extended version). *Networking, IEEE/ACM Transactions on*, 20(3):662–676, June 2012.

[77] Petar Maymounkov and David Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 53–65, London, UK, UK, 2002. Springer-Verlag.

[78] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceed-*

ings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '01, pages 149–160, New York, NY, USA, 2001. ACM.

[79] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability*, pages 46–66, New York, NY, USA, 2001. Springer-Verlag New York, Inc.

[80] Gnutella. `http://www.gnu.org/philosophy/gnutella.html`.

[81] David R. Choffnes and Fabián E. Bustamante. Taming the torrent: A practical approach to reducing cross-isp traffic in peer-to-peer systems. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, pages 363–374, New York, NY, USA, 2008. ACM.

[82] Haiyong Xie, Y. Richard Yang, Arvind Krishnamurthy, Yanbin Grace Liu, and Abraham Silberschatz. P4p: Provider portal for applications. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, SIGCOMM '08, pages 351–362, New York, NY, USA, 2008. ACM.

[83] Alok Shriram, Margaret Murray, Young Hyun, Nevil Brownlee, Andre Broido, Marina Fomenkov, and kc claffy. Comparison of public end-to-end bandwidth estimation tools on high-speed links. In *Proceedings of the 6th International Conference on Passive and Active Network Measurement*, PAM'05, pages 306–320, Berlin, Heidelberg, 2005. Springer-Verlag.

[84] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. *IEEE/ACM Trans. Netw.*, 11(4):537–549, August 2003.

[85] Vinay J. Ribeiro, Rudolf H. Riedi, Richard G. Baraniuk, Jiri Navratil, and Les Cottrell. pathchirp: Efficient available bandwidth estimation for network paths, 2003.

[86] Manish Jain and Constantinos Dovrolis. Ten fallacies and pitfalls on end-to-end available bandwidth estimation. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pages 272–277, New York, NY, USA, 2004. ACM.

[87] Joel Sommers, Paul Barford, and Walter Willinger. Laboratory-based calibration of available bandwidth estimation tools. *Microprocessors and Microsystems*, 31(4):222–235, 2007.

[88] Google TV. `http://www.google.com/tv/`.

[89] E.J. Candes and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, June 2010.

[90] V.K. Adhikari, S. Jain, Yingying Chen, and Zhi-Li Zhang. Vivisecting youtube: An active measurement study. In *INFOCOM, 2012 Proceedings IEEE*, pages 2521–2525, March 2012.

[91] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Scalable collaborative filtering approaches for large recommender systems. *J. Mach. Learn. Res.*, 10:623–656, June 2009.

[92] Netflix Prize. `http://www.netflixprize.com/`.

[93] Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 713–719, New York, NY, USA, 2005. ACM.

[94] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 556–562. MIT Press, 2001.

[95] T. G. Dietterich. *Ensemble learning*. MIT Press, Cambridge, MA, USA, 2nd edition, 2002.

[96] M. Wu. Collaborative filtering via ensembles of matrix factorizations. In *KDD Cup and Workshop 2007*, pages 43–47. Max-Planck-Gesellschaft, August 2007.

[97] James Surowiecki. *The Wisdom of Crowds*. Anchor, 2005.

[98] Jonathan Ledlie, Paul Gardner, and Margo Seltzer. Network coordinates in the wild. In *Proceedings of the 4th USENIX Conference on Networked Systems Design &#38; Implementation*, NSDI'07, pages 22–22, Berkeley, CA, USA, 2007. USENIX Association.

Narisu Tao

Curriculum Vitae

Education

2011–2015 **Ph.D.**, *Institute of Computer Science, Georg-August-Universität Göttingen*, Germany, **Expected**.

2007–2010 **M.E.**, *Department of Computer Science and Technology, Nanjing University*, China.

2003–2007 **B.S.**, *Department of Computer Science and Technology, Nanjing University*, China.

Research Interests

Big Data Analysis, Machine Learning, Probabilistic Modeling, Optimization, Computer Network, Inter-domain Routing, Compressive Sensing, Network Measurement, Wireless Sensor Network, Internet of Things.

Research Projects

01/2014–Present **MobileCloud - Linking Sino-European Research Institutions in the Mobile Cloud**, Georg-August-Universität Göttingen, Göttingen, Germany, Sponsored by EC FP7 IRSES Project (http://www.fp7-mobilecloud.eu/).

- **Extracting the spatio-temporal knowledge of the environment of interest through incomplete sensor readings in Wireless Sensor Network by using Probabilistic Graphical Models (Markov Random Field).**
- **Reconstructing the complete sensor readings out of incomplete data based on the inference on the probabilistic model.**

01/2013–Present **Simulation Center**, Georg-August-Universität Göttingen, Göttingen, Germany, Sponsored by Lower Saxony Ministry for Culture and Education, Germany (https://www.simzentrum.de/projekte/securing-communications-in-internet-of-things-iot-environments/).

