DATA ANALYTICS FOR NETWORKED AND POSSIBLY PRIVATE SOURCES

A Thesis
Presented to
The Academic Faculty

by

Ting Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Computer Science

Georgia Institute of Technology
May 2011

DATA ANALYTICS FOR NETWORKED AND POSSIBLY PRIVATE SOURCES

Approved by:

Professor Ling Liu, Advisor
School of Computer Science
Georgia Institute of Technology

Professor Mustaque Ahamad
School of Computer Science
Georgia Institute of Technology

Professor Calton Pu
School of Computer Science
Georgia Institute of Technology

Professor Leo Mark
School of Computer Science
Georgia Institute of Technology

Professor James Caverlee
Department of Computer Science and Engineering
Texas A&M University

Date Approved: April 1st 2011

To Mom and Dad.

ACKNOWLEDGEMENTS

This dissertation would not have been possible without the generous help and encouragement of many individuals. First and foremost, I wish to thank my advisor and mentor, Prof. Ling Liu. I feel extremely fortunate to have crossed paths with her and to have had the opportunity to work with her so closely for many years. Her amazing creativity, energy and colorfulness, combined with knowledge, wisdom and empathy, have made every meeting with her a joy. Ling has cultivated me with her fantastic taste of research problems while allowing me the freedom of pursuing the problems I found most interesting; at the same time, in critical moments, she has always been available to provide sincere and determined opinions, helping me make the right decisions. The example she set up as a perfect researcher and teacher will undoubtedly continue to light my way in my future career.

Prof. Calton Pu has been another great mentor for me. His rapid ideas, unusual associations and amazing ability of finding connections between seemingly disparate problems have always been inspirational for me. I also owe a great debt to every member of DiSL research group for providing such a dynamic and cultivating environment. They have always been willing to listen to me mumbling a lot of half-baked ideas and to help organize my perpetually changing and chaotic thinking. Our in-depth discussions have provided invaluable insights into my own research and have strengthened many conceptual aspects of this dissertation.

During my graduate studies, I have had the great opportunities to work as intern for consecutive four summers at IBM T.J. Watson Research Center. I would like to express my sincere gratitude to my mentors Dr. Dakshi Agrawal and Dr. Mudhakar Srivatsa for letting me work on projects relevant to my thesis. I have had great fun

working with both of them and learning from them not only practical perspectives of research methodology but also high standards of research excellence.

I would also like to thank my committee members Prof. Mustaque Ahamad, Prof. Leo Mark and Prof. James Caverlee who have been very supportive throughout my graduate career and have shared with me many invaluable suggestions and comments on my work from their unique research expertise.

Finally, I would like to dedicate this work to my parents, whose love and patience have been my constant source of inspiration, without which this accomplishment would have been impossible. My success is also theirs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The past decade has witnessed an unprecedented growth in the complexity and variety of information, as partially driven by the advances in the following three areas: first, advanced sensing and monitoring technologies; second, pervasive network connectivity and ubiquitous computing platform; and third, social media and web 2.0 technologies. We are now facing data coming from multiple sources and featuring rich context information. For example, operators today have at their disposal myriad measures (e.g., NetFlow, SNMP, "syslog") collected from all routers of large-scale enterprise networks. In contrast, the existing analytical tools are lagging way behind this astonishing growth in the complexity and variety of data. For example, even though analyzing routing data holds the promise for exposing important network failures, this promise is largely unfulfilled due to the complex, noisy and voluminous nature of the data. The lack of general design models and formal methods to effectively weave context-rich information from multiple sources motivates this thesis.

More specifically, in this thesis we focus on two grand challenges facing system designers and operators. First, how to fuse information from multiple autonomous, yet correlated sources and to provide consistent views of underlying phenomena? Second, how to respect externally imposed constraints (privacy concerns in particular) without compromising the efficacy of analysis?

In the first scenario, the correlation (e.g., dependency) among the data sources is usually reflected in the collected data in the form of spatial and/or temporal relevance. For example, the sympotoms caused by a given network failure typically demonstrate significant patterns in terms of where and when they are observed. This motivates us

to design data analytical frameworks that can effectively incorporate the relationships of underlying data sources.

In the second scenario, due to the possible sensitive nature of the data, the data sources expect the entire process of data collection, processing and dissemination to provide sufficient privacy protection of their contributed data, even though the expected level of protection may vary from one source to another. This essentially raises the question of how to ensure privacy protection (e.g., via information sanitization), meanwhile guaranteeing the utility of the information for intended purposes.

To address the first challenge, we apply a general *correlation network* model to capture the relationships among data sources, and propose *Network-Aware Analysis* (NAA), a library of novel inference models, to capture (i) how the correlation of the underlying sources is reflected as the spatial and/or temporal relevance of the collected data, and (ii) how to track causality in the data caused by the dependency of the data sources. We have also developed a set of space-time efficient algorithms to address (i) how to correlate relevant data and (ii) how to forecast future data.

To address the second challenge, we further extend the concept of correlation network to encode the semantic (possibly virtual) dependencies and constraints among entities in question (e.g., medical records). We show through a set of concrete cases that correlation networks convey significant utility for intended applications, and meanwhile are often used as the steppingstone by adversaries to perform inference attacks. Using correlation networks as the pivot for analyzing privacy-utility trade-offs, we propose *Privacy-Aware Analysis* (PAA), a general design paradigm of constructing analytical solutions with theoretical backing for both privacy and utility.

The general design models and formal methods shown in this thesis can help improve existing data analytical systems by making them more capable of weaving local observations (from multiple sources) into globally consistent pictures, and more privacy-preserving with respect to sensitive information.

CHAPTER I

INTRODUCTION

Several tremendous trends are dominating today's computing environments: first, the emergence of advanced sensing and monitoring technologies that enable users to record and collect varied types of information in closed and open environments; second, the development of pervasive network connectivity and ubiquitous computing platforms that enable users to send and receive their intended information virtually at anytime from anywhere; and third, the popularity of social media and web 2.0 technologies that enable such information to be accessible in a virtually real-time manner. With these technology advancements, we are now facing astronomical size of data that come from multiple autonomous, yet correlated sources and feature rich context information, which spurs the need of building large-scale multi-source data analytical systems. Examples include network monitoring systems, mobile computing infrastructures and social networking platforms. Such systems hold the promise of supporting critical decision making by fusing information from different sources and providing consistent, multi-scale views of underlying phenomena. Today this promise remains largely unfulfilled, because building and operating massive multi-source analytical systems still face a multitude of challenges, including

- Designing system architectures that support storing and processing large-volume data from multiple sources.
- Developing models and algorithms that exploit information from all the sources.
- Deploying and tuning multi-source analytical systems in real applications.

1.1 Motivation

This thesis in particular focuses on addressing the data-centric challenges facing system designers and operators; that is, developing models and algorithms that fully use data from all the sources to understand the underlying phenomena.

In conventional data management and mining models, data are typically the first-class citizen and are usually handled independent of the context wherein they are generated or collected. For example, in transactional data mining with the objective of finding popular purchase patterns among the population, transaction records are typically processed irrespective of the corresponding customers.

In contrast, multi-source data analysis requires treating the rich context also as a first-class citizen and understanding data and context in an integrated manner. The context information influences data analysis in two aspects. First, the correlation among the data sources is usually reflected in the data in the form of spatial and/or temporal relevance, which makes it necessary to account for the relationships of the sources in order to interpret the data. Second, each autonomous source may have specific constraints (e.g., privacy concern), which poses the question of how to respect such constraints without compromising the efficacy of analysis. For example, with the emergency of social media, it is now feasible to collect transactional data associated with customer information (which is possibly sensitive) and their social relationships; incorporating such context information in analyzing transactional data leads to understanding social-community-specific purchase patterns.

The challenge however lies in the complexity of context information and the lack of general design models to effectively fuse context-rich information from multiple sources. Systems designers and operators in different domains have developed a plethora of solutions relying on varied methodologies and heuristics; nevertheless it is typically difficult if not impossible to transfer or reuse the solutions developed for one context to another (even similar) one. For instance, as will be discussed in

Chapter 6, the privacy protection mechanisms proposed for sensitive census data is inadequate to address the vulnerability of sensitive location data.

1.2 Contributions: Common Design Philosophy

This thesis aims at developing a set of general design models and formal methods that can help address the difficulties above. In particular we focus on two grand research challenges, namely, (i) how to exploit information from multiple autonomous, yet correlated sources and to provide consistent views of global phenomena? and further, (ii) how to respect externally imposed constraints (privacy concerns, in particular) without compromising the efficacy of analysis?

1.2.1 Network-Aware Analysis

To address the first challenge, we apply a general *correlation network* model to capture the relationships (e.g., dependency, reference, inheritance) among the data sources, and propose *Network-Aware Analysis* (NAA), a library of novel inference models, to capture (i) how the relationships of the underlying sources is reflected as the spatial and/or temporal relevance of the collected data, and (ii) how to track causality in the data caused by the dependency of the data sources. We have also developed a set of space-time efficient algorithms to correlate relevant data and to even forecast future data. A wide range of applications can be benefited from our work: it can help diagnose failure in communication networks in a scalable manner, spot anomalous email communication without in-depth inspection, perform effective recommendation for online resources, and improve efficiency of mobile paging operation.

1.2.2 Privacy-Aware Analysis

For the second challenge, regarding privacy as one concrete constraint imposed by data sources, we consider the question of how to ensure privacy protection (e.g., via information sanitization), meanwhile guaranteeing the utility of the information for

intended purposes. We extend the correlation network model to encode the semantic (possibly virtual) dependencies and constraints among entities in question (e.g., medical records, transaction patterns, mobile clients). We show through a set of domain-specific cases that correlation networks convey significant utility for intended applications, and meanwhile are often used as the steppingstone by adversaries to perform inference attacks. Using correlation network as the pivot for analyzing privacy-utility trade-offs, we propose *Privacy-Aware Analysis* (PAA), a general design paradigm for constructing analytical solutions with theoretical backing for both privacy and utility.

1.3 Contributions: Concrete Case Studies

The methodology followed by this thesis is "from practice, to theory and back to practice". We start with examining concrete contexts in representative applications, for network-aware analysis, ranging from communication network (diagnosing network faults) to social network (modeling social influence), and for privacy-aware analysis, ranging from sensitive data dissemination (medical data publishing) to sensitive data mining (transaction data mining) and (location)-sensitive data management (mobile location based service). We focus on extracting the process of coupling data and context analysis into unified theoretical models, and then develop and deploy practical solutions under the guidance of these design models. In the following we detail our contributions in each specific case.

1.3.1 Network-Aware Correlation Reasoning

Modern communication networks generate massive volume of operational event data, e.g., alarm, alert and metrics, which can be used by a network management system (NMS) to diagnose potential faults. We introduce a new class of indexable *network signatures* that encode temporal evolution of events generated by a network fault as well as topological relationships among the nodes where these events occur. We present an efficient learning algorithm to extract such signatures from noisy historical

event data, and with the help of novel space-time indexing structures, we show how to perform efficient, online signature matching. We also point out potential applications of such signatures for many different types of networks including social and information networks.

1.3.2 Network-Aware Causality Tracking

Social influence, the phenomenon that the actions of a user induce similar behaviors among his/her friends via social ties, exists prevailingly in socially networked systems. We identify the importance of understanding and modeling social influence at a *microscopic* scale (i.e., at the granularity of individual users, actions and timestamps), and propose a microscopic social-influence model wherein: users' actions are modeled as interactions between social network (formed by users) and object network (formed by targets of actions); actions of a user influence his/her friends in a dynamic, network-wise manner (dependent on both social and object networks). We develop a set of novel inference tools that enable to answer questions of the form: how may an occurred interaction trigger another? More importantly, when and where may a new interaction be observed?

1.3.3 Privacy-Aware Data Dissemination

A prominent problem in privacy-preserving data dissemination is to prevent adversaries from inferring the sensitive information of a victim, even in an approximate sense (e.g., the victim is inferred to associate with several diseases with high chance). We systematically study the problem of protecting proximity privacy in a data model-neutral manner, and develop a general proximity privacy theory centering around the correlation network of individuals' sensitive information, independent of concrete data models (e.g., categorical or numerical). This definition of proximity privacy subsumes a number of state-of-the-art privacy principles; efficient anonynimization algorithms

are developed that guarantee privacy protection while preserving utility for data analysis in a best-effort manner.

1.3.4 Privacy-Aware Data Mining

The state-of-the-art of privacy-preserving data mining research focuses on the question "can mining be run over sanitized datasets?"; our work instead is among the first that systematically answers the question "can mining over sanitized datasets be safe and risk free?". We expose the *output privacy* risk, namely that mining output (models/patterns) may leak sensitive information regarding input raw data (especially when the output is partially, continuously released, e.g., in the case of stream mining), even though the input data is sanitized. Centering around the correlation network of mining output, we propose a novel proactive countermeasure based on random perturbation, which provides theoretically guaranteed privacy protection, strikes balance among multiple dimensions of output utility, and requires fairly limited computational resources.

1.3.5 Privacy-Aware (Location) Data Management

Location-dependent data management tasks impose specific requirements regarding privacy protection, data utility and system performance. For example, how might the user obtain quality-guaranteed location-based services without exposing her private, exact position information? Meanwhile, how could the privacy protection mechanism incur no disincentive, e.g., excessive computation or communication cost, for any service providers or mobile users to participate in such a scheme? We study location privacy issues in real-life mobile services deployed over roads wherein mobiles' locations are correlated by underlying road networks. Constructed atop this correlation model, our solution explores the inherent trade-offs among location privacy, service execution performance, and quality of delivered service, and strikes a best possible balance among these factors.

1.4 Roadmap

The rest of this thesis will be organized as a series of chapters, each one dedicated to a specific topic within the scopes of network-aware and privacy-aware data analytics. Each chapter will be organized in the following format. The introduction part gives background information, introduces the problem setting, and summarizes our solution; the main part detailes our contributions; and the related work part surveys relevant literature.

More specifically, Chapter 2 and 3 are dedicated to network-aware data analytics. Chapter 2 presents META, the first proposal that leverages topological and temporal relevance of network monitoring data to perform scalable, yet robust correlation reasoning. Chapter 3 presents μSI, a novel inference framework that tracks and predicts social influence at a microscopic level.

Furthermore, Chapter 4, 5 and 6 are dedicated to privacy-aware data analytics. Chapter 4 presents xCOLOR, a general data anonymization model for sensitive data dissemination. Chapter 5 presents BUTTERFLY, a light-weight countermeasure that can effectively eliminate privacy breaches in (stream) mining output without explicitly detecting them, and meanwhile minimizing the loss of output accuracy. Chapter 6 presents xSTAR, a robust and scalable location privatization framework for mobile services deployed over road networks, which achieves the best balance among privacy protection, system overhead and service quality.

This thesis is concluded in Chapter 7.

1.5 Bibliographic Notes

Material from Chapter 2 appears in papers co-authored with Mudhakar Srivatsa, Dakshi Agrawal and Ling Liu [133, 134]. An early version of some material from Chapter 3 appears in a paper co-authored with Mudhakar Srivatsa, Dakshi Agrawal and Ling Liu [135]. Material from Chapter 4 appears in a paper co-authored with

Shicong Meng, Bhuvan Bamba, Ling Liu and Calton Pu [132] and a paper co-authored with Ling Liu [130]. Material of Chapter 5 appears in papers co-authored with Ling Liu [128, 131]. Material from Chapter 6 appears in a paper co-authored with Ling Liu [129].

CHAPTER II

META: NETWORK-AWARE CORRELATION REASONING

2.1 Introduction

The motif of networks is ubiquitous in our lives [19]. In its simplest form, a network can be modeled as a graph where the vertices of the graph represent *network entities* and the edges represent pairwise interactions between network entities. It turns out that the simple, local, pairwise interactions between network entities can give rise to complex, interesting global phenomena [19]. Models of such global phenomena as a function of local interactions is one of the key issues being investigated in the area of networks. In this work, we propose a new class of models suitable for learning, indexing, and diagnosing a wide range of network phenomena while focusing on faults in the communication networks to exemplify our techniques.

Large communication networks have hundreds of thousands of network entities, and they are typically managed by a centralized network management system (NMS) that collects (local) monitoring data from network entities to diagnose network faults. When a fault occurs at a network entity, it tends to influence the "neighboring" entities. Consequently, faults often results in a large burst of messages being sent to the NMS from the affected entities. Each message contains a timestamp, an identifier of the affected device, and a *type* that signifies an event at the affected device[1]. The goal of NMS is to correlate the events occurring in the network, identify the root-cause fault event(s), suppress dependent events, and discard routine operational events.

[1]Henceforth, we will use the words "message" and "event" interchangeably, while ignoring the semantic distinction that events occur at network entities resulting in messages that are received by the NMS for diagnosis.

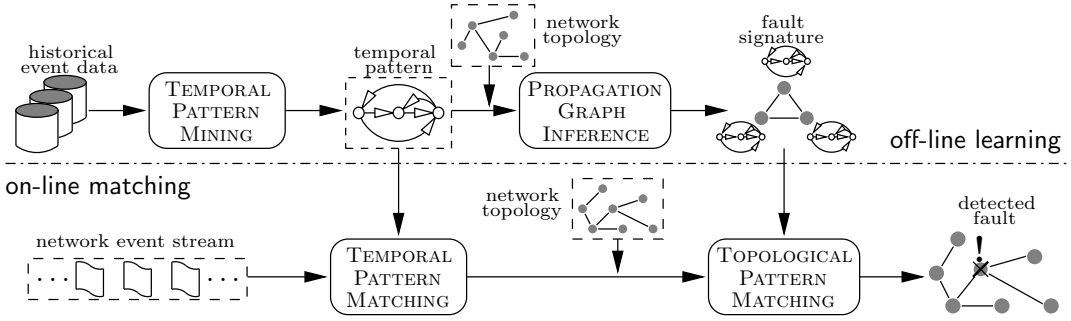


Figure 1: Illustration of the main architecture of META.

A key challenge faced by today's NMS is that of scalability. All widely deployed NMSes maintain a cache of "unresolved" events, and as each new event arrives, they use a rule-based mechanism to correlate the incoming event with all cached events to suppress dependent events and retain only the (unfixed) root-cause events in the cache. The resulting computation complexity is quadratic in network size since the number of unresolved events in the cache as well as event arrival rate is typically proportional to the network size.

Events triggered by a fault are typically generated by a small, constant-size subset[2] of nodes that are topologically related to the faulty node. Thus, each event arriving at the NMS only needs to be correlated with a small, constant-size subset of events in the cache, yielding a linear complexity of event correlation. Intuitively, achieving the linear complexity would require data structures that encode and exploit network topology. In this chapter, we propose a framework META (Monitoring network Events with Topology Assistance) that, to the best of our knowledge, is the first proposal to utilize topologically-aware event patterns to perform *scalable* network fault diagnosis.

The remainder of this chapter will be organized as follows. Section 2.2 formalizes the problem of network fault diagnosis; Section 2.3 and Section 2.4 describe in detail the offline learning and the online matching components of META, respectively

[2]The size of this subset depends on the degree distribution, etc., of the network and is independent of the size of the network.

(see Figure 1). An empirical analysis of our approach is presented in Section 2.5. Section 2.6 discusses related work.

2.2 Problem Formulation

This section introduces fundamental concepts and notations used in the chapter, and formalizes the problem of online network fault detection and localization.

Definition 1 (NETWORK). *A* **network** *is modeled as a graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ *with* $\mathcal{V}$ *representing the set of nodes, each corresponding to a network entity*[3]*, and* $\mathcal{E}$ *representing the set of edges over* $\mathcal{V}$*, each corresponding to a network link.*

Definition 2 (NMS/AGENT/SINK). *A* **network management system** (NMS) *consists of a set of monitoring* **agents** *and a* **sink**. *The agents are deployed on various network entities to collect monitoring data, and send them to the sink which is responsible for diagnosing potential faults.*

Definition 3 (NETWORK EVENT). *Each event* x *is a tuple of the form* $x = \langle e, v, t \rangle$*, where* e *represents the event type,* v *the network node generating the event, and* t *the timestamp of the event.*

In the rest of this chapter, we use e_x, v_x, and t_x to denote the type, entity, and timestamp of an event x, respectively.

Definition 4 (EVENT STREAM). *We model the event stream as a sequence of events,* $(x_1, \ldots, x_n)$*, where* x_n *is the most recent event. Real-time fault diagnosis focuses on events occurring within a time window of length* ω*, i.e., an event subsequence* $(x_i, x_{i+1}, \ldots, x_n)$*, with* $t_{x_n} - t_{x_i} \leq \omega$ *and* $t_{x_n} - t_{x_{i-1}} > \omega$*.*

Now we are ready to formally define the problem of real-time network fault detection and diagnosis:

[3] *Without ambiguity, in the following we use "entity" and "node" interchangeably.*

Problem Definition 1. *For each time-window, by analyzing the events occurring within the window and exploiting network topological information, detect potential faults (if any) and identify the fault types and failed entities.*

2.3 Offline Learning

In this section, we describe in detail our method of distilling the essential features of network faults manifested in network events and composing them into compact, indexable fault signatures. The signatures are designed to capture two critical aspects of network phenomena, namely, the *temporal evolution* of fault-triggered events, and the *topological relationship* of nodes associated with fault-triggered events. Furthermore, signatures are *universal* for a class of networks; that is, signatures learnt using data from one network instance can be used to diagnose faults in other network instances in the same class.

The feature learning process can be roughly divided into three phases. First, from the noisy historical data, we identify event subsets that correspond to network faults (with high probability) to train our feature extractor. Second, we extract temporal patterns embedded in the training data by extending the classical *expectation maximization* (EM) framework. Third, we combine the discovered temporal patterns with the topological information available to form compact fault signatures that are amenable to efficient indexing and matching. The details of the three phases are presented in Section 2.3.1, 2.3.2, and 2.3.3, respectively.

2.3.1 Preparation of Training Data

For the purposes of training fault signatures, it is necessary to separate events caused by faults and those triggered by regular network operations. In Meta, this step is achieved by applying three filters, *interval* filter, *support* filter, and *periodicity* filter to events generated by **each node**.

Interval Filter

Typically, the operational events caused by network faults occur in "bursts". The interval filter segments the event sequence generated by each node into a series of "event bursts".

Definition 5 (Event Burst)**.** *An* **event burst** *is an ordered sequence of events wherein two successive events are separated by no more than δ time units.*

As will be discussed in Section 2.5, an appropriate value for the parameter δ (typically a few milliseconds for communication networks) can be obtained by analyzing the distribution of inter-arrival time of events in a historical dataset. Since events in a burst occur within a very small time window, we arbitrarily define the timestamp of an event burst as the timestamp of its first event.

Support Filter

Given a set of event bursts $\mathcal{B}$ generated by a node, support filter treats each burst $s \in \mathcal{B}$ as a set and ignores the temporal ordering of events. Without ambiguity, we use s to denote both the event burst and its corresponding event set. We have the following definition.

Definition 6 (Support)**.** *Given a set of event bursts $\mathcal{B}$, the* **support** *of an event set s, $\mathsf{sup}(s)$, is the number of bursts in $\mathcal{B}$ with event sets identical to s.*

To make implementation resilient against noise, we consider two event sets s_i and s_j identical if their *Jaccard similarity coefficient*, $\frac{|s_i \cap s_j|}{|s_i \cup s_j|}$, is close to 1.

Support filter separates fault-triggered and regular event bursts at a node by exploiting differences in their support. Specifically, it selects the subset of bursts in $\mathcal{B}$ with support within a range $[\underline{\lambda}, \overline{\lambda}]$, since event sets with extremely low support are usually the noise component while over-frequent event sets typically correspond to regular network operations occurring at the node.

Periodicity Filter

The goal of the periodicity filter is to further reject event sets that occur periodically[4] since periodic event sets typically correspond to regular network operations, e.g., heartbeat messages.

Definition 7 (PERIODICITY)**.** *Let t_0^s, ..., t_k^s be the timestamps when the event set s occurs, and $d_i^s = t_i^s - t_{i-1}^s$ denote the interval between the $(i-1)$-th and i-th occurrences. The periodicity of s, $\mathsf{prd}(s)$, is defined as the relative standard deviation of intervals:*

$$\mathsf{prd}(s) = \frac{1}{\overline{d^s}} \cdot \sqrt{\frac{\sum_{i=1}^{k}(d_i^s - \overline{d^s})^2}{k-1}}$$

where $\overline{d^s} = \sum_{i=1}^{k} d_i^s/k$ denotes the average interval. We reject an event set s if $\mathsf{prd}(s)$ $< \gamma$.

2.3.2 Modeling of Event Bursts

Taking the event bursts $\mathcal{B}$ at a node (after the filtering of the first phase as input), this phase utilizes Markov chains to model $\mathcal{B}$ and produces a set of chains $\mathcal{MC}$ as the summarization of event bursts at a node. Markovian properties have been verified to be common in network operational events, e.g., [100, 111]. However, note that while we use Markov chains to model event bursts, we do not claim its optimality. It is worth emphasizing that our framework is flexible enough to support many other models that can be used to summarize events at a node.

We adopt a mixture model that contains multiple Markov chains $\mathcal{MC} = \{c_k\}_{k=1}^{K}$. We assume that each event burst is independently generated by one specific chain. Each chain $c \in \mathcal{MC}$ describes one type of sequential behaviors; thus, the mixture model is able to capture diverse behaviors embedded in event bursts.

Now we proceed to describing the structure of the mixture model. Without ambiguity, let $\mathcal{B}$ represent the collection of event bursts after initial filtering, and let $\Sigma =$

[4] A failed entity may also periodically generate failure events (e.g., syslog messages, ping fails, etc.); however, the NMS eliminates such duplicate failure events from the event stream using a standard process known as *de-duplication* before passing them to the diagnosis engine.

$\{e_i\}_{i=1}^{M}$ represent the event types that appear in $\mathcal{B}$. All chains in $\mathcal{MC}$ share the same structure: in a chain c_k, for each event $e_i \in \Sigma$, there is a corresponding state $o_{k,i}$; each state $o_{k,i}$ is associated with an initial probability $\theta^I_{k,i}$, indicating the probability that a burst starts with e_i; each state $o_{k,i}$ can transit to all other states $o_{k,j}$ (including to the state $o_{k,i}$ itself) with certain transition probability $\theta^T_{k,i,j}$; there is a special ending state $o_{k,0}$ for which all transition probabilities are zero. In the mixture model, each chain c_k is associated with a prior probability π_k, which satisfies $\sum_{k=1}^{K} \pi_k = 1$.

Let $\pi = \{\pi_k\}_{k=1}^{K}$, $\theta^I_k = \{\theta^I_{k,i}\}_{i=1}^{M}$, and $\theta^T_k = \{\theta^T_{k,i,j}\}_{i=1,j=0}^{M}$; the parameter space of this mixture model is represented as $\theta = (\pi, \{\theta^I_k, \theta^T_k\}_{k=1}^{K})$. Given an event burst $s = (e^s_1, e^s_2, \ldots, e^s_L)$, the likelihood that chain c_k generates s is given by:

$$\mathsf{like}(s|c_k, \theta) \propto \theta^I_{k,e^s_1} \cdot \left(\prod_{i=1}^{L-1} \theta^T_{k,e^s_i,e^s_{i+1}} \right) \cdot \theta^T_{k,e^s_L,0}$$

Meanwhile, the posterior probability that c_k generated s can be calculated as:

$$\mathsf{prob}(c_k|s, \theta) = \frac{\pi_k \cdot \mathsf{like}(s|c_k, \theta)}{\sum_{k'=1}^{K} \pi_{k'} \cdot \mathsf{like}(s|c_{k'}, \theta)}$$

The optimal setting of the parameters θ and the number of chains K remain to be determined. In the following, we first discuss how to determine θ that maximizes the posterior probability of the given set of event bursts. Let

$$\theta^* = \arg\max_{\theta} \mathsf{like}(\mathcal{B}|\theta) \cdot \mathsf{prob}(\theta)$$

where $\mathsf{prob}(\theta)$ is a prior distribution over θ and $\mathsf{like}(\mathcal{B}|\theta)$ represents the likelihood of observing the whole set of event bursts $\mathcal{B}$ under this model: $\prod_{s \in \mathcal{B}} \left(\sum_{k=1}^{K} \pi_k \cdot \mathsf{like}(s|c_k, \theta) \right)$. Unfortunately, no closed-form solutions exist for such maxima. Here, as sketched in Algorithm 1, an *expectation maximization* (EM) [36] algorithm can be used to iteratively search for the maxima (in the pseudo code below, Q denotes the objective function over the posterior distribution using current parameter estimation θ_{old}).

Now, we discuss how to set K. Essentially, by controlling the number of chains, K determines the complexity of the mixture model. Here, we apply the *Akaike's*

Algorithm 1: MODELING EVENT BURSTS

Input: event bursts $\mathcal{B}$, number of chains K
Output: parameter setting θ^*

initialize θ_{old};
while *not converged yet* **do**
  compute $Q(\theta, \theta_{old}) = \sum_{s \in \mathcal{B}} \sum_{k=1}^{K} \mathsf{prob}(c_k|s, \theta_{old}) \log[\pi_k \cdot \mathsf{like}(s|c_k, \theta)] + \log(\theta)$;
  compute $\theta_{new} = \arg\max_\theta Q(\theta, \theta_{old})$;
  set $\theta_{old} = \theta_{new}$;
return $\theta^* = \theta_{new}$;

information criterion [8, 100] to select K. Specifically, the information criterion of the mixture model is given by: $\mathsf{aic}(\theta) = 2|\theta| - 2\log[\mathsf{like}(\mathcal{B}|\theta)]$, where $|\theta|$ is the number of parameters to be estimated. The setting of K leading to a minimum $\mathsf{aic}(\theta)$ is considered as optimal.

2.3.3 Incorporation of Topological Dependency

A novel feature that significantly distinguishes META from existing solutions lies in its incorporation of network topology information in learning and matching faults. In this chapter, we consider the following set of relationships, {*selfing, neighboring, containing/contained, down/up-streaming, tunneling*}, with brief descriptions listed in Table 1. Note that the relationships *down/up-streaming* are referred from the perspective of the sink, i.e., u is at v's down-stream side if the route from the sink to u contains v. In the following, we use $\mathcal{R} = \{SE, NE, CN/CD, DS/US, TN\}$ to denote this set of topological relationships. Each relationship $r \in \mathcal{R}$ is associated with an inverse counterpart $\bar{r}$, e.g., 'down-streaming' to 'up-streaming', 'contained' to 'containing', etc. Given a node v, we refer to the set of network nodes with a specific relationship r to v as a *topo-set*, denoted by $\mathcal{N}_r(v)$.

Intuitively, we construct our fault signature based on the following two fundamental observations. (1) Typically, when a fault occurs at a root-cause node u, symptom events may be triggered in affected nodes that are topologically related to u. (2) The triggered event burst at an affected node v differs depending on the topological

Table 1: Topological relationships and descriptions.

relationship	**description**
selfing	u and itself
neighboring	u and v are directly connected
containing/contained	u/v contains v/u as a sub-component (e.g., a router and its interfaces)
down/up-streaming	u is at v's down/up-stream side (route from sink to u/v contains v/u)
tunneling	u is on a tunnel (a special type of network connection) with v as one end

relationship between u and v. For example, in an Internet Protocol (IP) network if v is a direct neighbor of u (*neighboring*), the failure of u may lead to the event burst of ("*OSPF Interface Down*", "*OSPF Neighbor Down*") at v; while if u is on a tunnel with v as one end (*tunneling*), the failure of u may cause the event burst of ("*Failed Connection Attempt*", "*Open Tunnel Failure*") at v. Therefore,

Definition 8 (Fault Signature)**.** *For a specific type of fault f, we define its signature $\mathsf{sig}(f)$ as a series of tuples $\langle c, r, \mathsf{prob}(c|f,r)\rangle$, where $c \in \mathcal{MC}$, $r \in \mathcal{R}$, and $\mathsf{prob}(c|f,r)$ is the probability of observing an event burst with temporal pattern c at an affected node that has the topological relationship r with the root-cause node where the fault f occurs.*

To learn fault signatures from historical data, we make the following assumptions: each event burst $s \in \mathcal{B}$ (observed at a node v) has been classified into a Markov chain c_v, and represented as a pair (v, c_v); the number of fault types $|\mathcal{F}|$ is known and all faults are reflected in the historical data; the time-window size ω is set as the maximum delay between observing the first and the last event bursts triggered by a single fault.

Algorithm 2 sketches our solution. (i) The event bursts $\mathcal{B}$ are first divided into subsets, each within a time-window less than ω, i.e., the event bursts in the same subset are possibly triggered by a single fault. (ii) In every subset, for each involved

node v, one identifies the topological relationship r_v^* that leads to the minimum non-empty intersection of all the topo-sets, i.e., the set of candidate causes. Note that the principle of *minimum explanation* is applied here. (iii) All the tuples $\langle v, r_v^*, c_v \rangle$ in a subset $\mathcal{B}_i$ are then used to compute a potential signature S_i (a $|\mathcal{R}| \times |\mathcal{MC}|$ matrix). (iv) A K-means (with $K = |\mathcal{F}|$) clustering algorithm is applied to the set $\{S_i\}$; the centers of the clusters are regarded as the signatures for the $|\mathcal{F}|$ faults.

Algorithm 2: LEARNING FAULT SIGNATURE

Input: event bursts $\mathcal{B}$, window-size ω, number of fault types $|\mathcal{F}|$
Output: $|\mathcal{F}|$ fault signatures

divide $\mathcal{B}$ into a set of subsets $\{\mathcal{B}_i\}$ according to ω;
for *each subset* $\mathcal{B}_i$ **do**
    `// apply the principle of minimum explanation`
    compute $\{r_v^*\}_{(v,c_v)\in\mathcal{B}_i} = \arg\min_{r_v} |\bigcap_{(v,c_v)\in\mathcal{B}_i} \mathcal{N}_{r_v}(v)|$;
    for *each* $r \in \mathcal{R}$ *and* $c \in \mathcal{MC}$ **do**
        compute $S_{i,r,c} = \frac{\sum_{(v,c_v)\in\mathcal{B}_i} \mathbf{1}(r_v^*=\bar{r}, c_v=c)}{\sum_{(v,c_v)\in\mathcal{B}_i} \mathbf{1}(r_v^*=\bar{r})}$;
apply K-means clustering to $\{S_i\}$ with $K = |\mathcal{F}|$;
set fault signatures as the cluster centers;

2.4 Online Matching

The online matching component of META attempts to detect and localize faults as follows: (1) the incoming events are aggregated into event bursts, and for each burst s occurring at an affect node v, the probability $\mathsf{prob}(c|s)$ is calculated for all $c \in \mathcal{MC}$; (2) topologically-aware fault signatures are used to compute the probability $\mathsf{prob}(f|\bar{r}, v \leftarrow s)$ that the faulty node incurred the fault f and has a topological relationship r to the affected node v. If this probability is greater than a certain threshold, then $\langle f, v, \bar{r} \rangle$ is termed an *evidence* that points to the set of all nodes with relationship r to v as the set containing the faulty node; (3) all collected evidences within a short time window are used to narrow down the set of nodes that include the faulty node.

We need four main data structures to accomplish the online matching: a buffer for aggregating incoming events into event bursts and to compute probability of a

temporal model generating an event burst; an index of fault signatures to compute evidences; and an index of network topological dependency and a signature matching tree to enable efficient fault localization. Due to the space constraint, we will only describe the latter three data structures.

2.4.1 Indexing Fault Signature

To support efficient model-to-fault lookup, we devise an inverted fault signature structure $\mathcal{I}_s$ which maintains the association between models and possible faults. Recall that the signature of a fault f is a series of tuples of the form $\{\langle c, r, \mathsf{prob}(c|f,r)\rangle\}_{c\in\mathcal{MC}}$, where c and r represent a chain and a topological relationship, respectively, and $\mathsf{prob}(c|f,r)$ is the probability of observing c at a node with topological relationship r to the faulty node. Corresponding to each signature, we create an entry in $\mathcal{I}_s$: $\{\langle f, \bar{r}, \mathsf{prob}(f|\bar{r},c)\rangle\}_{c\in\mathcal{MC}}$, where $\bar{r}$ is the inverse relationship of r, and $\mathsf{prob}(f|\bar{r},c)$ is the posterior probability that f occurs at a node with topological relationship $\bar{r}$ to a given node observing c. Its computation is given by:

$$\mathsf{prob}(f|\bar{r},c) = \frac{\mathsf{prob}(c|f,r)\cdot\mathsf{prob}(f)}{\sum_{f'\in\mathcal{F}}\mathsf{prob}(c|f',r)\cdot\mathsf{prob}(f')}$$

where the prior probability of the occurrence of f, $\mathsf{prob}(f)$, can be derived from the overall statistics of network faults.

Now, the posterior probability that a fault f occurs at certain node with relationship r to a node v which observes an event burst s can be calculated as follows:

$$\mathsf{prob}(f|\bar{r}, v\leftarrow s) = \sum_{c\in\mathcal{MC}}\mathsf{prob}(f|\bar{r},c)\cdot\mathsf{prob}(c|s)$$

For each f and v (with event burst s), we select the set of topological relationships $\mathcal{R}_v$ that satisfies $\mathsf{prob}(f|\bar{r}, v\leftarrow s) \geq \kappa$ $(\bar{r}\in\mathcal{R}_v)$. We term such a triple $\langle f, v, \mathcal{R}_v\rangle$ as an *evidence*.

2.4.2 Indexing Network Topology

While the incorporation of network topological information significantly boosts the precision of fault diagnosis, such improvement incurs extra computation cost in terms of 1) storing the topological information, and 2) correlating event bursts according to their underlying topological relationships. Here, we introduce novel space-efficient indexing structures for topological correlation.

As will be shown Section 2.4.3, a key operation heavily involved in the fault localization is computing the intersection of two topo-sets, e.g., joining the down-streaming neighbors of one node and the direct neighbors of another; therefore, for each indexing structure, we are particularly interested in analyzing its storage demand and the cost of retrieving (constructing) a topo-set from it. Here, we assume network configurations to be static, and consider incremental maintenance of indices for evolving networks as one direction for our further research. Due to space limitations, we focus our discussion on building indices for up/down-streaming and tunneling relationships.

Up-Streaming/Down-Streaming

A naïve solution that stores the up/down-streaming nei-ghbors for each network entity, results in $O(1)$ retrieval cost and totally $O(|\mathcal{V}|^2)$ storage cost. We construct our indexing structure based on the following two observations: (1) the shortest path routes from the sink to all the nodes form a spanning tree rooted at the sink, i.e., a tree cover of $\mathcal{G}$ [4]; (2) the diameter ϕ of a typical management domain (as observed in four large enterprise networks) is about 3-7 hops. Therefore, in this setting, the set of up-streaming neighbors (utmost ϕ) of a node can be directly cached. We then traverse the routing tree in a level-order (breadth-first) assigning each node a traversal-order number. The down-streaming neighbors of a given node u can be summarized as ϕ intervals, $\{[l_i, r_i]\}_{i=1}^{\phi}$, where l_i (r_i) denotes the order number of its left-most (right-most) descendent on the ith level below u (see example in Figure 2).

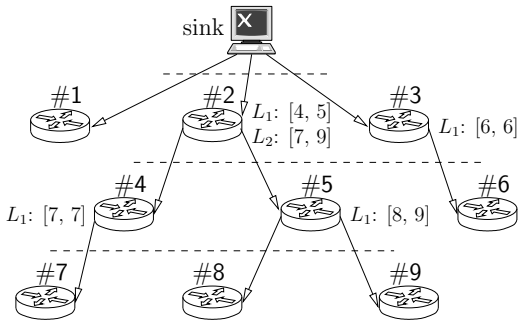


Figure 2: Index of up/down-streaming neighbors.

Clearly, this indexing structure requires $O(|\phi \cdot \mathcal{V}|)$ space; maintaining a mapping (sorted on traversal-order number) that projects traversal-order numbers to node-identifiers, this scheme achieves retrieval cost of $O(\phi)$, since the neighbors on the same level can be retrieved in one consecutive chunk.

Tunneling

For tunneling relationship, we are interested in retrieving the set of nodes on tunnels with a given node u as one end, or, reformulated as: *given two nodes u and v, what set of nodes are on the tunnel (if any) connecting u and v?*

Without loss of generality, we assume that all the tunnels follow approximately shortest paths (e.g., OSPF and IGP routing [68]); hence, the problem is cast as indexing a set of shortest paths. Our solution is constructed atop the notion of hop cover of a collection of paths [29].

Definition 9 (Hop/Hop Cover)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph and $\mathcal{P}$ be the set of shortest paths we intend to index. A* **hop** *is a tuple (p, u), where p is a shortest path with u as one end. A collection of hops $\mathcal{H}$ is said to be a* **hop cover** *of $\mathcal{P}$ if for any $P \in \mathcal{P}$, there is a subset of $\mathcal{H}$ such that P is a concatenation of these hops.*

We construct the collection of hops as follows: starting from an arbitrary path $P_1 \in \mathcal{P}$, we incrementally add in paths $P_2, P_3, \ldots$ from $\mathcal{P}$; the intersected segments between P_i and previous ones $P_1, \ldots, P_{i-1}$ break paths into disjoint hops, or divide

existing hops into smaller ones; we collect the set of hops after inserting all paths of $\mathcal{P}$ as $\mathcal{H}$. A moment of reflection shows that (1) two paths can have at most one intersected segment and (2) the set of hops are invariant of the insertion order of paths.