- **Data analysis based on the Big Data (5GB/15min) from the Internet inter-domain (BGP) routing tables from the perspectives of several different backbones and locations around the Internet.**
- **Proposing new inference methods for Autonomous System level path in the Internet by exploiting the metrically tree-likeness of the Internet.**
- **Improving the locality property of overlay networks by inferring the end-to-end network connection quality with the Matrix Completion method (compressive sensing).**

09/2011–04/2012 **Microsoft Hawaii project**, GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN, Göttingen, Germany, http://research.microsoft.com/en-us/projects/hawaii/default.aspx.

- **Develpment of an smart phone application for real time train tickets sharing.**

10/2008–06/2010 **Multiagent Coalition Formation in Antagonistic Environment**, NANJING UNIVERSITY, Nanjing,China, (National Natural Science Foundation of China, No. 60875038.

03/2008–09/2008 **Applying Data Mining Technology into the Cell Phone Call Records Data Analysis**, NANJING UNIVERSITY, Nanjing, China, (High Tech Project of Jiangsu Province, P. R. China, No. BG2007038).

Positions

04/2012–Present **Research and Teaching Assistant**, COMPUTER NETWORK GROUP, GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN, Göttingen, Germany.

04/2015–05/2015 **Visiting Scholar**, KNOWLEDGE ENGINEERING GROUP, DEPARTMENT OF COMPUTER SCIENCE, TSINGHUA UNIVERSITY, Beijing, China.

09/2007–06/2010 **Research and Teaching Assistant**, STATE KEY LABORATORY FOR NOVEL SOFTWARE TECHNOLOGY AT NANJING UNIVERSITY, Nanjing, China.

03/2008–07/2008 **Internship**, PRODUCT DEVELOPMENT DEPARTMENT, JIANGSU FUYUE TECHNOLOGY CO., LTD, Nanjing, China.

Awards

2011–2015	German Academic Exchange Service Long-Term Scholarship
2008	Excellent Graduate Student of Nanjing University
2008	GUANGHUA Scholarship for Excellent Students

Skills

Basic	SCALA, APACHE SPARK(MLLIB), APACHE HADOOP, MapReduce, Python with scikit-learn, MySQL.
Intermediate	PYTHON, C#,C++,C,JAVA, LaTeX, OpenOffice, , Microsoft Windows, Microsoft Office, Windows Phone Application Development.
Advanced	Matlab/Octave. CVX optimizer, Gurobi optimizer, Linux

Online Course Certificates

08/2015	**Scalable Machine Learning** (UC Berkeley)(100%)
07/2015	**Autonomous Navigation for Flying Robots** (Technische Universität München)
07/2015	**Introduction to Big Data with Apache Spark** (UC Berkeley)(98%)
11/2014	**Functional Programming Principles in Scala** (Ècole Polytechnique Fédérale de Lausanne) (100%) (With Distinction).
06/2014	**Convex Optimization** (Stanford University) (With Distinction).

07/2013 **Probabilistic Graphical Models** (Stanford University) (With Distinction).

Presentations

2015 Paper presentation, IFIP Networking conference, Toulouse, France.

2013 Poster, Workshop on Statistical Issues in Compressive Sensing, Göttingen, Germany.

2013 Worshop on Nonsmooth Optimization in Machine Learning, Liège, Belgium.

2012 Demonstration, IEEE Pervasive Computing conference, Lugano, Switzerland.

Publications

2015 Narisu Tao, Xu Chen, Farshid Hassani Bijarbooneh, Wei Du, Edith Ngai, Xiaoming Fu, Probabilistic Model Enhanced Compressive Sensing for Environment Reconstruction in Sensor Networks. INFOCOM 2016, April 2016.(under submission)

2015 Narisu Tao, Xu Chen, Xiaoming Fu, AS Path Inference from Complex Network Perspective. IFIP Networking 2015, May 2015.

2014 Wei Du, Yongjun Liao, Narisu Tao, Pierre Geurts, Xiaoming Fu, Guy Leduc, Rating Network Paths for Locality-Aware Overlay Construction and Routing. IEEE/ACM Transactions on Networking, July 2014.

2013 Narisu Tao, Wei Du, Xiaoming Fu. Ratings of Network Paths Inference by Compressive Sensing, Workshop on Statistical Issues in Compressive Sensing, Göttingen, November 2013

2012 Narisu Tao, Konglin Zhu, Xiaoming Fu. Thumb: A Real-Time Resource Information Sharing Application over Mobile Phones. IEEE International Conference on Pervasive Computing and Communications (PerCom 2012), Demo Session, Marh 19-23, 2012.

Languages

Mongolian	**Native or bilingual proficiency**	
Mandarin	**Native or bilingual proficiency**	
English	**Full professional proficiency**	*TOEFL(103/120), GRE(1350/1600)*
German	**Limited working proficiency**	*Goethe Institute B1 certificate*

Interests

- Soccer
- Learning
- Kaggle
- Running
- Programming