Within this setting, the space-time tradeoff is achieved by caching a subset of $\mathcal{H}$ and leaving uncached hops to online computation; therefore, we are interested in selecting the optimal subset $\mathcal{H}^* \subseteq \mathcal{H}$ that leads to the minimum overall cost as follows. For a hop h, let $\mathsf{len}(h)$ be its length and $\mathsf{sup}(h)$ be the number of paths in $\mathcal{P}$ that contain h as a component. For simplicity, we model storage cost $\mathsf{cost}_{space}(h) = \alpha \cdot \mathsf{len}(h)$ and computation cost $\mathsf{cost}_{time}(h) = \beta \cdot \mathsf{len}(h)$. Assuming that all the paths are queries with equivalent frequency, the overall cost of caching a subset $\mathcal{H}'$ of $\mathcal{H}$ can be modeled as: $\mathsf{cost}(\mathcal{H}') = \sum_{h \in \mathcal{H}'} \mathsf{cost}_{space}(h) + \sum_{h \in \mathcal{H} \setminus \mathcal{H}'} \mathsf{sup}(h) \cdot \mathsf{cost}_{time}(h)$; and the optimal subset $\mathcal{H}^*$ leads to the minimum overall cost: $\mathcal{H}^* = \arg\min_{\mathcal{H}'} \mathsf{cost}(\mathcal{H}')$. Clearly, in this model, a hop h should be cached if and only if its storage cost exceeds its computation cost with respect to all the paths, formally: $\mathsf{cost}_{space}(h) > \mathsf{sup}(h) \cdot \mathsf{cost}_{time}(h)$.

2.4.3 Correlating Relevant Evidences

With the help of the network topology index, in each evidence $\langle f, v, \mathcal{R}_v \rangle$, $(v, \mathcal{R}_v)$ can be replaced with the corresponding topo-sets $\bigcup_{r \in \mathcal{R}_v} \mathcal{N}_r(v)$ ($\mathcal{N}_{\mathcal{R}_v}(v)$ for short). Two evidences $\langle f, \mathcal{N}_{\mathcal{R}_u}(u) \rangle$ and $\langle f', \mathcal{N}_{\mathcal{R}_v}(v) \rangle$ are considered *relevant* if (1) $f = f'$, (2) $\mathcal{N}_{\mathcal{R}_u}(u) \cap \mathcal{N}_{\mathcal{R}_v}(v) \neq \varnothing$, and (3) they are within a time-window of size ω. This concept can be generalized to multiple evidences.

While it is straightforward to check conditions (1) and (3), computing the intersection of $\mathcal{N}_{\mathcal{R}_u}(u)$ and $\mathcal{N}_{\mathcal{R}_v}(v)$ is expensive: even if both sets are stored in a hash-table, the complexity is $O(\min\{|\mathcal{N}_{\mathcal{R}_u}(u)|, |\mathcal{N}_{\mathcal{R}_v}(v)|\})$. Moreover, following the naïve pairwise comparison paradigm, each incoming evidence is compared with all existing ones to detect relevance, and thus scales poorly with the network event rate.

We devise a novel structure, signature matching tree $\mathcal{T}_s$, which enables efficient correlation of relevant evidences. Our design follows the one-pass clustering philosophy, [58, 150], which endows $\mathcal{T}_s$ with high throughput and scalability.

Basic Structures and Operations

$\mathcal{T}_s$ is a hierarchical structure, with the highest level containing $|\mathcal{F}|$ buckets, each corresponding to one fault $f \in \mathcal{F}$. Within each bucket is a height-balanced tree $\mathcal{T}_s^f$, into which evidences of the form $\langle f, \mathcal{N}_{\mathcal{R}_v}(v)\rangle$ are inserted. Each leaf of $\mathcal{T}_s^f$ corresponds to a cluster of relevant evidences; each non-leaf represents the union of all the clusters in its subtree.

For each leaf (cluster) C containing a set of evidences, we maintain the intersection of their topo-sets, called its aggregation, $\rho(C) = \bigcap_{\langle f, \mathcal{N}_{\mathcal{R}_v}(v)\rangle \in C} \mathcal{N}_{\mathcal{R}_v}(v)$. For each non-leaf (super cluster) SC, we maintain the union of the aggregations of the clusters in its subtree, $\rho(SC) = \bigcup_{C \in SC} \rho(C)$.

The signature matching tree supports two basic operations, insertion and deletion. In an insertion operation, a newly coming evidence $\langle f, \mathcal{N}_{\mathcal{R}_v}(v)\rangle$ recursively descends down $\mathcal{T}_s^f$ by testing $\mathcal{N}_{\mathcal{R}_v}(v) \cap \rho(SC)$ for each non-leaf SC encountered, until being clustered into an appropriate leaf that can absorb it; if no such leaf exists, a new one is created which solely contains this evidence; it then updates the aggregations of the nodes on the path from the leaf to the root of $\mathcal{T}_s^f$. Those evidences with timestamps out of the current time-window are considered as expired. In a deletion operation, expired evidences are removed from the tree, and the aggregations of the nodes on the paths from the affected leaves to the root are updated in a bottom-up manner.

Optimizations

Two expensive operations involved in the signature matching tree are (1) testing the intersection of the topo-sets of an evidence and the aggregation of a (non-)leaf, and (2) updating the aggregations of the affected (non-)leaves when deleting expired

evidences. Here, we introduce two-folded optimizations to ameliorate these two operations.

Filtering-and-Refining

Instead of performing direct comparison of two sets, we follow a filtering-then-refining paradigm: in the filtering phase, we perform fast check to determine if the intersection is non-empty, which may contain false positive results, but no false negative ones; in the refining phase, we make the real comparison. To this end, for each evidence $\langle f, \mathcal{N}_{\mathcal{R}_v}(v)\rangle$, we maintain its bloom filter encoding, $\mathsf{bf}[\mathcal{N}_{\mathcal{R}_v}(v)]$; for each leaf C, we maintain the bloom filter encoding of its aggregation, $\mathsf{bf}[\rho(C)]$; while for each non-leaf SC, a counting filter [43] encoding (to support efficient update) of its aggregation, $\mathsf{cf}[\rho(SC)]$, is maintained. Therefore, the intersection of $\mathcal{N}_{\mathcal{R}_v}(v)$ and $\rho(SC)$ (or $\rho(C)$) can be easily pre-tested using $\mathsf{bf}[\mathcal{N}_{\mathcal{R}_v}(v)] \cap \mathsf{cf}[\rho(SC)]$ (or $\mathsf{bf}[\mathcal{N}_{\mathcal{R}_v}(v)] \cap \mathsf{bf}[\rho(C)]$).

Slotted-Aggregations

To ameliorate the impact of frequent deletions of expired evidences over updating the aggregations of (non-)leaves, we introduce the slotted-aggregates mechanism [7]. Assuming that the sliding window size is ω, a slot cache maintains the aggregations in m slots, the i^{th} slot corresponding to the evidences with timestamp falling in the i^{th} sub-window of size ω/m time units. Now, the deletion of expired evidence affects at most one slot, and the aggregations in all remaining slots can be reused.

2.5 Empirical Analysis

This section presents an empirical evaluation of META by using it in the context of communication networks. The experiments are specifically designed to center around the following metrics: (1) the efficacy of the signature model in capturing real network-faults, (2) the effectiveness of online matching in detecting and localizing network faults, and (3) its space and time complexity. We start with describing the datasets and the setup of the experiments.

2.5.1 Experimental Setting

We used two datasets collected from real-life communication networks to evaluate the learning and matching components of META. The first dataset is an archive of SNMP (Simple Network Management Protocol) trap messages collected from a large enterprise network (7 ASes, 32 IGP networks, 871 subnets, 1,268 VPN tunnels, 2,068 main nodes, 18,747 interfaces and 192,000 entities) over several days in 2007; this dataset is used to extract fault signatures. Event attributes of interest to us are listed in Table 2. The second dataset is a European backbone network consisting of 2,383 network nodes (spans 7 countries, 11 ASes and over 100,000 entities). We generate a synthetic event stream for this network (with tuneable failure rate) to quantify the efficacy and scalability of the online matching component.

Table 2: Format of the network event data.

Attribute	**Description**
IPaddress	address where the agent resides
PeerIPaddress	address of the master peer (if any)
Event-Count	sequence number of the event
specific-trap	enterprise specific snmp trap type
RawCaptureTimeStamp	time-stamp of the trap message

A majority of the algorithms are implemented using Java. All the experiments are conducted on a Linux workstation running 1.6GHz Pentium IV and 1G memory.

2.5.2 Experimental Results

Offline Learning Component

In this set of experiments, we studied the effectiveness of our methodology in all three phases of the learning component of META: preparation of training data, modeling of event bursts, and incorporation of topological information.

Preparing Training Data

Table 3: Specific-trap codes and descriptions.

specific-trap	description
28	linkModeSniffingTrap
79	bsnAPRegulatoryDomainMismatch
104	metro1500TDMLocModuleData1
124	metro1500TDMLocModuleClockFail

The first set of experiments studied the distribution of timestamp intervals, frequency, and periodicity of event sets to demonstrate the effectiveness of *interval* filter, *support* filter, and *periodicity* filter, respectively.

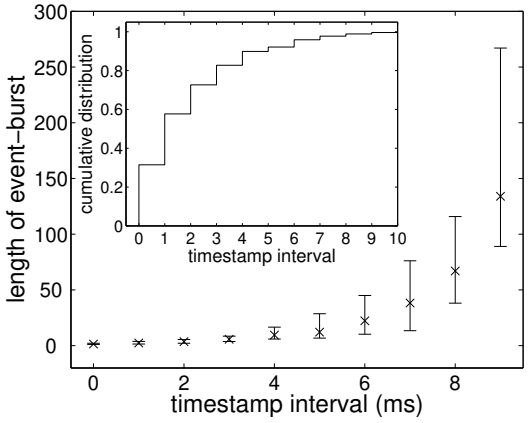


Figure 3: Length of event bursts with respect to the maximum timestamp interval.

Figure 3 illustrates the impact of interval size δ on the average length of event bursts. It is clear that the average length increases significantly as the interval grows, e.g., 50 events for $\delta = 8$ms; meanwhile, a wider interval also enlarges the length deviation of the event bursts. We are interested in an optimal setting of δ that filters spurious event bursts. In our implementation, we used the following heuristic: in the cumulative distribution function (CDF) of intervals, find the interval value with the largest derivate, i.e., the one resulting in the most significant change of the number of event bursts. For example, in the CDF plotted in Figure 3, we selected $\delta = 1$ms as the optimal setting.

The normalized histogram of event sets with respect to support (in log scale) is depicted in Figure 4, which approximately follows a *power law* distribution. It is

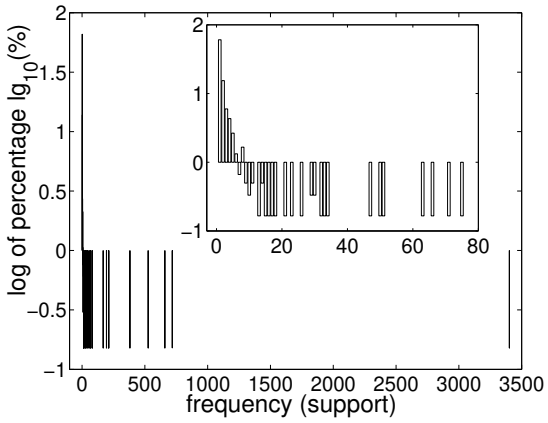


Figure 4: Distribution of frequencies of event sets.

observe that more than 60% event sets have fairly low support, e.g., below 5, which, as we confirmed by examining the definition of traps, are mainly caused by infrequent network operations, e.g., the event set {3} represents "the cisco NetReg server has started on the host from which this notification is sent", or certain non-recurrent faults, which are of modest interest for our purposes of leveraging existing diagnosis efforts. Meanwhile, the event sets with significantly higher support than others are typically due to regular network operations, e.g., the event set {102, 104} which appears with support 348 indicates "data from the remote side is available for the TDM channel".

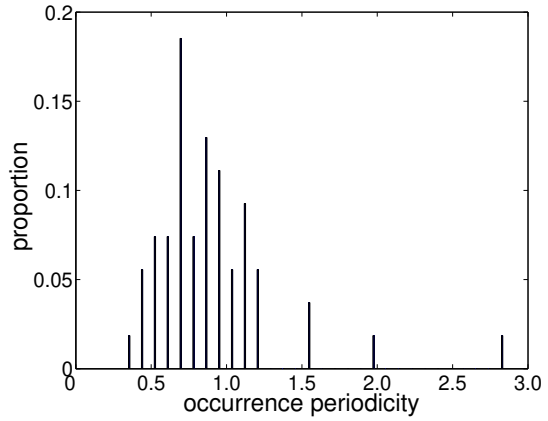


Figure 5: Normalized histogram of the periodicity of event sets.

The distribution of the periodicity of event sets is illustrated in Figure 5. Observe

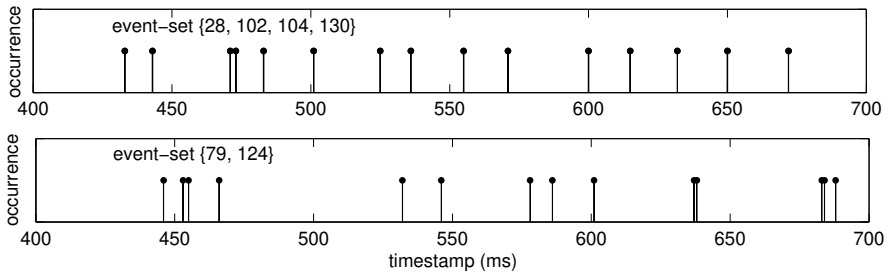


Figure 6: Occurrences of two sample event sets.

that most of the event sets demonstrate low deviation of occurrence intervals, i.e., they are resulted from normal network operations. We randomly selected two event sets {28, 102, 104, 130} and {79, 124} with periodicity 0.43, and 1.09 (lower periodicity ⇒ more regular), respectively, and examined their implications. Figure 6 compares their occurrences. From the descriptions of the traps as shown in Table 3, it is confirmed that the event set {79, 124} indicates potential network faults, while the event set {28, 102, 104, 130} is caused by regular network operations, e.g., link mode sniffing.

Modeling of Event Bursts

We verified the Markovian assumption on event bursts using the *event burst length histogram* metric. More specifically, by running Monte Carlo simulation, we derived the histogram of event burst length from the learned Markov model, and compared it against that extracted from the real data.

The upper plot of Figure 7 illustrates the comparison of these two histograms (normalization is applied). It is clear that the distribution of the model-generated data fits that of the underlying data fairly tightly. Furthermore, we analyzed the distribution of individual events for real data and model generated data, respectively. As shown in the lower plot of Figure 7, these two distributions demonstrate strong consistency, which empirically proves that our learning model can capture the essential features of the real data.

Incorporation of Topological Dependency

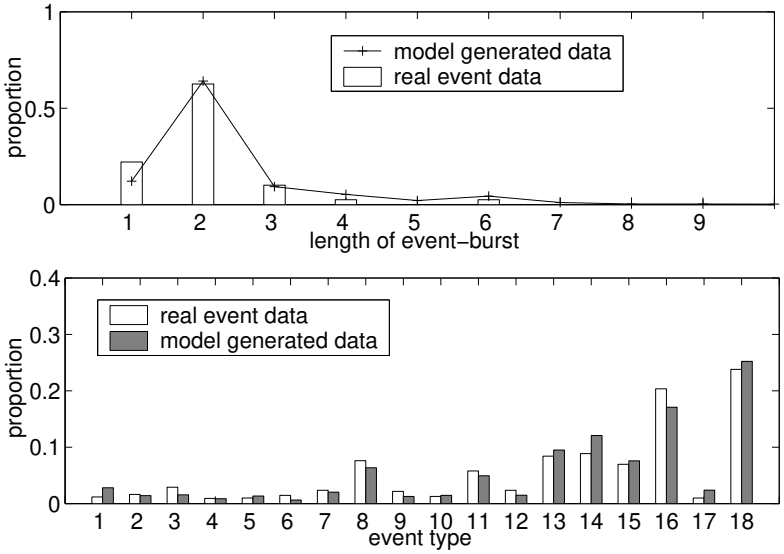


Figure 7: Histograms of lengths of event bursts and individual events in real event data and model generated data.

Now, we proceed to verifying the imperative need of incorporating topologically correlated nodes in detecting and localizing network faults. Figure 8 illustrates the fractions of fault-triggered events reported at network nodes in different categories: the fault node itself (SE), and nodes with specific relationships (neighboring - NE, down-streaming - DS, up-streaming - US, and tunneling - TN) to the fault one using event data collected from real enterprise networks. Note that Enterprise 3 is a small-scale network where no VPN tunnels are deployed. Observe that the fault node itself reports only 18-29% of overall events, while those topologically correlated nodes take up to an overwhelming 71-82%.

Online Matching Component

In evaluating the online matching component of META, we first generate synthetic event data for a European backbone network using the features of the real-life event data extracted in the previous phase, including 1) the temporal models for generating event bursts as symptoms of network faults, 2) the topological correlations for selecting the network entities where the symptoms will be observed, 3) the frequencies of

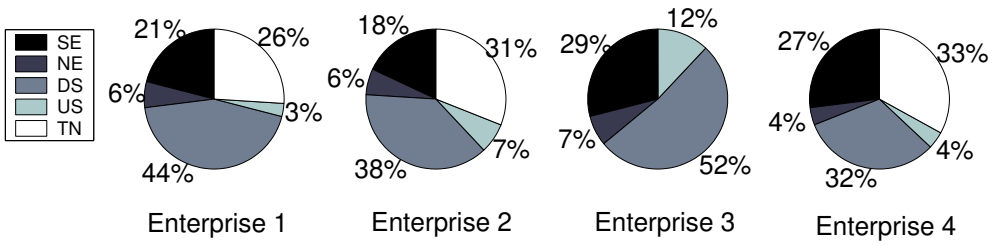


Figure 8: Fractions of fault-triggered events reported at fault nodes (SE) and nodes with specific relationships to fault ones (NE, DS, US, TN) in four real enterprise networks. The network sizes are listed as follows. Enterprise 1: 2514 nodes, Enterprise 2: 3200 nodes, Enterprise 3: 141 nodes, Enterprise 4: 12,444 nodes.

Table 4: Setting of parameters for synthesizing network event data.

Parameter	**Value**
number of fault types	50
number of event types	42
significant chains	16
fault occurrence rate	0.02 ~ 0.1
candidate size	3 ~ 7

individual events (type), and 4) the frequencies and periodicities of event sets. By controlling failure rates, we simulate network environments under both well-regulated and unstable conditions. The setting of major parameters is listed in Table 4.

Accuracy of Fault Diagnosis

This set of experiments evaluated the effectiveness of the Meta framework in detecting potential network faults by analyzing streaming network event data. We aim at achieving fault detection and localization in a unified framework; therefore, we consider that a fault is successfully diagnosed only if the fault type is correctly determined and the fault node is localized with sufficient accuracy (in our implementation, we require that for every detected fault, the system suggest no more than D potential fault nodes (candidate size)). We refer to a successfully diagnosed fault as a *hit*. Viewing fault diagnosis as an information retrieval task, we measure the exactness

and completeness of fault diagnosis using *precision* and *recall*. Formally,

$$\text{precision} = \frac{\#\text{ hits}}{\#\text{ reported faults}} \quad \text{recall} = \frac{\#\text{ hits}}{\#\text{ actual faults}}$$

Moreover, we measured the impact of topological information on fault diagnosis. We construct a baseline approach that is agnostic to topology information as follows: it has access to complete knowledge associating network faults with the observed symptoms, but has no possession of topological information. Given a symptom, the baseline approach attempts to identify the minimum set of faults that may trigger these symptoms; we regard it as a hit if the fault type suggested by the baseline approach is correct.

We measured the performance of Meta and the baseline approach under varying configuration of fault occurrence rate and topological correlations. The fault occurrence rate indicates the frequency of network faults (resulting in abnormal event bursts) relative to regular network operations (leading to normal event bursts); the configuration of topological correlation refers to the fractions of fault-triggered events observed at nodes with various topological relationships to the faulty node, e.g., SE, NE, DS, US, etc. Here, we adopt four different configurations as observed in real enterprise networks (shown in Figure 8).

Figure 9 compares the accuracy of fault diagnosis by Meta and the baseline approach. We make the following observations: (1) Meta achieves steady precision and recall scores under all the four configurations; the accuracy of the baseline approach is strongly correlated with the fraction of SE (fault node itself) events – even under configuration 3 (29% SE events), its recall (0.6) is substantially lower that of Meta (0.85). (2) The recall of Meta increases significantly as we increase D, for example, under configuration 2, the recall score of Meta grows from 0.65 to 0.84 as the candidate size D varies from 3 to 7. (3) The precision of Meta also increases as the candidate size D grows, which at the first glance may seem to contradict the

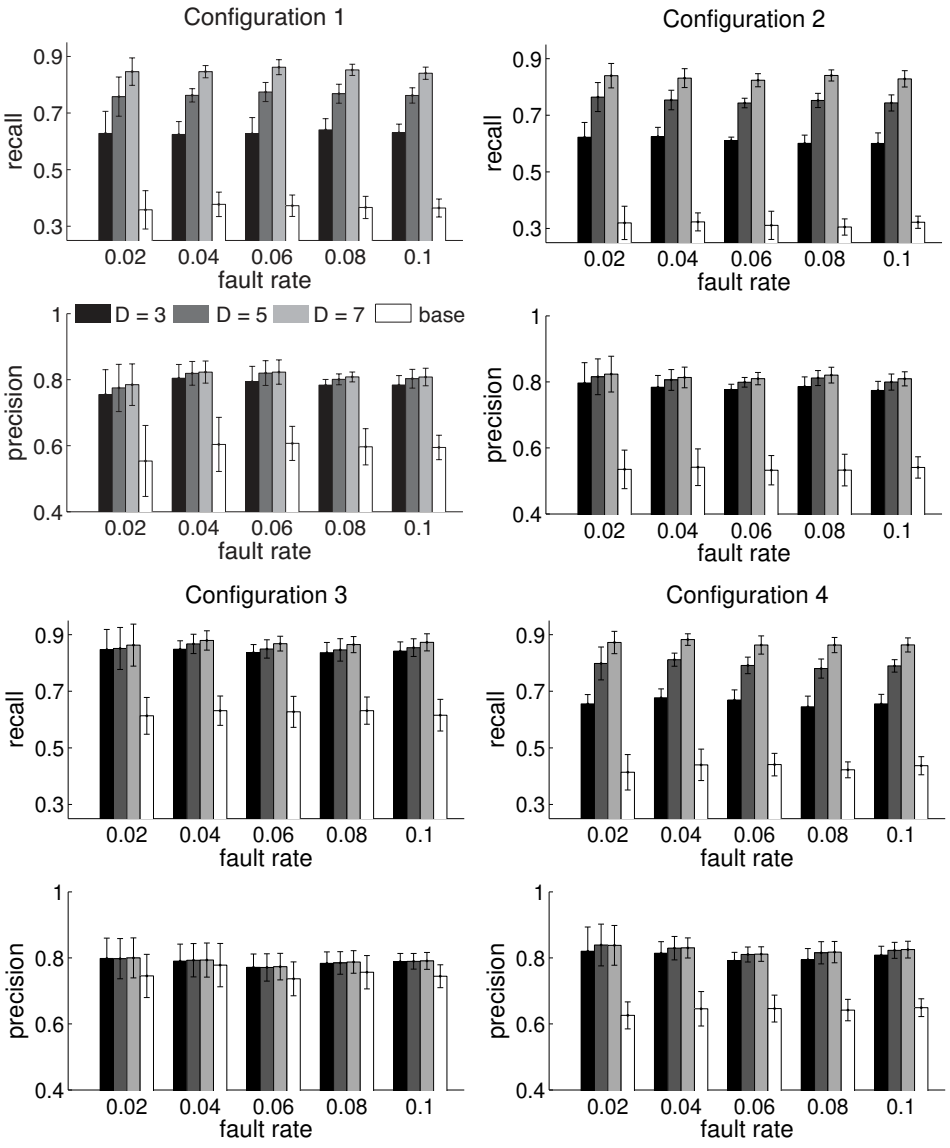


Figure 9: Accuracy of fault diagnosis by META (with candidate size $D = 3$, 5, and 7, respectively) and baseline approach with respect to fault occurrence rate and configuration of topological correlations.

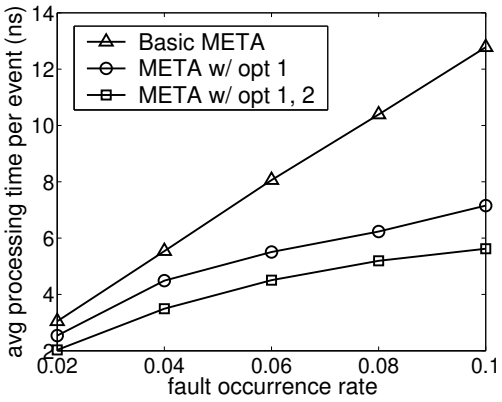


Figure 10: Average processing time (ns) per event with respect to fault occurrence rate, where opt_1 and opt_2 refer to the filtering-then-refining and the slotted aggregation strategies, respectively.

inverse relationship between precision and recall typically observed in information retrieval systems; however, this can be explained by the fact that a larger D essentially provides more leeway in identifying the fault node.

Efficiency of Execution

This set of experiments are designed to measure the scalability of the fault diagnosis in META. Specifically, we evaluate the average processing time of each incoming event, under varying condition of fault occurrence rate, with and without the multi-folded optimizations introduced in Section 2.4.3. Here, the fault occurrence rate refers to the fraction of event bursts caused by network faults.

Figure 10 shows the average processing time per event by three variants of META (the basic version, the one with the filtering-then-refining strategy, and the one with both optimization strategies) as the fault occurrence rate varies. We can obtain the following observations. (i) The processing cost of the basic META grows approximately linearly with the fault rate. (ii) The multi-folded optimizations significantly boost system performance, and the processing cost of both optimized variants of META manifest sub-linear growth rate with respect to the fault rate. (iii) The cost saving achieved by the optimization strategies demonstrates an increasing trend as

the fault occurrence rate grows, which can be explained by the fact that a higher fault rate results in a greater number of evidences being fed to the diagnosis engine, thus resulting in superior performance gains.

2.6 Related Work

For anomaly detection, a plethora of work has been done that uses analysis of low-level metric data, e.g., traffic or routing data, for anomaly detection. For example, in [45], BGP update messages are clustered along three dimensions, time, prefix, and views to detect network disruptions; in [82, 64], multivariate analysis is applied to model normal network traffic and detect deviations; in [148], a wavelet-based clustering algorithm is used to detect abnormal routing behavior. Nevertheless, targeting static analysis of low level metric data, these techniques are not suitable for real-time analysis of high-level event stream. Meanwhile, anomaly detection using historical data has also been an important topic for computing systems in general [30, 146], whose application to networked systems, however, is not clear.

Another line of research is specifically dedicated to fault localization from a set of observations or symptoms (detailed survey in [116]). The existing solutions can be categorized roughly as expert-system techniques and graph-theoretic techniques. The first category of approaches attempt to imitate the knowledge of domain experts, with examples including rule-based systems, e.g., [140], cased-based systems, e.g., [88], and model-based systems, e.g., [105]. The graph-theoretic techniques rely on a graphical model of the system, which describes the propagation for each specific fault, with examples including dependency graph [74], codebook technique [145], and belief-network [110]. These techniques suffer from two main drawbacks: first, they require accurate dependency information amongst network entities, which is usually not available for large scale enterprise networks; second, fault inference typically involves complicated computation and scales poorly with network size and complexity.

In contrast, our approach only requires elementary topological information and fault signatures to support matching over high-volume event data.

CHAPTER III

MUSI: NETWORK-AWARE CAUSALITY TRACKING

3.1 Introduction

Cascading behavior, diffusion and propagation of ideas, innovations, and information are fundamental processes taking place in socially networked systems [53, 75]. It is well recognized that *social influence* is one complex and subtle force that governs these dynamics [9, 114]: the actions of a user may induce his/her friends to behave in a similar way via their social connections. Interpreting a user's behavior in the context of his/her friends and correlating the actions of socially connected users is thus of tremendous interests from both analysis and design perspectives.

Recently social influence analysis has attracted intensive research interests, with examples such as differentiating the effects of social correlation (e.g., homophily and confounding) and social influence on users' activities [9], verifying the existence of correlation between personal behavior and social connection [114], estimating the influence strength of social ties [141], and inferring influence channels for implicit networks [55]. Most of these studies, however, focus on general, macro-level influence phenomena, irrespective of concrete users, actions or time-stamps.

Equally important is **understanding and modeling social influence at microscopic scale** (i.e., at the level of individual users, actions, and time-stamps), which may carry significant benefits for a range of applications, such as

- *Information filtering.* With the advance of Web 2.0 technologies and social media applications, the amount of information received by normal users can easily go beyond their processing capacities, e.g., an average Twitter[1] user receives

[1] http://twitter.com

over 93 tweets per day. Understanding at individual level how the browsing behaviors of users sharing common interests affect each other may facilitate *personalized* recommendation to effectively filter uninteresting information and deliver high-quality information in time.

- *Mobile phone-call service.* Recent studies on human mobility and phone call patterns [59, 115] have revealed strong correlation between geographical and social distance of individuals, e.g., users tend to call friends (one type of social influence) within geographically close areas. Modeling such influence at the level of individual users and locations offers valuable insight into characterizing user mobility and creating caller-callee profiles, which may significantly improve the efficiency of locating mobile users and devices (i.e., paging [147] operation).

- *Targeted advertising.* Through "word of mouth" and product comparison, a user's purchase of a product may trigger his/her friend to buy a functionally similar product, but with more personally favorable or ongoing fashion features. Clearly, accounting for the influence at the scale of particular users and time-stamps is crucial for developing successful advertisement strategy.

Not surprisingly, modeling microscopic social influence is in general a difficult task, featuring a series of unique challenges, including (i) the model should be able to account for the distinct characteristics of individual users, actions, and time-stamps, as well as their complex interconnections; (ii) it should support application-specific, micro-level influence mechanisms, e.g., how one action induces another concretely; (iii) it should provide scalable inference tools that enable to track the influence of occurred actions and predict the occurrence of new actions. None of these challenges are trivial. Consider challenge (iii) for example: given that we target the level of individual users and actions, we need to consider all the potential actions by all the users when performing inference, which implies prohibitive computational complexity

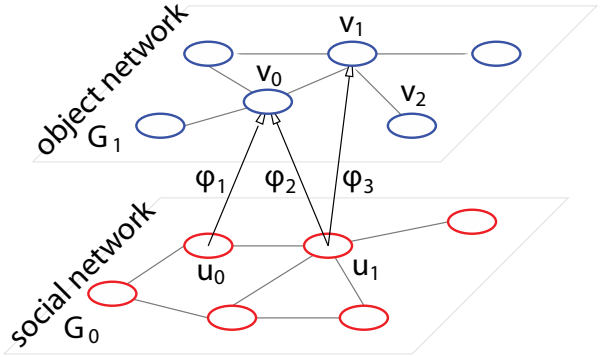


Figure 11: Social and object networks.

for large user or action space.

This work, to the best of our knowledge, represents the first attempt to model microscopic social influence at the granularity of individual users, actions, and time-stamps.

For the first challenge, we define a fairly general influence model that replicates all individual users and actions. We model actions as dynamic interactions between *social network* (formed by individual users) and *object network* (formed by targets of actions, e.g., blogs). An example is shown in Figure 11. In contrast of alternative models such as bipartite or heterogenous network [120], our model exhibits features desirable for our purpose: it fully captures individual users and actions as well as their interconnections; also, actions are represented as temporary inter-network connections, which reflect their dynamic nature.

For the second challenge, we propose a novel *heat field over product network* (HFPN) model to encode the concrete influence mechanism. Loosely speaking, in the product [67] of social and object networks, each node represents one potential action, while each edge represents the "influence channels" between two actions. The influence is modeled as "heat" that flows through such channels. The flexibility of this model lies in the potentially unlimited number of ways to specify the influence channels, and the influence flow rates.

For the third challenge, we develop a complete library of inference operations

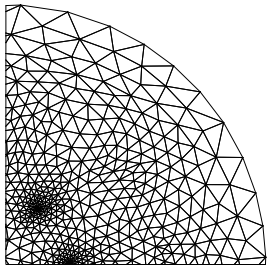


Figure 12: FEA (note the denser mesh around objects of interest).

that capacitate to continuously *track* the state of heat field in HFPN model, *update* the state once new actions are observed, and *predict* where and when new actions may occur. We carefully address the computational challenges for inference over HFPN model by drawing analogies to Finite Element Analysis (FEA). FEA has been successfully applied to numerically solve propagation models in physical systems, e.g., weather simulation, car frontal crash, etc. Its key idea is to vary the precision of numerical methods over the physical domain (see example in Figure 12). In this work, we present solutions that dynamically identify sub-domains of interest (over the entire space of potential actions), as guided by the prediction operation, and dynamically vary the precision of influence estimation around sub-domains of interest.

We entitle this complete framework of tracking, updating and predicting social influence as μSI. To evaluate its efficacy, we conduct an empirical evaluation of μSI in two seemingly disparate applications using real-life datasets. In the context of social tagging service (e.g., Del.icio.us[2]), we perform resource recommendation based on the prediction of μSI, which demonstrates significant improvement over conventional solutions (e.g., collaborative filtering) in terms of accuracy and freshness of recommendation. In the context of mobile phone call service, we apply μSI to improve the efficiency of paging operation by narrowing down the number of potential cells associated with call recipients. It is shown that μSI enhances the state-of-the-art callee profile-based approach [147] by reducing 25% of signaling traffic.

[2]http://delicious.com/

The remainder of this chapter is organized as follows. Section 3.2 introduces the fundamental concepts and building blocks of μSI; a library of inference operations are presented in Section 3.3, in which we also formalize the complete framework of μSI; Section 3.4 details scalable implementation of μSI, followed by its empirical evaluation in Section 3.5; Section 3.6 surveys relevant literature.

3.2 Fundamentals

In this section we present the construction of the cornerstones of μSI. We start with introducing the fundamental concepts used throughout the paper.

3.2.1 Preliminaries

We consider two types of entities. For simplicity of presentation, we use "users" (subjects of actions, e.g., Internet users) and "objects" (targets of actions, e.g., weblogs) to refer to them. We first formalize the concept of interaction[3].

Definition 10 (Interaction/Action)**.** *An interaction (or action) is a temporary (say, occur at time t) association between a user u and an object v, denoted by $\phi(u, v, t)$.*

Examples of interactions include Internet users reading weblogs, customers purchasing products, and mobile users connecting to base stations. Next we introduce the concept of network.

Definition 11 (Network)**.** *A network is modeled as a (directed) graph $G = (V, E)$ where V and E represent a set of entities and their interconnections, respectively. Each $e \in E$ may be further associated with weight $w(e)$ specifying the strength of such interconnection.*

The interconnections in social and object networks however may convey significantly different meanings. In social network, an interconnection between two users

[3]Without ambiguity, in following we use the terms "interaction" and "action" interchangeably.

indicates their social tie, which is the channel of social influence; while in object network, an interconnection between two objects indicates their proximity in certain sense, e.g., semantic similarity for two weblogs (as reflected in their reference to each other), geographical closeness for two cellular towers. Interestingly, such proximity may also become the channel via which one interaction influences another. For example, after reading one weblog, a user may further read another relevant one by following their reference.

We are interested in understanding and modeling the influence of interaction $\phi(u, v, t)$ on triggering[4] another (potential) interaction $\phi(u', v', t')$ $(t < t')$ as transferred via the social and object networks.

3.2.2 Building Blocks

To model the influence of occurred interaction $\{\phi\}$ on triggering (potential) interactions $\{\phi'\}$, we essentially need to describe (i) how the influence of $\{\phi\}$ is passed to $\{\phi'\}$, (ii) how much influence of $\{\phi\}$ is exerted over $\{\phi'\}$, and (iii) how the accumulated influence at $\{\phi'\}$ triggers their occurrence.

Block 1: Product Network Model

The structures of both social and object networks constrain the locality of the influence between interactions, i.e., interaction $\phi = (u, v, t)$ tends to affect "neighboring" users of u or objects of v.

Example 1. In Figure 11, assume G_0 as an online social network and G_1 as a blogosphere. After reading weblog v_0, u_0 may further read v_1 if interested in the topic covered by v_0. Meanwhile, knowing that u_0 reads v_0, u_1 may also be interested in reading it.

However, as we have revealed in Section 3.1, the concrete "influence channels"

[4]Strictly speaking, the influence of ϕ over ϕ' can also be negative, i.e., suppressing its occurrence, which can be typically modeled using signed edges [87].

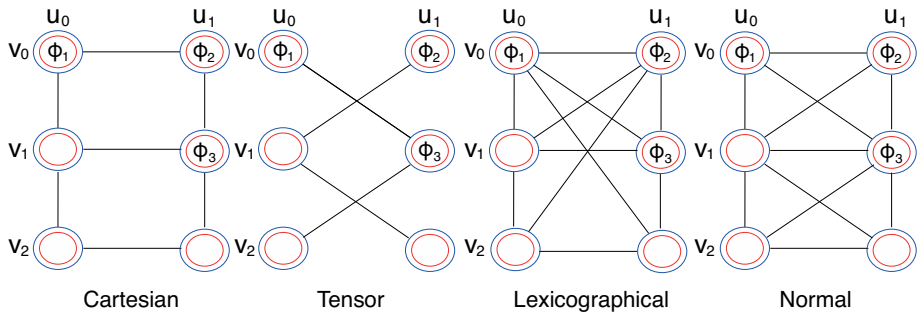


Figure 13: Examples of product networks.

can be highly application-specific; the influence may spread over a single network or over both networks simultaneously, or may feature fairly different spread speed within two networks. Instead of attempting for a one-size-fit-all model, we enable a class of influence channel instantiations based on the concept of product network.

Definition 12 (Product Network). *For two networks* $G_0 = (V_0, E_0)$ *and* $G_1 = (V_1, E_1)$*, their product network is defined as a graph* $G_\times = (V_\times, E_\times)$ *such that node* $\phi = (u, v) \in V_\times$ *if* $u \in V_0, v \in V_1$*, while the existence of edge* $\phi_i - \phi_j$ $((u_i, v_i) - (u_j, v_j))$ *can be specified by any logical statements in terms of* $u_i - u_j$*,* $v_i - v_j$*,* $u_i = u_j$*, and* $v_i = v_j$*, with examples including*

- *Cartesian product* $(G_\square)$*, if* $(u_i = u_j$ *and* $v_i - v_j \in E_1)$ *or* $(u_i - u_j \in E_0$ *and* $v_i = v_j)$*;*
- *Tensor product* $(G_\otimes)$*, if* $u_i - u_j \in E_0$ *and* $v_i - v_j \in E_1$*;*
- *Lexicographical product* $(G_\circ)$*, if* $u_i - u_j \in E_0$ *or* $(u_0 = u_1$ *and* $v_i - v_j \in E_1)$*;*
- *Normal product* $(G_\boxtimes)$*, if* $(u_i = u_j$ *and* $v_i - v_j \in E_1)$ *or* $(u_i - u_j \in E_0$ *and* $v_i = v_j)$ *or* $(u_i - u_j \in E_0$ *and* $v_i - v_j \in E_1$*.*

In the product of two networks G_0 and G_1, each node $\phi = (u, v)$ essentially corresponds to a (potential) interaction between $u \in G_0$ and $v \in G_1$, while each edge specifies an influence channel between two interactions. The flexibility of the specification of edges allows application-dependent instantiation, as shown in the next example.

Example 2. Figure 13 shows a subset of possible product networks corresponding to the two sub-networks $\{u_0, u_1\} \subset G_0$ and $\{v_0, v_1, v_2\} \subset G_1$ in Figure 11. In particular, the lexicographical product emphasizes the propagation of influence within the first network G_0, and allows interaction $\phi = (u, v)$ to affect all the interactions "neighboring" to u in G_0.

The next question is how to determine the weight w_{ij} of edge $\phi_i - \phi_j \in E_\times$. As we will show shortly, the weight w_{ij} essentially specifies the influence spread rate from ϕ_i to ϕ_j. In general it can be an arbitrary function of the attributes of all involved nodes $\{u_i, u_j, v_i, v_j\}$ and edges $\{u_i - u_j, v_i - v_j\}$. To make our model concise, in this paper we set w_{ij} based on the weight w_{ij}^0 and w_{ij}^1 of corresponding edges $u_i - u_j$ and $v_i - v_j$ (or nodes) in G_0 and G_1. We use tensor product and normal product as examples to show this.

For tensor product network $G_\otimes$, a natural way of doing so is to set $w_{ij} = w_{ij}^0 w_{ij}^1$; intuitively, we assume that the influence spread rate between two neighboring interactions $\phi_i = (u_i, v_i)$ and $\phi_j = (u_j, v_j)$ is proportional to the similarity of u_i and u_j, and that between v_i and v_j. In the matrix form, the weight matrix $W = W_0 \otimes W_1$, where $\otimes$ is the tensor operator. While for product networks that allow "stationary walk" in either one of the networks (e.g., in Cartesian product $G_\square$, u_i and u_j or v_i and v_j can be equivalent), we need to determine the weight of self-loop, e.g., $u_i - u_i$. If no further information is available, we may use a non-informative setting, say "1". In the matrix form, the weight matrix is specified as $W = W_0 \otimes I + I \otimes W_1$ where I is an identify matrix. If networks G_0 and G_1 allow self-loops, one can replace the diagonal elements of I with the weight of corresponding self-loops.

Block 2: Heat Field with Update Model

Heat diffusion [79] is a physical phenomenon wherein heat flows from position with high temperature to that with low temperature. We observe that the heat diffusion

model is a natural way to describe the spread of influence of occurred interactions to other (potential) interactions. Using this model, occurred interactions can be considered as heat sources featuring high temperature, and their influence is modeled as heat that flows to other (potential) interactions (featuring low temperature) through the underlying geometric structure of product network; the accumulated heat at a potential interaction indicate its tendency to happen; different initial heat sources (occurred interactions) or geometric structures (influence channels) determine different heat distributions. Heat kernel is used to quantify the amount of heat one point receives from another in a medium.

Over a known geometric manifold, the heat flows throughout a manifold with initial condition can be described by the second order differential equation: $\frac{\partial f(x,t)}{\partial t} - \Delta f(x,t) = 0$, where $f(x,t)$ is the heat at position x at time t, and Δf is the Laplace-Beltrami operator on a function f. The heat diffusion over a network can be considered as an approximation to the diffusion on the manifold. While the discussion can be readily generalized to the case of directed or probabilistic networks, for ease of presentation, here we consider the product network as an undirected graph.

Let $f_i(t)$ denote the heat at node[5] ϕ_i at time t. During a tiny time period $[t, t+\Delta t]$, we consider two types of heat movement.

- The heat node ϕ_i receives from or transfers to its neighbor ϕ_j through edge (influence channel) $\phi_i - \phi_j$, $T(i,j,t,\Delta t)$ over a period Δt. Intuitively, $T(i,j,t,\Delta t)$ should be proportional to the heat difference $(f_j(t) - f_i(t))$, the elapsed time Δt, and the weight w_{ij} of edge $\phi_i - \phi_j$ (which specifies the influence spread rate over $\phi_i - \phi_j$).

- The heat at ϕ_i diffuses to the surrounding media outside the network, $D(i,t,\Delta t)$, which captures the intuition that the influence tends to decay with time, even

[5]In the context of product network, we consider the terms "node" and "interaction" interchangeable.

without being transferred to other interactions. We have the assumption that $D(i,t,\Delta t)$ should be proportional to the heat $f_i(t)$ and elapsed time Δt.

Combing the two components above, we can formulate the heat change at a node ϕ_i at time t as follows:

$$f_i(t+\Delta t) - f_i(t) = \sum_{\phi_i - \phi_j \in E_\times} \alpha w_{ij}[f_j(t) - f_i(t)]\Delta t - \beta f_i(t)\Delta t$$

where α and β denote the diffusion rate inside and outside the network, respectively.

Expressed in a matrix form, the equation above is formulated as:

$$\mathbf{f}(t+\Delta t) - \mathbf{f}(t) = [\alpha(W-D) - \beta I]\mathbf{f}(t)\Delta t \tag{1}$$

where W is the weight matrix of the product network (which is symmetric due to the undirectionality of the network), the diagonal matrix D is defined as $D_{ii} = \sum_{\phi_i - \phi_j \in E_\times} W_{ij}$, and I represents an identify matrix. Let $H = W - D$. In the limit that $\Delta t \to 0$, we have $\frac{\mathrm{d}\mathbf{f}(t)}{\mathrm{d}t} = (\alpha H - \beta I)\mathbf{f}(t)$, which has the closed-form solution as:

$$\mathbf{f}(t) = e^{t(\alpha H - \beta I)}\mathbf{f}(0)$$

Here, the matrix exponential e^M of a matrix M is defined as $e^M = I + M + M^2/2! + M^3/3! + \ldots$. We name the heat distribution over the network as a *heat field* (HF).

It is noticed that this is a time-invariant system, i.e., for $\mathbf{f}(t_1) = \mathbf{f}(t_2)$, we have $\mathbf{f}(t_1 + \Delta t) = \mathbf{f}(t_2 + \Delta t)$. Therefore, in a sequel, this system can be incrementally computed, i.e., for any given $t_1, t_2 > 0$, we have $\mathbf{f}(t_1 + t_2) = e^{t_2(\alpha H - \beta I)}\mathbf{f}(t_1)$. Based on this nice property, we can readily incorporate the mechanism of *update* in our model:

Definition 13 (Heat Diffusion with Update)**.** *Under initial configuration* $\mathbf{f}(0)$*, the heat field* $\mathbf{f}(\cdot)$ *evolves over time. An update at time* t *interrupts the heat field* $\mathbf{f}(t)$ *and replaces it with a (possibly arbitrary) configuration* $\mathbf{f}^*$*, i.e., starting from* t*, the system evolves with* $\mathbf{f}^*$ *as the initial condition.*

We entitle the combination of product network and heat diffusion with update models as *heat field over product network* (HFPN). Using HFPN, we can continuously track the heat distribution over the product network; for newly occurred interactions, we consider them as new heat sources, and update the heat field accordingly. In general the initial heat should be set according to the *influence capacity* of the corresponding user or object (e.g., users who have more friends tend to have more capacity). In this paper, without loss of generality, we assume that each new heat source is initiated with one unit of heat.

Block 3: Invocation Model

Interactions observed at time t function as new sources that inject influence into the product network and potentially trigger some other interactions at a future time $(t + \Delta t)$. In this model, the accumulated influence at a node $\phi_i \in G_\times$ at given time $(t + \Delta t)$ indicates the potential that the corresponding interaction occurs at $(t + \Delta t)$. Yet, we need an invocation model that describes how the accumulated influence at ϕ_i actually triggers its occurrence.

Definition 14 (Invocation Model)**.** *The heat at a node ϕ_i indicates the potential of the corresponding interaction to be activated. We assume that with heat as $f_i(t)$, ϕ_i is activated with probability $g(f_i(t) - \lambda)$, where λ is a threshold. The function $g(\cdot)$ has the property that $g(x) = 0$ if $x \leq 0$, and is non-decreasing as x increases.*

One example instantiation of $g(\cdot)$ could be $g(x) = 1 - e^{-\gamma x \mathbf{1}_{x>0}}$, where $\mathbf{1}_{(\cdot)}$ represents an indicator function, and γ is a parameter controlling the increasing rate. The validity of this invocation model is empirically evaluated in real applications (details in Section 3.5).

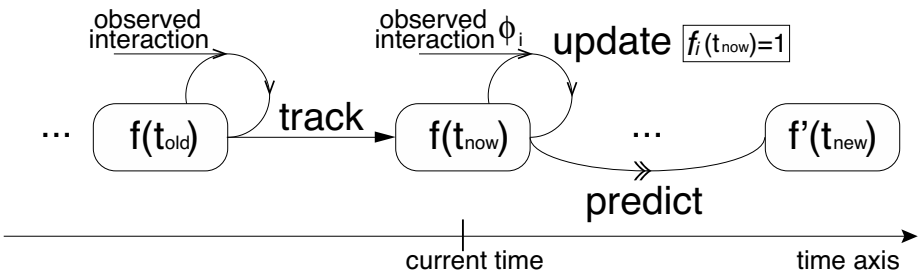


Figure 14: Operations of μSI.

3.3 Tracking, Updating, Predicting

Atop the heat field over product network and invocation models, we are now ready to construct the complete framework of μSI, which enables to track, update and predict interactions between social and object networks G_0 and G_1.

3.3.1 A Complete Framework

Starting with a heat field of the product network $G_\times$ of G_0 and G_1 at time 0, $\mathbf{f}(0) = 0$, μSI capacitates us to perform the following three operations:

- Track. Given the estimation of heat field $\mathbf{f}(t_{old})$ at the time t_{old} when the latest interaction is observed, we estimate the current heat field $\mathbf{f}(t_{now})$;
- Update. For an interaction ϕ that is observed at the current time t_{now}, we update the estimation of heat field $\mathbf{f}(t_{now})$ accordingly;
- Predict. Based on the estimation regarding current heat field $\mathbf{f}(t_{now})$, we predict the heat field state $\hat{\mathbf{f}}(t_{new})$[6] at a future time t_{new}.

The relationships between operations above are illustrated in Figure 14. Equipped with the three operations, we are able to query the state of heat field at arbitrary time-stamp[7].

[6] We use $\hat{f}$ and f to distinguish the estimates using predicting and tracking, respectively.

[7] Note that essentially we can also derive a "Backtrack" operation that takes current state $\mathbf{f}(t_{now})$ as input, and estimates the heat field Δt ago: $\bar{\mathbf{f}}(t_{now} - \Delta t)$.

Next we discuss how to implement these operations using the HFPN model. The implementation of *Track* and *Update* is fairly straightforward. Specifically, for tracking, given previous state $\mathbf{f}(t_{old})$, the current heat field can be estimated as $\mathbf{f}(t_{now}) = e^{(t_{now}-t_{old})(\alpha H-\beta I)}\mathbf{f}(t_{old})$; for updating, given an observed interaction $\phi_i = (u_i, v_i, t_{now})$, we consider it as a new heat source, and update $f_i(t_{now}) = 1$.

We now focus our discussion on the *Predict* operation. It essentially takes as input the current state $\mathbf{f}(t_{now})$, and attempts to estimate the state at a future time t_{new}. The difficulty lies in taking account of those interactions that may occur during the time interval $(t_{now}, t_{new}]$. These potential interactions (invocation) may make the actual result significantly deviate from the invocation-agnostic prediction $e^{(t_{new}-t_{now})(\alpha H-\beta I)}\mathbf{f}(t_{now})$.

3.3.2 More on Prediction

A naïve solution would be a Monte Carlo simulation scheme: one divides the time axis into a sequence of high-resolution "time-ticks" of length δ_t: at the $(i+1)$-th time-tick, estimate $\mathbf{f}(t+(i+1)\delta_t)$ based on the simulation result of $\mathbf{f}(t+i\delta_t)$ (essentially a tracking operation); for an interaction ϕ_i with heat above λ, activate it with probability according to $g(f_i(t+(i+1)\delta_t) - \lambda)$ in the invocation model (essentially a sampling operation). While intuitive and faithful to the invocation mechanism, this scheme suffers scalability issues especially when the network scale is large or the granularity of time-tick is small, because it involves the tracking and sampling (possibly for all potential interactions) operations at each time-tick. We consider this scheme as the baseline solution in comparison.

We attempt to construct a scheme that provides flexible trade-off between the number of tracking and sampling operations, and the quality of simulation results. We start with defining the concept of *activation window*.

Definition 15 (Activation Window)**.** *For given threshold λ and interaction ϕ_i, a*

time interval $[t_s, t_e]$ *is called an activation window of* ϕ_i *if (i)* $f_i(t) > \lambda$ *for* $t \in (t_s, t_e)$ *and (ii)* $\nexists\, [t'_s, t'_e] \supset [t_s, t_e]$ *has such property. The peak value* $\max_t f_i(t)$ $(t \in [t_s, t_e])$ *is called the window height, and* t_s *and* t_e *are the starting and ending bounds of the window.*

Clearly, according to the invocation model, inside each activation window, the potential interaction obtains the opportunity to be activated. A model faithful to the exact simulation should guarantee that such opportunities are fully respected. We therefore strive for the following objective: *each interaction is ensured with high probability one activation opportunity during each of its windows, while the probability is positively correlated with the height of the window.* This is similar to Finite Element Analysis in spirit.

Based on this approximation, we propose a "lazy-probing" scheme that allows to skip state estimation during a "safe window" without denying the activation opportunity for qualified interactions. To do so, we first need to understand the maximum length of such safe window. For simplicity of exposition, we assume that each occurred interaction is initiated with one unit of heat, and the weights of all the edges are uniformly set as one. We have the following theorem:

Theorem 1. *At time* t*, for an interaction* ϕ_i *with current heat* f $(f < \lambda)$*, degree* d*, the ending bound* t_e *of its next activation window with height* h *must satisfy:*

$$t_e - t \geq \frac{1}{\alpha d + \beta} \ln \frac{\alpha d - (\alpha d + \beta) f}{(\alpha d - (\alpha d + \beta) h)(\lambda + 1 - h)} \tag{2}$$

Proof. Assume that the next appearing activation window has height h. We divide the time interval $[t, t_e]$ into two phases, *charge* and *discharge*. Let Δt_c and Δt_d be their length. In the charge phase, the heat of ϕ increases; after reaching the expected height h, it enters the discharge phase, and the heat drops below λ. These concepts are illustrated in Figure 15. We intend to find the lower bounds of Δt_c and Δt_d. Clearly, $t_e - t \geq \Delta t_c + \Delta t_d$.

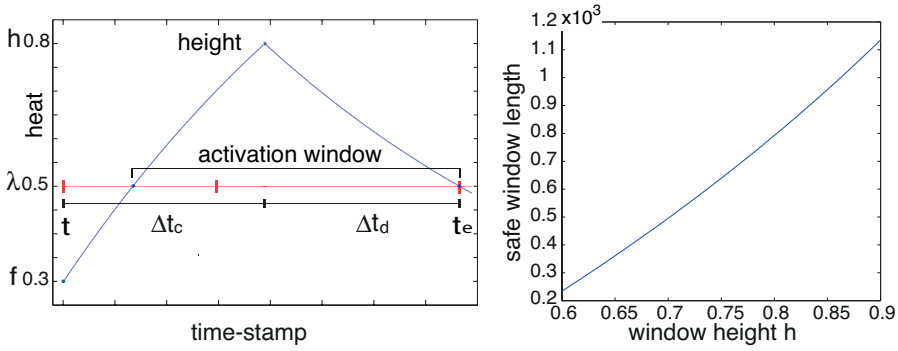


Figure 15: Charge and discharge phases of an activation window, and length of safe window with respect to window height h (default setting: $\alpha = 0.2$, $\beta = 0.05$, $\lambda = 0.5$, $f = 0.3$, $h = 0.8$, $d = 2$).

First notice that the lower bound of Δt_c is achieved when all of neighboring interactions have maximum possible heat "1". In this case, the heat equation for the charge phase can be written as $\frac{\mathrm{d}f_i}{\mathrm{d}t} = \alpha d(1 - f_i) - \beta f_i$, with initial condition $f_i(0) = f$ and end condition $f_i(\Delta t_c) = h$. Solving the differential equation, we get $\Delta t_c \geq \frac{1}{\alpha d+\beta} \ln \frac{\alpha d-(\alpha d+\beta)f}{\alpha d-(\alpha d+\beta)h}$. Meanwhile, the lower bound of Δt_d is achieved when ϕ_i receives no heat, but diffuses its heat to neighboring nodes and the media. We have the corresponding heat equation as $\frac{\mathrm{d}f_i}{\mathrm{d}t} = -(\alpha + \beta)f_i$, with initial condition $f_i(0) = h$ and end condition $f_i(\Delta t_d) = \lambda$. Solving the equation gives us: $\Delta t_d \geq \frac{1}{\alpha d+\beta} \ln \frac{1}{\lambda+1-h}$. Combining these two inequalities leads to Eq.(2). The overall behavior of $(\Delta t_c + \Delta_d)$ with respect to window height h is illustrated in the right plot of Figure 15, which shows approximately linear correlation. □

Based on Theorem 1, we introduce our lazy-probing prediction scheme, as sketched in Algorithm 3. At each examined time-stamp t', we attempt to activate each interaction ϕ_i with heat above threshold λ according to probability $g(\hat{f}_i(t') - \lambda)$ (line 5-7); for each interaction with heat below λ, we estimate the maximum skippable interval according to Eq.(2) (line 8-9); the minimum of all such skippable intervals is considered as the global skippable interval Δt, and the state estimation is updated for time $(t' + \Delta t)$ (line 10). This process repeats until $t' = t_{new}$.

In the scheme above, one key parameter is the expected height h of the activation window, which controls the heat intensity required to guarantee an activation chance. Next we discuss how to select a proper h. We consider the two probabilities, the activation probability, $\mathsf{prob}(c|h)$ (c is binary variable, with "1" indicating the occurrence), and the initial distribution of h, $\mathsf{prob}(h)$ (which can be typically estimated using historical data). We are interested in the posterior probability $\mathsf{prob}(h|c) \propto \mathsf{prob}(c|h)\mathsf{prob}(h)$; that is, given that an interaction is activated, what is the probability that its window height is h. One may select the threshold $h^* = \arg\max_h \int_h^1 \mathsf{prob}(x|c=1)\mathrm{d}x \int_\lambda^h \mathsf{prob}(x) \ln((\alpha d - (\alpha d + \beta)h)(\lambda + 1 - h))\mathrm{d}x$, where the first term corresponds to the range of heights that are guaranteed an activation chance, and the second term is proportional to the average skipped interval, $\bar{d}$ represents the average incoming degree, and κ ($\kappa > 0$) is a parameter controlling the trade-off.

Algorithm 3: Lazy-probing prediction.

Input: current time t_{now}, current state $\mathbf{f}(t_{now})$, future time t_{new}, activation threshold λ, activation window height h

Output: estimated future state $\hat{\mathbf{f}}(t_{new})$

$t' \leftarrow t_{now}$, $\hat{\mathbf{f}}(t') \leftarrow \mathbf{f}(t_{now})$;
while $t' < t_{new}$ **do**
  $\Delta t \leftarrow t_{new} - t'$;
  foreach $\phi_i \in V_\times$ **do**
    if $\hat{f}_i(t') \geq \lambda$ **then**
      `// sampling (activation trial)`
      if *i is not activated* **then**
        set $\hat{f}_i(t') = 1$ with prob. $g(\hat{f}_i(t') - \lambda)$;
    else
      `// estimate minimum skippable interval`
      $\Delta t \leftarrow \min\{\Delta t, \frac{1}{\alpha d + \beta} \ln \frac{\alpha d - (\alpha d + \beta)\hat{f}_i(t')}{[\alpha d - (\alpha d + \beta)h](\lambda + 1 - h)}\}$
  `// update estimation`
  $\hat{\mathbf{f}}(t' + \Delta t) \leftarrow e^{\Delta t(\alpha H - \beta I)}\hat{\mathbf{f}}(t')$;
  $t' \leftarrow t' + \Delta t$;

3.4 Computational Implementation

In this section, we detail the computational implementation of μSI. As we have revealed in Section 3.3, the operations of μSI are constructed atop the model of heat field, with equation $\mathbf{f}(t+\Delta t) = e^{\Delta t(\alpha H - \beta I)}\mathbf{f}(t)$ as its cornerstone. Directly evaluating this model, however, features prohibitive computational complexity: (i) the strict computation of matrix exponential involves an infinite sequence of matrix multiplications; (ii) for two networks both of cardinality n, the matrix H (corresponding to the product of the networks) is at the scale of n^4; (iii) furthermore, the tracking and predicting operations require to evaluate the equation for varying time interval Δt and initial condition $\mathbf{f}(t)$.

In what follows, we expose possible solutions that can simplify the computation by (i) fully exploiting the structural properties of product network and (ii) intelligently caching and reusing invariant intermediate results. We start our discussion with the properties of matrix exponential.

3.4.1 Matrix Exponential

The exponential of matrix M, e^M, is defined as an infinite sequence: $e^M = I + M + M^2/2! + M^3/3! + \ldots$. Clearly, this formation is not amenable to an efficient implementation. In general, we may resort to a discrete approximation: $e^M \approx (I + M/N)^N$, where N is the number of iterations that controls the error $||(I+M/N)^N - e^M||$; while for a set of special cases of M, closed-form solutions exist.

Property 2. *If M is a diagonal matrix, with the i-th diagonal element as m_i, e^M is also a diagonal matrix, with the i-th diagonal element as e^{m_i}.*

Property 3. *If M is a nilpotent matrix (i.e., $M^q = 0$ for some integer q), then e^M is a finite sequence: $e^M = I + M + M^2/2! + \ldots + M^{q-1}/(q-1)!$. In this case, the computation of e^M can be implemented in an incremental manner: starting from the q-th entry, let $X_0 = \frac{1}{q!}M$, $X_{i+1} = (X_i + \frac{1}{(q-i-1)!})M$, $X_q = (X_{q-1} + I)M$, and $e^M = X_q$.*

Property 4. *If $M = X + Y$, and X and Y are commutable, i.e., $XY = YX$, then $e^M = e^X e^Y$.*

Following we will identify the structural features of product network matrix, and exploit the properties above to speed up the computation.

3.4.2 Heat Field Equation

Now we look at the concrete heat diffusion equation in our model. We target the following equation, assuming that $\mathbf{f}(t)$ is given:

$$\mathbf{f}(t + \Delta t) = e^{\Delta t(\alpha H - \beta I)}\mathbf{f}(t)$$

where $H = W - D$, W denotes the weight matrix of product network $G_\times = (V_\times, E_\times)$, and D is a diagonal matrix with the i-th diagonal element as the sum of the i-th row of W, i.e., $D_{ii} = \sum_{\phi_i - \phi_j \in E_\times} W_{ij}$. Due to the space constraint, more specifically, we consider $G_\times$ as an undirected graph. In this case, we have the equation as $\mathbf{f}(t + \Delta t) = e^{\Delta t \alpha W - \Delta t(\alpha D + \beta I)}\mathbf{f}(t)$. Clearly, the two matrices $\Delta t \alpha W$ and $-\Delta t(\alpha D + \beta I)$ are commutable; the heat equation is therefore calculated as: $\mathbf{f}(t+\Delta t) = e^{-\Delta t(\alpha D + \beta I)} e^{\Delta t \alpha W}\mathbf{f}(t)$. While the computation of $e^{-\Delta t(\alpha+\beta)}$ (which is diagonal) is straightforward based on Property 2, we focus our discussion on the computation of $e^{\Delta t \alpha W}\mathbf{f}(t)$.

Recall that in our setting W corresponds to the product of two networks. Let W_i $(i = 0, 1)$ represent the weight matrices of corresponding component networks. Conceivably, the computational complexity of heat equation depends on the concrete formation of W in terms of $\{W_i\}$. Recall the types of product network as defined in Section 3.2.2, which can be categorized as two classes, namely, *Kronecker product* and *Kronecker sum*. The Kronecker product of two matrices M_0 (of size m-by-n) and M_1 (of size p-by-q) is a mp-by-nq block matrix $M_0 \otimes M_1$ with the i-th row, j-th column block as $[M_0]_{ij} M_1$. Meanwhile, the Kronecker sum of M_0 and M_1 is defined as $M_0 \oplus M_1 = M_0 \otimes I + I \otimes M_1$. Cartesian and Normal products belong to the class of Kronecker sum, Tensor product belongs to the class of Kronecker product,

while Lexicographical product belongs to the mixture of the two classes. Hence, we use Cartesian and Tensor products as examples to show how we can simplify the computation of heat diffusion equation.

For Cartesian product, the weight matrix W is defined as a Kronecker sum $W = W_0 \otimes I + I \otimes W_1$. The matrix exponential of Kronecker sum has the following nice property.

Property 5. *For matrices M_0 and M_1, the exponential of their Kronecker sum satisfies $e^{M_0 \oplus M_1} = e^{M_0} \otimes e^{M_1}$.*

We therefore have the following transformation:

$$e^{\Delta t \alpha W} = e^{\Delta t \alpha W_0} \otimes e^{\Delta t \alpha W_1}$$

Now, the challenge lies in computing the matrix-vector multiplication $e^{\Delta t \alpha W_0} \otimes e^{\Delta t \alpha W_1} \mathbf{f}$. According to the definition of Kronecker product, directly computing it requires $O(n^4)$ time for matrices $|M_i| = n^2$ $(i = 0, 1)$. Let $\mathbf{m} = \mathsf{vec}(M)$ denote the *vectorization* of matrix M formed by stacking the columns of M into a single column vector $\mathbf{m}$, and $M = \mathsf{dvec}(\mathbf{m})$ be the inverse operation (here, we omit the dimension restriction on M). We have the next property.

Property 6. *For three (multiplication compatible) matrices $M^{(0)}$, $M^{(1)}$, and X, the following vectorization property holds: $(M_0 \otimes M_1)\mathsf{vec}(X) = \mathsf{vec}(M_1 X M_0^T)$.*

Clearly, the matrix-vector multiplication can be computed in $O(n^3)$ time for $|M_i| = n^2$ $(i = 0, 1)$. Especially, when M_0 and M_1 are sparse (which is typically the case), it can be more efficient: if there are $O(n)$ non-significant entries in M_0 and M_1, the computation takes only $O(n^2)$ time.

Summarizing the discussion above, for Cartesian product network, we have the

following transformation of heat field equation:

$$\begin{aligned}
\mathbf{f}(t+\Delta t) &= e^{\Delta t(\alpha H-\beta I)}\mathbf{f}(t) \\
&= e^{-\Delta t(\alpha D+\beta)I}e^{\Delta t\alpha W}\mathbf{f}(t) \\
&= e^{-\Delta t(\alpha D+\beta I)}e^{\Delta t\alpha W_0}\otimes e^{\Delta t\alpha W_1}\mathbf{f}(t) \\
&= e^{-\Delta t(\alpha D+\beta I)}\mathsf{vec}(e^{\Delta t\alpha W_1}\mathsf{dvec}(\mathbf{f}(t))e^{\Delta t\alpha W_0^T})
\end{aligned}$$

Now let us consider the case of Tensor product. In this case, the weight matrix W is given as a Kronecker product $W = W_0 \otimes W_1$. We introduce the *Jordan canonical form* [52].

Property 7. *Let $P_iJ_iP_i^{-1}$ be the non-trivial Jordan canonical form of matrix W_i $(i = 0, 1)$. The exponential of Kronecker product of W_0 and W_1, $e^{W_0\otimes W_1}$, satisfies*

$$e^{W_0\otimes W_1} = (P_0 \otimes P_1)e^{J_0\otimes J_1}(P_0^{-1} \otimes P_1^{-1})$$

Here J_0 and J_1 are *block diagonal* matrices (say, with r and s blocks, respectively). Let $J_i^{[j]}$ denote the j-th diagonal block of J_i, defined as

$$J_i^{[j]} = \begin{bmatrix} \lambda_{i,j} & 1 & & \\ & \lambda_{i,j} & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_{i,j} \end{bmatrix}$$

where $\lambda_{i,j}$ is the j-th eigenvalue of W_i. The matrix $J = J_0 \otimes J_1$ is also a block diagonal matrix (with rs blocks). Let J_{ij} be the ij-th diagonal block of J: $J_{ij} = J_0^{[i]} \otimes J_1^{[j]}$. It can be derived that the exponential of J_{ij} is a *block semi-circulant* matrix, given as:

$$e^{J_{ij}} = e^{\lambda_{0,i}J_1^{[j]}}\begin{bmatrix} 1 & \frac{J_1^{[j]}}{1!} & \frac{(J_1^{[j]})^2}{2!} & \cdots & \frac{(J_1^{[j]})^{m_i-1}}{(m_i-1)!} \\ & 1 & \frac{J_1^{[j]}}{1!} & \cdots & \frac{(J_1^{[j]})^{m_i-2}}{(m_i-2)!} \\ & & \ddots & & \\ & & & & 1 \end{bmatrix}$$

where m_i is the dimension of the i-th block of W_0.

In general the Jordan form is sensitive to numerical rounding [52]; fortunately, we can typically represent the elements of the matrices W_0 and W_1 using integers by scaling the parameters α and β (more details in Section 3.5).

Note that each block $J_1^{[j]}$ is the sum of a diagonal and a nilpotent matrices; the computation of its exponential is fairly straightforward, using Property 2, 3 and 4. One therefore only needs to compute rs small-scale matrix exponentials. Also, it is rather easy to insert the component $\Delta t\alpha$ into the model above by replacing $J_1^{[j]}$ with $\Delta t\alpha J_1^{[j]}$. Therefore, all the intermediate results, e.g., the matrix decompositions, are reusable for different Δt (details in Section 3.4.3).

The complete procedure of estimating $\mathbf{f}(t + \Delta t)$ is summarized in Algorithm 4. We repeatedly apply Property 6 in computing matrix-vector multiplication (line 2, 4). The only exception is the matrix $e^{\Delta t\alpha J_0 \otimes J_1}$, which in general cannot be further decomposed into Kronecker product. It is, however, a block diagonal matrix, which implies that $e^{\Delta t\alpha J_0 \otimes J_1}\mathbf{f}$ can be typically computed in $O(n^3)$ for $|W_i| = n^2$ $(i = 0, 1)$. The overall complexity is thus $O(n^3)$ (not including computing Jordan canonical form, which is one-time cost).

Algorithm 4: Estimation of $\mathbf{f}(t + \Delta t)$ (Tensor product).

Input: previous state $\mathbf{f}(t)$, elapsed time Δt, weight matrices W_0, W_1
Output: current state $\mathbf{f}(t + \Delta t)$

```
// Jordan canonical form (reusable)
```
1 $P_i J_i P_i^{-1} \leftarrow W_i$ $(i = 0, 1)$;
```
// apply Property 6
```
2 $\mathbf{f} \leftarrow \mathsf{vec}(P_1^{-1}\mathsf{dvec}(\mathbf{f}(t))P_0^{-1T})$;
3 $\mathbf{f} \leftarrow e^{\Delta t\alpha J_0 \otimes J_1}\mathbf{f}$;
```
// apply Property 6 again
```
4 $\mathbf{f} \leftarrow \mathsf{vec}(P_1\mathsf{dvec}(\mathbf{f})P_0^T)$;
5 $\mathbf{f}(t + \Delta t) \leftarrow e^{-\Delta t(\alpha D + \beta I)}\mathbf{f}$;

3.4.3 Optimization for Time Axis

It is noticed that in the heat field equation for given $\mathbf{f}(t)$, α and β, $\mathbf{f}(t+\Delta t)$ is essentially a function of Δt.

Clearly, if the time dimension can be discretized into time-ticks δt, one can essentially cache the intermediate results $e^{\Delta t(\alpha H-\beta I)}$ for $\Delta t = \delta t$, $2\delta t$, ..., and reuse them in state estimation. In following we use $\kappa(i)$ to denote the "kernel" $e^{i\delta t(\alpha H-\beta I)}$. The saving of computation, however, comes at the expense of storage: it is noted that $\kappa(\cdot)$ is a matrix of size n^4; storing $\kappa(i)$ for i from 1 to m requires $O(mn^4)$ space. We introduce two optimization strategies to strike a balance between computation and storage costs.

Geometric Increment

Instead of storing $\kappa(i)$ for all i's from 1 to m, we may selectively cache a subset of i's, and leave the rest of $\kappa(i)$'s to online computation.

Recall that the formation of $\kappa(\cdot)$ is a matrix exponential, we have the following "linearity" property: $\kappa(i_1+i_2)=\kappa(i_1)\kappa(i_2)$. Hence, we may store $\kappa(i)$ in a *geometric* manner, i.e., $i = 2^0, 2^1, \ldots, 2^{\log_2 m}$ (assuming m is an exponential of 2). Clearly, within this model, only $O(\log mn^4)$ space is required (which will be further reduced in the next step of optimization).

Now consider computing $\mathbf{f}(t+k\delta t) = \kappa(k)\mathbf{f}(t)$ where $k \in [1,m]$, which can be essentially transformed as the multiplication of less than $\log_2 m$ matrices $\kappa(\cdot)$ and vector $\mathbf{f}(t)$. More formally, we define an indicator function $h(k,i) = \mathbf{1}_{k/2^{i-1}\%2=1}$ (integer division/module); that is, it returns 1 if the i-th bit (in increasing order of magnitude) of k is 1 or 0 otherwise. We have

$$\mathbf{f}(t+k\delta t) = \left[\prod_{i=1}^{\log_2 m}\left((1-h(k,i))I + h(k,i)\kappa(2^i)\right)\right]\mathbf{f}(t)$$

It may seem at the first glance that the cost of computing the matrix multiplication of $\kappa(\cdot)$ (of size n^4) and vector may dwarf that of directly evaluating $\kappa(k)$ (which

involves matrix exponential). Next we introduce the second optimization strategy, *partial materialization*. Instead of caching an entire matrix exponential, we may only need to cache its core part, which, in conjunction of geometric increment, leads to a both space and computation efficient scheme.

Partial Materialization

Following the discussion in Section 3.4.2, we exemplify our techniques with the cases of Cartesian and Tensor product networks. For ease of exposition, we consider the evaluation of $\mathbf{f}(t + k\delta t)$ and $k = 2^{k_1} + 2^{k_2} + \ldots + 2^{k_j}$.

Recall that in Cartesian product, the heat field function is defined as $\mathbf{f}(t + k\delta t) = e^{-k\delta t(\alpha D+\beta I)} e^{k\delta t\alpha W_0} \otimes e^{k\delta t\alpha W_1} \mathbf{f}(t)$. Clearly, the term $e^{-k\delta t(\alpha D+\beta I)}$ is directly computable; while the part $e^{k\delta t\alpha W_0} \otimes e^{k\delta t\alpha W_1}$ can be re-written as:

$$\begin{aligned} e^{k\delta t\alpha W_0} \otimes e^{k\delta t\alpha W_1} &= (\prod_{i=1}^{j} e^{2^{k_i}\delta t\alpha W_0}) \otimes (\prod_{i=1}^{j} e^{2^{k_i}\delta t\alpha W_1}) \\ &= (\prod_{i=1}^{j} \kappa_0(k_i)) \otimes (\prod_{i=1}^{j} \kappa_1(k_i)) \end{aligned}$$

Instead of caching the entire matrix exponential in the heat diffusion equation, we only cache the part $\{\kappa_0(x)\}$ and $\{\kappa_1(x)\}$ $(x = 0, \ldots, \log_2 m - 1)$. First notice that this scheme reduces the storage cost to $O(n^2 \log m)$. Meanwhile, the computation now consists of no more than $(2\log_2 m + 2)$ (2 corresponds to the final matrix-vector multiplication as indicated in Property 6) matrix multiplication, thus featuring complexity of $O(n^{2.376} \log m)$ if the state-of-the-art matrix multiplication algorithms (e.g., Coppersmith-Winograd algorithm [31]) are applied, which improves significantly the complexity of $O(mn^3)$ of using methods such as Padé approximant directly.

In the case of Tensor product, the same principle applies. It is noticed that the major cost of computing $\mathbf{f}(t+k\delta_t)$ is attributed to $e^{k\delta t\lambda_i J_1^{[j]}}$ for each Jordan block $J_1^{[j]}$. We may cache $e^{2^x \delta t\lambda_i J_1^{[j]}} \triangleq \pi(i, j, x)$ $(x = 0, \ldots, \log_2 m-1)$, and for $k = 2^{k_1}+2^{k_2}+\ldots+2^{k_j}$, $e^{k\delta t\lambda_i J_1^{[j]}}$ can be efficiently evaluated using $\prod_{i=1}^{j} \pi(i, j, k_i)$. The complexity analysis is similar to the case of Cartesian product, which is omitted here.

To achieve further space saving, instead of storing a set of entire matrices. We may store a basis matrix M and the difference matrix $M' \ominus M$ for rest M'. Each matrix M' can then be constructed based on the basis M and the difference $M' \ominus M$. This scheme is especially effective when the difference between M and M' is in-significant, which is indeed the case as shown in our experiments.

3.5 Application and Evaluation

Following we empirically evaluate the efficacy of μSI in two concrete applications, namely, social tagging service and mobile phone call service, using real-life datasets collected from these services. We start with introducing the datasets used in our experiments.

3.5.1 Experimental Datasets

For the application of social tagging service, we used two datasets. The first dataset is the social network corresponding to a set of IBM employees who participated in the SmallBlue project [93]. The dataset comprises two snapshots of the network as of January 2009 and July 2009, involving 41,702 and 43,041 individuals, respectively. All links of the network are of uniform weight. The second dataset is an archive of bookmarks tagged by the individuals appearing in the SmallBlue dataset, collected by Dogear [39], a personal bookmark management application. The archive contains 20,870 bookmark records, with respect to 7,819 urls. Since the urls are fully anonymized, we solely use their associated tags define their semantic relationships and to construct the url network. Let $\mathsf{tag}(r)$ denote the set of tags associated with url r. Specifically, in our implementation, we define the weight w_{ij} of the link $r_i r_j$ using their *Jaccard similarity coefficient*,

$$w_{ij} = w_{ji} = \frac{|\mathsf{tag}(r_i) \cap \mathsf{tag}(r_j)|}{|\mathsf{tag}(r_i) \cup \mathsf{tag}(r_j)|}$$

Note that both social and object networks can be refined if extra information is available, e.g., the link between urls can incorporate directionality information if the reference relationship is known (e.g., one blog refers to another), while the link between users can incorporate weight information if the strength of their social tie is available [141].

For the application of mobile phone-call service, we use Reality Mining dataset [41] in our experiments. It consists of the communication logs of 94 users over the period from September 2004 to June 2005, with information including location logs (nearest base stations), phone call logs, social relationships (friends or non-friends). We use the cell transition information in location logs to infer the cell network (totally 3,138 cells), and use the friend and non-friend relationships to construct the social network. In both networks, the links are of uniform weight. We extract the part of logs corresponding to communications (voice, data, SMS) between the 94 users. The resulting archive contains 3,984 communication records.

We implement our models in Matlab. All the experiments are conducted on a workstation running Linux with 4G RAM and 2GHz Intel Core 2 Duo.

3.5.2 Case Study 1: Resource Recommendation in Social Tagging Service

Social tagging services (e.g., Del.icio.us[8], CiteULike[9], etc.) allow users to annotate online resources (e.g., urls) with freely chosen keywords (tags), which provide meaningful collaborative semantic data that could be exploited by recommender systems. However, performing resource recommendation based on tagging data is a challenging problem due to: First, it involves multi-type, complexly interrelated entities, namely, users, tags and resources, the available tagging data in the form of user - tag - resource may only expose a fairly sparse set of views of the hidden semantic correlation between users and resources. Second, the tagging behavior of users may demonstrate

[8]http://delicious.com/
[9]http://www.citeulike.org/

strong time sensitivity, e.g., most tags of a resource may be generated during a short time period when the resource features an ongoing hot topic among users, while the historical tagging records of a user may not reflect his/her current interest.

Existing solutions can be roughly categorized as two main classes. The first class attempts to augment recommender systems by leveraging tagging data (e.g., [18, 35]). They treat tags either as users or as items and then apply traditional item-based or user-based collaborative filtering (CF) algorithms, in which the structural information of tagging data is partially lost. The second class of solutions specifically exploit the tagging information (e.g., [57, 112]). They typically assume bipartite graph structures for the relationships between users and tags, and that between tags and resources, and attempt to construct a low dimensional representation as an approximation to this underlying semantic manifold. In both cases, the semantic-rich interconnections between users (and that between resources) have not been fully exploited. Further, these solutions are agnostic to the temporal sensitivity of users' tagging behavior, and rely on historical tagging records to infer users' current interests.

With the help of the microscopic social influence model, we provide a *dynamic, social* perspective of resource recommendation. We consider a tagging action as a dynamic interaction between the social network and the resource network. We have the following assumptions. First, the tagging behavior of a user regarding a resource indicates his/her ongoing interest in the resource. Second, the influence of a tagging action may spread over both social and resource networks: (i) socially close users tend to share common interests, and thus one may follow his/her peers to tag the same resource (which is frequently observed in social networking sites); (ii) meanwhile, users tend to tag similar resources if they find the topic covered by these resources interesting.

Based on these two observations, we propose a novel resource recommendation approach for social tagging service. Intuitively, we use the social influence model to

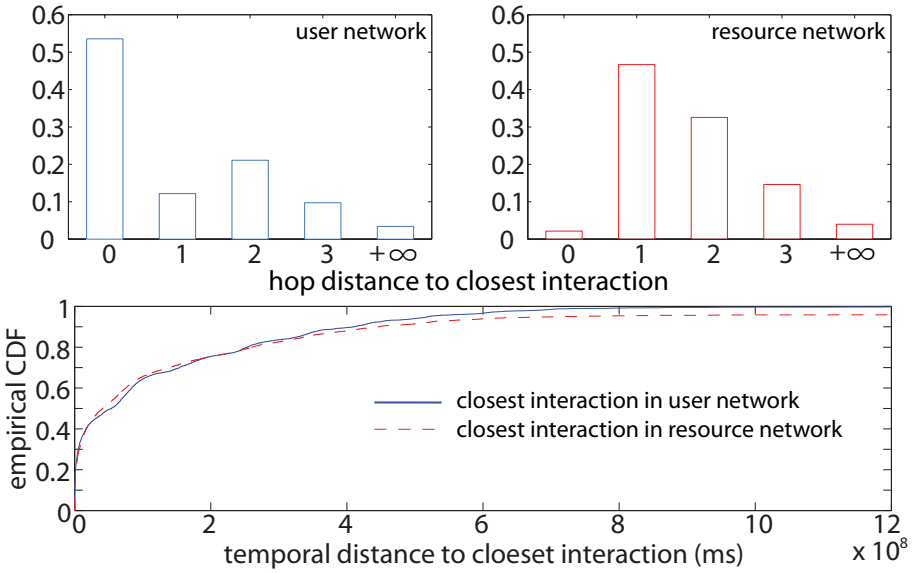


Figure 16: Spatial (in user and resource networks) and temporal locality of influence of tagging actions.

capture the influence of specific tagging actions on inducing other tagging actions. Leveraging this model, we may predict the tagging actions of potential interested users before the actions actually occur, such that the resources can be recommended to these users in advance of time.

Assumption Validation and Model Instantiation

We start with analyzing the dataset to validate the assumptions of our solution. Specifically three important assumptions are made in our model: (i) the influence of occurred interactions on triggering other interactions is local to both networks; (ii) the influence may exhibit different patterns in the two networks (captured by the concrete instantiation of production network and parameterization); and (iii) such influence tends to decay over time.

We examine the entire history of interactions (tagging actions) that occurred during the time period from April 20, 2008 to August 20, 2009. For each interaction (u, v) (u in user space and v in resource space), we search for its closest neighboring interaction (u', v') in the history before (u, v) along the dimensions of user and resource,

i.e., $\min_{(u',v')} \texttt{dist}(u,u') \vee \texttt{dist}(v,v')$, and measure this closest (hop) distance.

The upper plot of Figure 16 shows the distribution of such shortest distance in both user and resource networks, where hop-0 refers to the node itself, and hop-∞ includes all the cases of shortest distance above 4 hops. It is noticed that both distributions demonstrate strong locality: typically a majority of closest neighbors belong to the category of hop-0 or hop-1, and the distribution decays as hop increases. Further, the distributions exhibit fairly different patterns in the two networks: the hop-0 category dominates in the social network, while the hop-1 category accounts for the majority in the resource space. This may be explained by that the influence in the resource space is more "dynamic" than the user space, i.e., the tendency that a specific user looks for resources with similar topics is stronger than that he/she is influenced by the interests of other users. Leveraging this observation, in our implementation, we adopt a Normal product network to model such type of influence, while assigning the resource network with higher diffusion rate. In particular, we fix the diffusion rate $\alpha = 0.1$ and 10 for the user and resource networks, respectively. By default, the time-tick is fixed as 10^8 ms.

One questions remains: what are the temporal characteristics of such closest neighboring interactions? To answer this, we measure the temporal distance between the interactions (u,v) and (u',v'), and examine the distribution of the closest interactions along the user and resource dimensions, with results shown in the lower plot of Figure 16. It is clear that (i) both distributions match fairly well, indicating that with high chance the closest neighboring interactions in the two networks are essentially the same one, and (ii) both distributions decay quickly as the temporal distance grows, indicating the strong temporal locality of influence.

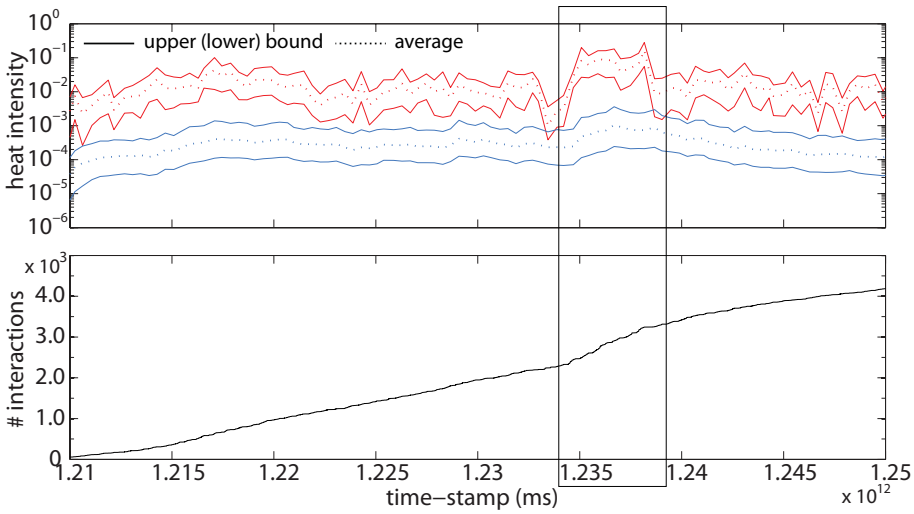


Figure 17: Estimated heat intensity of observed interactions and inactive (background) interactions.

Predictive Power

This set of experiments evaluate the predictive power of our social influence model, i.e., its ability of differentiating interactions highly likely to occur from the rest "background" interactions. We run the tracking process. Specifically at the t-th time-tick, we update the estimation $\mathbf{f}(t)$ of the heat field based on the observed interactions at t, based on which we predict $\mathbf{f}(t+1)$ for the $(t+1)$-th time-tick. We then compare the estimated heat intensity of those interactions actually observed at $(t+1)$ and the rest inactive ones.

The results (mean and deviation) are shown in the upper plot of Figure 17 (in logarithmic scale[10]). It is observed that the two bands are clearly separable, indicating the predictive power of our model: the heat of actually observed interactions is typically two orders of magnitude higher than that of background ones. It is also observed that there exists certain "fluctuation" at the heat estimates (for both observed and inactive interactions) for the time period ($1.234 \sim 1.239 \times 10^{12}$ ms), as highlighted in box. We then examine the overall growth of number of interaction, as shown in the

[10]In following, all the heat intensity measures are in logarithmic scale, unless otherwise noted.

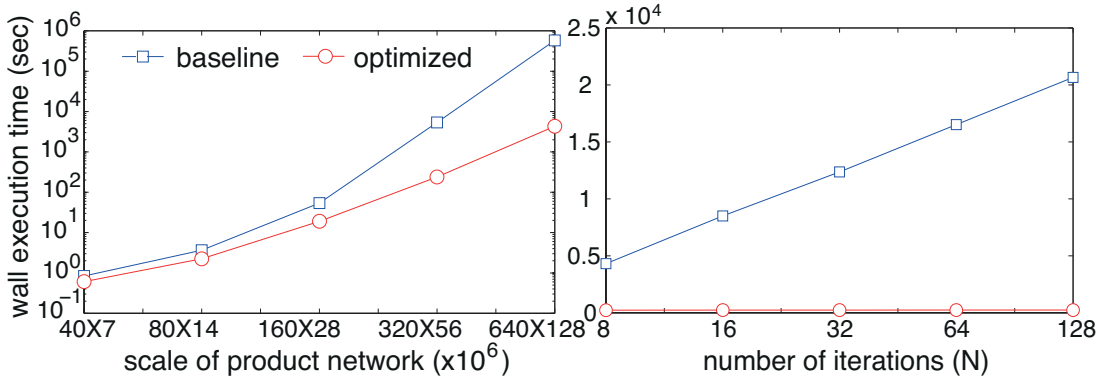


Figure 18: Running time of Track operation with respect to varying network scale and precision setting.

lower plot of Figure 17. It is seen that for the corresponding period, there exists a burst of growth. We therefore conclude that (i) the absolute heat intensity measure is sensitive to the occurrence rate of interactions, which tends to evolve over time, yet (ii) the relative heat intensity (over the average level) may still serve as a good indication of the occurrence likelihood of interactions.

Scalability

This set of experiments are designed to evaluate the operation efficiency of μSI, by measuring the execution time of *Track*, the most fundamental operation of μSI, with respect to varying network scale and precision requirement. More specifically, we apply the *KronFit* tool in Stanford Network Analysis Platform[11] (SNAP) to extract the "cores structures" of user and resource networks, and then apply the *KronGen* tool in SNAP over such cores to generate networks of different scales; also we generate different precision settings by varying the number of iterations required in computing the discrete approximation of matrix exponential. We evaluate both baseline and optimized versions of Track operation (details in Section 3.4), with results shown in Figure 18. As predicted by the theoretical analysis, the optimized Track operation achieves orders of magnitude of performance gain over the baseline implementation,

[11]http://snap.stanford.edu

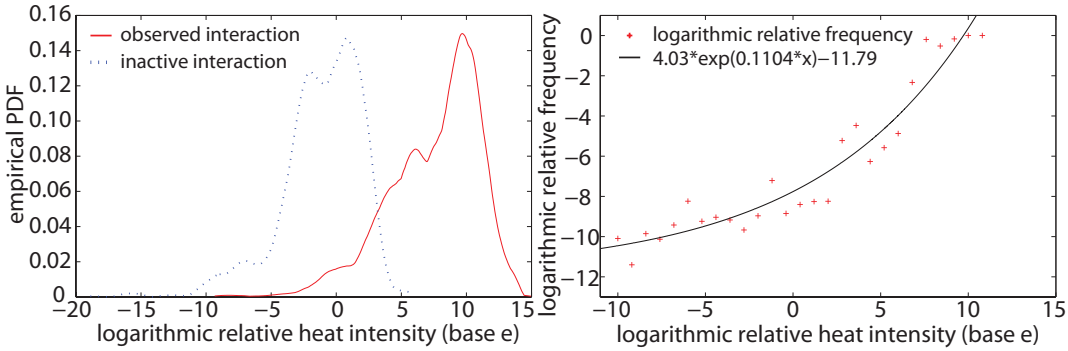


Figure 19: Relative frequency of observed interactions with respect to (relative) heat itensity.

and can easily scale up to product network of scale 8.2×10^{11} or hundreds of iterations.

Prediction Accuracy versus Execution Efficiency

Following the estimation regarding the state of heat field, the next step is to accurately predict the interactions that are highly likely to occur. The invocation model bridges the gap between the two. The concrete form of the invocation model is typically application-dependent; in this set of experiments, we consider one possible construction for the application of resource recommendation.

At each time-tick t, we measure the estimated heat intensity for both observed interactions and inactive ones. We plot the results in Figure 19. The left plot shows the distribution of the heat intensity of both occurred and inactive interactions. Note that the measures here have been normalized to the average heat intensity, i.e., the average level as 0 (in logarithmic scale). One can notice the significant statistical difference between observed and inactive interactions. For each specific heat level (in logarithmic scale), we then measure the relative frequency of observed interactions over the total number of interactions at that level. The right plot of Figure 19 shows the result (in logarithmic scale). We use an exponential curve to fit the relative frequency, which shows a tight match with the data. We can then use this fitted model as our invocation model.

Next we evaluate the *Predict* operation of μSI. Given the current time t and

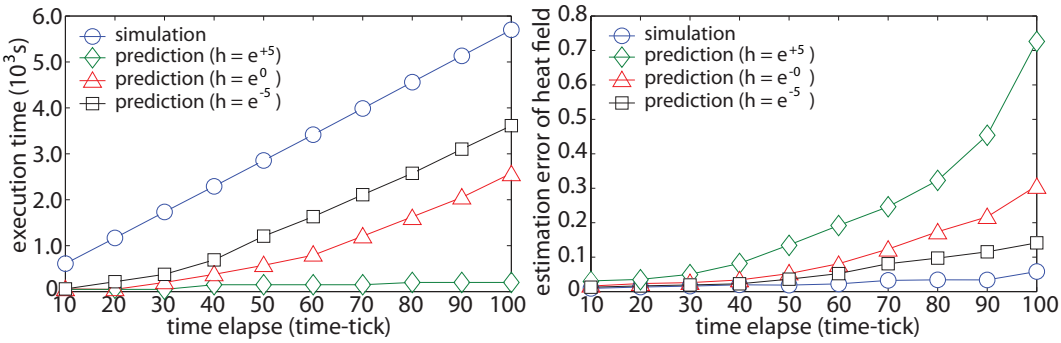


Figure 20: Trade-off between prediction accuracy and execution time.

the elapsed time Δt, to predict the heat field over all the (active or inactive) interactions at $(t + \Delta t)$, the baseline approach is to faithfully simulate the estimating/sampling/updating procedure at each time-tick, which provides the best possible estimation regarding the future state $\mathbf{f}(t + \Delta t)$, featuring linear complexity in terms of Δt. We propose a novel prediction model that trades the prediction accuracy for the execution efficiency. We achieve this by focusing on interactions with higher invocation likelihood (above a threshold h) while ignoring less likely ones.

This prediction accuracy-execution efficiency trade-off is illustrated in Figure 20. Here the accuracy is measured by the cosine distance between the predicted heat field vector $\hat{\mathbf{f}}(t + \Delta t)$ and the tracked result $\mathbf{f}(t + \Delta t)$. We compare the prediction accuracy and execution time for the baseline simulation approach and the lazy-probing approach (with three different settings of threshold h). Clearly, under acceptable accuracy loss (less than 0.1), the lazy-probing approach achieves considerable saving on execution time, e.g., about 37% for $h = e^{-5}$ and 57% for $h = e^0$. However, as h increases, the accuracy loss becomes significant, especially when Δt is large, the error is accumulated at each time-tick. It is thus important to properly tune h to achieve the balance, e.g., $h = e^0$ in the case here.

3.5.3 Case Study 2: Paging Operation in Mobile Phone Call Service

For incoming mobile service requests (e.g., voice, SMS, data), efficiently locating requested mobiles or devices is a critical operation for service providers. This is typically done using a combination of location update (by the mobile) and paging (by the network). The paging operation determines where (i.e., which cells) to search for the mobile given its latest location update. Because paging consumes valuable spectrum and signaling resources and the paging channel is a low bandwidth channel, which can also easily become overwhelmed by denial of service attacks, it is considered critical to decrease the utilization of this signaling channel (i.e., minimize the number of cells to be probed).

Existing work on paging schemes has mostly relied on location management-based approaches (e.g., [125, 147]), i.e., create mobility profiles for mobile users, and predict their current locations based on their latest updates and historical profile. In this chapter, we take a different approach: instead of focusing on the mobility patterns of requested mobiles (callee), we analyze the *combined social and geographical characteristics* of both caller and callee. We then use such characteristics to design smart paging schemes that minimize signaling traffics.

More specifically, we consider two networks: the social network formed by mobile users, and the cell network formed by cellular base stations. A successfully connected mobile phone call essentially comprises two interactions: caller - caller's station and callee - callee's station. Clearly, understanding the characteristics of these two interactions is the key for us to design effective paging schemes.

Characteristics and Feasibility

We start with examining the geographical characteristics of caller' and callee's base stations. We differentiate the communications between friends and that between non-friends. The distribution of calls with respect to the hop distance of base stations is

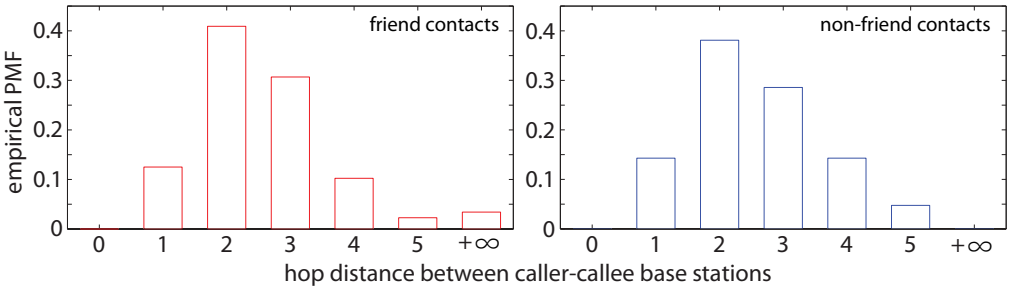


Figure 21: Hop distance between caller's and callee's base stations ($+\infty$ indicates no connection).

illustrated in Figure 21. It is noted that both friend and non-friend contacts show fairly similar patterns: the callers tend to contact the callees geographically close to them (but not too close, e.g., within the same cell), while such likelihood decays almost exponentially as hop distance grows. It seems a natural option to use this geographical characteristics to design paging schemes, e.g., given a call request (caller + caller's station + callee), first search base stations with shorter distance to caller's station.

To validate the feasibility of this scheme, and also contrast our solution against traditional mobility profile-based approaches (e.g., [147]), we perform the following experiments. Assume (i) the base station associated with callee as a random variable X, (ii) the callee's historical location logs as: received M calls with the corresponding base stations as a vector $L = (l_1, l_2, \ldots, l_M)$ (K distinct stations in total), each of these K stations appears x_i times in L, $\sum_{i=1}^{K} x_i = M$. We consider the following five measures: (1) entropy of X, entropy of X conditional on (2) previous $N = 1$ station, (3) previous $N = 2$ stations, (4) caller's station, and (5) caller's identify (phone number) and base station. Taking (1) and (2) as an example, the entropy of X is measured as $H(X) = -\sum_{i=1}^{K}(x_i/M)\log(x_i/M)$, while the conditional entropy of X given previous station Y is defined as $H(X,Y) - H(X)$ where $H(X,Y)$ is the entropy of the joint distribution of two consecutive stations in L. Here, scheme (1), (2), and (3) have been used in the state-of-the-art paging schemes [147], while scheme

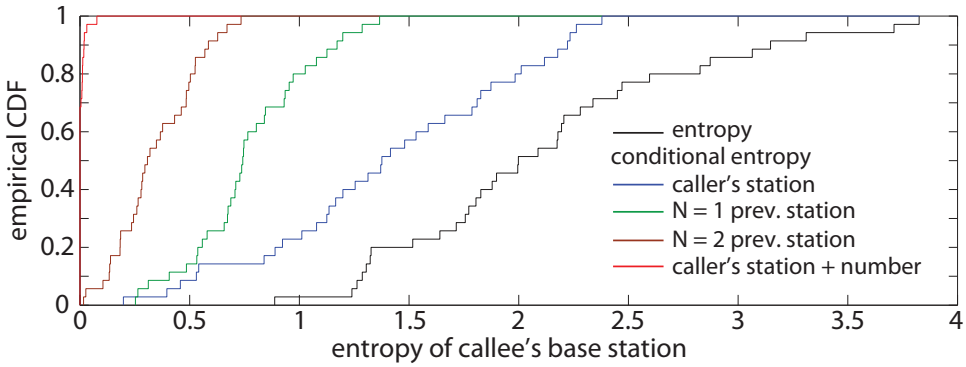


Figure 22: CDF of entropy and conditional entropies of callee's base stations.

(4) corresponds to the aforementioned caller's station-based scheme. The CDFs of these entropy and conditional entropy are plotted in Figure 22. It is observed: first, the information of caller's station alone may not be sufficient to determine callee's station, while caller's station and caller' and callee' identities together may pinpoint callee's station with fairly high chance; second, this scheme may help improve the traditional mobility profile-based solutions.

Success Rate and Paging Cost

Leveraging the observations in Figure 21 and 22, we construct a influence-based paging scheme. We first modify the cell network: for each station, add links to its hop-2 neighbors, and delete links to hop-1 neighbors (observation in Figure 21); we adopt an instantiation of Tensor product of the cell and social networks. For the set of candidate base stations (maybe returned by profile-based approaches), we sort them according to their heat intensity, and probe the base stations according to this order. Figure 23 shows the efficacy of this paging scheme. We construct five schemes, according to the measures discussed in Figure 22. We measure them using two metrics, the success rate of paging operation, which is defined as the probability that the callee's actual base station appears in the "first-search list" of paging scheme (or a broadcast would be invoked), and the cost of paging, which is defined as the number of base stations actually searched (the results are normalized to $[0, 1]$ where 1 indicates a

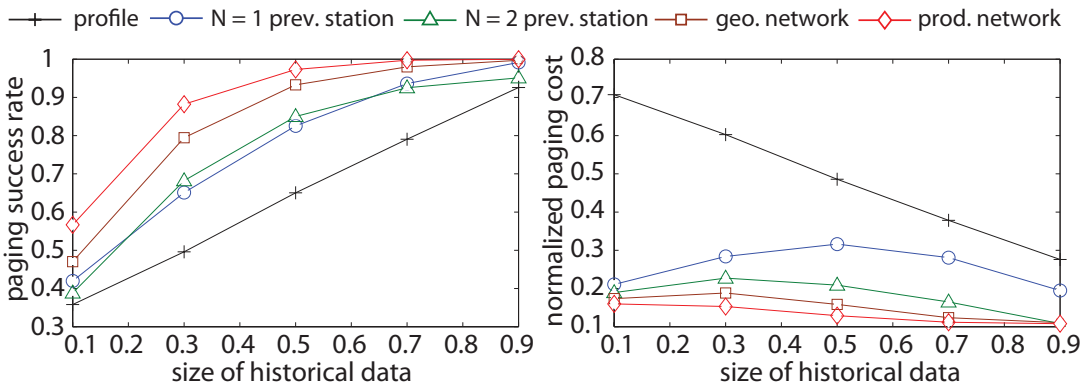


Figure 23: Paging success rate and cost with respect to historical data size.

global broadcast). It is noticed that the influence-based scheme outperforms all the rest in terms of both success rate and paging cost, particularly, when the learning data is limited. This validates our theoretical analysis on its advantage of handling data sparsity issues.

3.6 Other Related Work

Network dynamics has been a long-lasting topic for network science: existing work mostly studied the dynamics inside a single, homogenous network, such as diffusion and propagation [53], influence maximization [75], community formation [13], or network evolution [107]. Recently, intensive research efforts have been directed to social influence in online networks [9, 55, 141]. Using our microscopic model, these works essentially focus on the interactions between one (implicit) object and multiple users.

More recently, a few works have started to account for the rich information conveyed by interactions between multiple objects and users, to detect communities [118], to analyze the evolution of both object and social networks [151], and to model topic-sensitve social influence [95, 120]. In all these cases, however, the relationships between users and objects are modeled as a heterogenous network wherein user-object interactions are treated as static links. This formation, though amenable to graphical model-based learning, completely ignores the dynamic nature of interactions (i.e.,

time-sensitivity), which we deem as one key element in understanding and modeling microscopic social influence.

There are several other lines of work we build upon. In a set of recent works, heat field model on a manifold has been applied for dimension reduction [16], classification [79], spam-resilient ranking [144], and viewpoint-based network analysis [10]. Meanwhile, product network, especially Kronecker product has been applied to model graph generation [86] and dynamic tensor analysis [117]. To the best of our knowledge, the present work is the first that considers the optimization of heat field model in the context of product network.

CHAPTER IV

XCOLOR: PRIVACY-AWARE DATA PUBLISHING

4.1 Introduction

Privacy preservation has become a paramount concern in numerous data dissemination applications that involve private personal information, e.g., medical data and census data. Typically, such *microdata* is stored in a relational table T: each record in T corresponds to an individual; the attributes of T are categorized as either *sensitive* or *non-sensitive*. In the setting of *central publication*, a publisher intends to release an *anonymized* version T^* of the microdata table T, such that no malicious user, called an *attacker*, can infer the sensitive information regarding any individual from T^*, whereas the statistical utility of T is still preserved in T^*.

Towards this end, a bulk of work has been done on anonymized data publication [15, 85, 92, 91, 98, 99, 104, 119, 139, 143, 142, 149]. One of the major aims is to address *association attack*: the attacker possesses the exact non-sensitive (*quasi-identifier* (QI)) values of the victim, and attempts to discover his/her sensitive (SA) value from the published table T^*. A popular methodology of thwarting such attacks is *generalization* [119]: after partitioning the microdata table T into a set of disjoint subsets of tuples, called *QI-group*, generalization transforms the QI-values in each group to a uniform format such that all tuples belonging to the same group are indistinguishable in terms of their QI-values.

Example 3. Consider publishing the medical data as shown in Table 5: *age* and *zip-code* are QI-attributes, while *syndrome* is a composite SA-attribute, each component indicating the severity of a patient's suffering the corresponding symptom. The generalization of the microdata produces two QI-groups, as indicated by the group

Table 5: Anonymized data publication.

	age	zip-code	syndrome			GID
			allergy	asthma	myocarditis	
Alice 1	[18, 30]	[12k, 17k]	**0.8**	**0.0**	**0.0**	1
2	[18, 30]	[12k, 17k]	0.6	0.4	0.4	1
3	[18, 30]	[12k, 17k]	**0.7**	**0.1**	**0.1**	1
4	[18, 30]	[12k, 17k]	1.0	0.2	0.2	1
5	[18, 30]	[12k, 17k]	0.1	0.9	0.9	1
6	[32, 40]	[22k, 30k]	0.2	0.5	0.2	2
7	[32, 40]	[22k, 30k]	0.8	0.1	0.9	2
8	[32, 40]	[22k, 30k]	0.4	0.3	0.5	2
9	[32, 40]	[22k, 30k]	0.6	0.9	0.3	2
10	[32, 40]	[22k, 30k]	1.0	0.7	0.7	2

identifiers (GID). An attacker who knows *Alice*'s QI-values can no longer uniquely identify her SA-value: any tuple in the first group may belong to her; without further information, the attacker can only conclude that *Alice* associates with each specific *syndrome* value with identical probability 20%.

4.1.1 State of the Art

Essentially, the attack above is performed by leveraging the association between quasi-identifier and sensitive attributes (QI-SA association) appearing in the published data. Generalization weakens such association by reducing the representation granularity of QI-values. The protection is sufficient if the weakened association is no longer informative enough for the attacker to infer individuals' SA-values with high confidence. To gauge the quality of protection, a plethora of generalization principles have been proposed, which can be classified according to their targeted types of QI-SA association, namely, *exact* and *proximate* association.

Exact QI-SA association refers to the links between specific QI-values and SA-values in the published data. Exact association is particularly meaningful for publishing categorical sensitive data wherein different values tend to have no sense of

proximity; it is desired to prevent the attacker from linking the victim to a specific SA-value with high confidence. The principles in this category include k-anonymity [119], l-diversity [98], and its variants [139, 143], all with the objective of avoiding uncovering exact SA-values.

Proximate QI-SA association, meanwhile, refers to the links between specific QI-values and a set of proximate SA-values. Clearly, this concept generalizes exact QI-SA association in that it takes account of the semantic proximity among SA-values. After studying the published data, even though with low certainty about the exact value, the attacker might have learned with high confidence that the SA-value of the victim belongs to a set of proximate values. A number of principles, including (k, e)-anonymity [149], variance control [85], and (ϵ, m)-anonymity [91], have been proposed to address such *proximity breach* under the model of one-dimensional numeric sensitive data (different values are strictly ordered).

4.1.2 Motivation

While focusing on tackling proximity breach under specific data models, be it categorical or numeric sensitive data, existing research efforts, however, fail to address the threat for a much richer set of models wherein the semantic proximity might be defined by arbitrarily complex or customized functions. An example is given as follows.

Example 4. Recall the running example of Table 5. Assume that the semantic distance between two SA-values, represented as two vectors $P = \langle p_i \rangle_{i=1}^{n}$ and $Q = \langle q_i \rangle_{i=1}^{n}$, is defined as $\Delta(P, Q) = \min_i |p_i - q_i|$. Measuring the pairwise distance of the *syndrome* values appearing in the first QI-group, one can notice that the first four tuples form a compact "neighborhood" structure, wherein the value of #3 is semantically proximate to that of #1, #2, and #4, as shown in Figure 24.

From the attacker's perspective, every tuple in this group belongs to *Alice* with

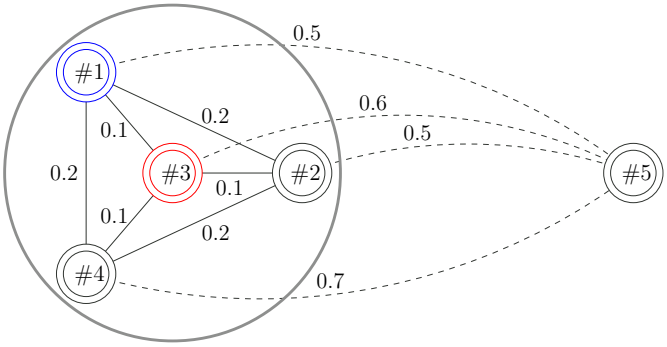


Figure 24: General proximity breach.

equal possibility; she can thus conclude that *Alice* associates with the neighborhood structure with probability 80%. Moreover, she might choose the value of the center node (#3) as an estimation, and arrives at a privacy intruding claim that "*Alice*'s *syndrome* value is fairly close to $(0.7, 0.1, 0.1)$".

Existing privacy principles and definitions, however, are incapable of capturing this general form of proximity breach because of their assumptions regarding the underlying data models. For example, in Table 5, the SA-values in each QI-group are all distinct, thereby satisfying l-diversity [98] with the maximum possible $l = 5$; meanwhile, these multi-dimensional values can not be strictly ordered, thus rendering the techniques developed in [91] inapplicable.

Therefore, in this paper, we advocate studying proximity breach in a data-model-neutral manner, which we refer to as *general proximity breach*, with the objective of providing 1) a better understanding regarding proximity privacy and 2) anonymization solutions of general applicability. We argue that the proximity breaches addressed in the literatures, e.g., *homogeneity breach* [98], are essentially instantiations of this concept. In this paper, we aim at developing effective countermeasures to tackle such general breach.

It is worth contrasting our targeted setting with that of multiple sensitive attributes. We focus our discussion on the case of a single sensitive attribute, which might comprise multiple components (e.g., Table 5), but is associated with a unified

distance metric; while in the setting of multiple sensitive attributes, each attribute might be associated with a different distance metric, and possibly no single measure exists to capture the overall proximity. Following existing practices such as [98], our results can be readily extended to the case of multiple sensitive attributes.

4.1.3 Contributions

To the best of our knowledge, this paper presents the first systematic study on proximity privacy in a data-model-neutral manner, with findings of general applicability.

Concretely, we define QI-SA association under a highly abstract data model, with the only assumption of a semantic distance metric over the domain of the SA-attribute; then, we formalize general proximity breach in the framework of QI-SA association, and address such breach with a unified privacy definition, (ϵ, δ)-*dissimilarity*. It intuitively requires that in each QI-group, every SA-value should be "dissimilar" to a sufficient number of others.

We provide sound theoretical proof that (ϵ, δ)-dissimilarity, used in conjunction with k-anonymity [119] (called $(\epsilon, \delta)^k$-dissimilarity), offers effective protection against association attack, in terms of both exact and proximate QI-SA association. We further show that a number of existing privacy definitions are essentially instantiations of this general principle under the data models that they are designed for.

We conduct an analytical study on the characteristics of $(\epsilon, \delta)^k$-dissimilarity, derive criteria enabling to efficiently test its satisfiability for given microdata, and discuss the optimal setting of the parameters ϵ, δ, and k to achieve the best quality of privacy protection and utility preservation.

Most importantly, we propose a novel anonymization model, XCOLOR, to fulfill $(\epsilon, \delta)^k$-dissimilarity, with guarantees on both operation efficiency and utility preservation, by extending the techniques of *defect graph coloring*. Extensive experiments are conducted over real data to validate the practical performance of XCOLOR.

4.2 Formalization

In this section, we clarify the concept of general proximity breach, and prove the effectiveness of $(\epsilon, \delta)^k$-dissimilarity in remedying such breaches. Furthermore, we discuss the relevance of $(\epsilon, \delta)^k$-dissimilarity to existing privacy principles against proximity breach.

4.2.1 Models and Assumptions

Let T denote a microdata table intended to be published, which consists of d quasi-identifier (QI) attributes $\{A_i^{qi}\}_{i=1}^d$ and a sensitive (SA) attribute A^s. Particularly, 1) A^s can be of arbitrary data type, e.g., categorical, numeric, and customized defined type; 2) a semantic distance metric $\Delta(\cdot, \cdot)$ is defined over the domain of A^s, with $\Delta(x, y)$ denoting the distance between two SA-values x and y.

We begin with formalizing the concept of generalization:

Definition 16 (QI Group/Partition)**.** *The microdata table T is divided into m disjoint subsets of tuples $\mathcal{G}_T = \{G_i\}_{i=1}^m$, which satisfy (i) $\bigcup_{i=1}^m G_i = T$ and (ii) $G_i \cap G_j = \varnothing$ for $i \neq j$. Each G_i is called a QI-group, and $\mathcal{G}_T$ is referred to as a partition of T.*

Definition 17 (Generalization)**.** *Given a partition $\mathcal{G}_T = \{G_i\}_{i=1}^m$, a generalization of T is a table T^* that is obtained by transforming the QI-values in each group G_i to a uniform format, i.e., all tuples in the same QI-group are indistinguishable with respect to their QI-values.*

For a numeric QI-attribute A^{qi}, the generalized value could be the minimum bounding interval of all A^{qi} values in the group; while for a categorical attribute A^{qi}, it could be the lowest common ancestor (LCA) of all A^{qi} values in the group on the domain generalization taxonomy of A^{qi}.

Further, we introduce another fundamental concept underlying the attack model, the neighborhood of a SA-value.

Definition 18 (ϵ-Neighborhood)**.** *In a QI-group G with SA-values as a multi-set $\mathcal{SV}_G = \{v_i\}_{i=1}^n$, the ϵ-neighborhood of a value $v \in \mathcal{SV}_G$, $\Phi_G(v, \epsilon)$, is defined as the subset of $\mathcal{SV}_G$ with their distance to v within ϵ, formally*

$$\Phi_G(v, \epsilon) = \{v' |\ v' \in \mathcal{SV}_G \text{ and } \Delta(v, v') \leq \epsilon\}$$

Example 5. In the running example of Figure 24, given $\epsilon = 0.1$, the ϵ-neighborhood of v_3 consists of $\{v_1, v_2, v_3, v_4\}$.

We proceed to presenting the attack model. The attacker attempts to exploit the generalized table T^* to infer the SA-value $o.A^s$ of a targeted individual o. We assume that she possesses full identification information [99]:

Definition 19 (Background Knowledge)**.** *The attacker possesses the information including (i) the exact QI-values of o, (ii) the QI-group G in T^* which o belongs to, and (iii) the semantic distance metric $\Delta(\cdot, \cdot)$ over A^s.*

By assuming the background knowledge (ii), we are dealing with the worst-case scenario that only one QI-group matches the QI-value of o in the generalized table T^*.

4.2.2 General Proximity Breach

After identifying the QI-group G containing o, the attacker proceeds to estimating $o.A^s$ following a probabilistic model:

Definition 20 (Attack Model)**.** *From the attacker's perspective, every tuple in G belongs to o with identical possibility; therefore, the probability that $o.A^s$ belongs to the ϵ-neighborhood of a SA-value v can be formulated as:*

$$\mathsf{prob}[o.A^s \in \Phi_G(v, \epsilon)] = |\Phi_G(v, \epsilon)|/|G| \tag{3}$$

where $|\Phi_G(v, \epsilon)|$ denotes the cardinality of the neighborhood.

It is clear now that exact QI-SA association is a special case of proximate QI-SA association under the setting of $\epsilon = 0$; for any v, $\mathsf{prob}[o.A^s = v] \leq \mathsf{prob}[o.A^s \in \Phi_G(v, \epsilon)]$ for any $\epsilon > 0$.

Next, we formalize the concept of general proximity breach. Intuitively, if the ϵ-neighborhood $\Phi_G(v, \epsilon)$ encompasses a considerable proportion of the SA-values in G, the attacker can conclude that the victim o is associated with the SA-values appearing in $\Phi_G(v, \epsilon)$ with high probability, though she may not be sure about the exact value. Furthermore, by choosing v as the representative, she can arrive at fairly precise estimation about $o.A^s$, if ϵ is sufficiently small.

To measure the severeness of the privacy threats, and particularly, to capture the impact of proximate SA-values on enhancing the attacker's estimation, we introduce the metric of *proximity risk*.

Definition 21 (Proximity Risk)**.** *Given the neighborhood radius ϵ, the risk of general proximity breach of a QI-group G, $\mathsf{risk}(G, \epsilon)$, is formulated as:*

$$\mathsf{risk}(G, \epsilon) = \max_{v \in \mathcal{SV}_G} \frac{|\Phi_G(v, \epsilon)| - 1}{|G| - 1} \tag{4}$$

Intuitively, $\mathsf{risk}(G, \epsilon)$ measures the relative size of the largest ϵ-neighborhood in G; by excluding v from the neighborhood, it highlights the effect of proximate SA-values on improving the attacker's belief: she priorly associates the victim with each SA-value[1] with identical probability $1/|G|$.

We note that $\mathsf{risk}(G, \epsilon)$ is a real number within the interval $[0, 1]$. In particular, G is free of proximity breach ($\mathsf{risk}(G, \epsilon) = 0$) if all the SA-values are dissimilar, and reaches its maximum ($\mathsf{risk}(G, \epsilon) = 1$) if a SA-value v is proximate to all other SA-values. Specially, we define that $\mathsf{risk}(G, \epsilon) = 1$ for the extreme case of $|G| = 1$.

Furthermore, we define the risk of general proximity breach, $\mathsf{risk}(\mathcal{G}_T, \epsilon)$, for a partition $\mathcal{G}_T$ of the microdata table T, as the maximum risk of all the QI-groups in

[1]We consider the collection of SA-values in a QI-group as a multi-set, and regard each SA-value as unique.

$\mathcal{G}_T$, formally

$$\mathsf{risk}(\mathcal{G}_T, \epsilon) = \max_{G \in \mathcal{G}_T} \mathsf{risk}(G, \epsilon) \tag{5}$$

4.2.3 (Epsilon, Delta)^K-Dissimilarity

To remedy general proximity breach, we propose a novel privacy definition, (ϵ, δ)-*dissimilarity*.

Definition 22 ((ϵ, δ)-Dissimilarity)**.** *A partition $\mathcal{G}_T$ satisfies (ϵ, δ)-dissimilarity if for each $G \in \mathcal{G}_T$, every SA-value v in G has less than $(1-\delta) \cdot (|G|-1)$ ϵ-neighbors.*

Here the parameter ϵ specifies the threshold of semantic proximity; while the parameter δ essentially controls the risk of potential proximity breach.

Next, we prove the effectiveness of this definition against association attack. Concretely, we show that a partition $\mathcal{G}_T$ is free of general proximity breach if and only if it satisfies (ϵ, δ)-dissimilarity. We have the following theorem:

Theorem 8. *Given the microdata table T and the neighborhood radius ϵ, for a partition $\mathcal{G}_T$, $\mathsf{risk}(\mathcal{G}_T, \epsilon) \leq 1-\delta$, if and only if $\mathcal{G}_T$ satisfies (ϵ, δ)-dissimilarity.*

Theorem 8. (Necessity) If the partition $\mathcal{G}_T$ violates (ϵ, δ)-dissimilarity, i.e., $\exists G \in \mathcal{G}_T$, $\exists v \in \mathcal{SV}_G$, $|\Phi_G(v, \epsilon)| - 1 > (1-\delta) \cdot (|G|-1)$, then trivially, $\mathsf{risk}(\mathcal{G}_T, \epsilon) \geq (|\Phi_G(v, \epsilon)| - 1)/(|G|-1) > (1-\delta)$, which implies a proximity breach.

(Sufficiency) If $\mathcal{G}^T$ contains a proximity breach with risk at least $(1-\delta)$, then there must exist certain $G \in \mathcal{G}_T$ and certain $v \in \mathcal{SV}_G$ which violates (ϵ, δ)-dissimilarity.

□

Essentially, (ϵ, δ)-dissimilarity counters general proximity breach via specifying the maximum number of ϵ-neighbors that each SA-value can have, relative to the QI-group size. It captures the impact of proximate SA-values on improving the adversary's estimation, who has a prior belief of $1/|G|$ for each SA-value in G.

Nevertheless, it is insensitive to the trivial case of small-sized QI-groups with pair-wise dissimilar SA-values. In such scenarios, despite the weak proximate QI-SA association, the small cardinality of G offers the attacker with a strong prior belief, $1/|G|$, for each SA-value. To remedy this drawback, we introduce k-anonymity [119] into our framework: by requiring every QI-group to contain at least k tuples, we upper bound this prior belief with $1/k$.

Thus, (ϵ, δ)-dissimilarity, in conjunction with k-anonymity as auxiliary, can effectively thwart association attack in terms of both exact and proximate QI-SA association. We entitle this combination $(\epsilon, \delta)^k$-dissimilarity.

4.2.4 Relevance to Principles in Literatures

It is worth emphasizing again that $(\epsilon, \delta)^k$-dissimilarity makes no specific assumption regarding the underlying data model; hence, it is effective to tackle proximity breach under most existing models. In the following, we show that most generalization principles in literatures are either in-adequate in preventing proximity breach, or essentially the special instantiations of $(\epsilon, \delta)^k$-dissimilarity under the data models which they are designed for.

Principles for categorical data

Motivated by the *homogeneity breach* wherein a majority of tuples in a QI-group share an identical SA-value, l-diversity [98] and its variant (α, k)-anonymity [139] have been proposed to ensure sufficient diversity of SA-values in every QI-group; essentially, they are both special forms of $(\epsilon, \delta)^k$-dissimilarity for data models wherein different SA-values have no sense of semantic proximity, e.g., categorical data.

Let us take (α, k)-anonymity as an example. It combines k-anonymity and l-diversity, and demands that every QI-group must contain at least k tuples, and at most α-percent of these tuples carry an identical SA-value. It is trivial to notice that (α, k)-anonymity is equivalent to $(\epsilon, \delta)^k$-dissimilarity under the setting of $\epsilon = 0$ and

$1 - \delta \approx \alpha$.

Principles for numeric data

For data models wherein different SA-values can be strictly ordered, e.g., one-dimensional numeric data, it qualifies as a several privacy violation if the attacker can identify the victim individual's SA-value within a short interval, even though not the exact value. To address such privacy breach, a plethora of principles have been proposed for publishing numeric sensitive data, e.g., variance control [85] and (k, e)-anonymity [149]. Specifically, variance control specifies that in every QI-group, the variance of the SA-values must be above certain threshold t; (k, e)-anonymity states that every QI-group must have at least k different SA-values, and the difference between the maximum and minimum ones must be at least e. Unfortunately, it is proved in [91] that none of these principles provide sufficient protection against proximity breaches.

The principle most relevant to $(\epsilon, \delta)^k$-dissimilarity is probably (ϵ, m)-anonymity [91]; it demands that in each QI-group G, for every SA-value x in G, at most $1/m$ of the SA-values in G belong to the interval of $[x - \epsilon, x + \epsilon]$. Clearly, (ϵ, m)-anonymity is an instantiation of $(\epsilon, \delta)^k$-dissimilarity for one-dimensional numeric data, with $1/m \approx 1 - \delta$. Nevertheless, targeting a specific data model, the theoretical analysis and generalization algorithms in [91] are inapplicable for addressing general proximity breach. Moreover, since m is an integer, the users can only specify their privacy requirements in a harmonic sequence manner, i.e., $\frac{1}{2}, \frac{1}{3}, \ldots$, instead of a "stepless" continuous adjustment as supported by $(\epsilon, \delta)^k$-dissimilarity.

4.3 Characterization

Next we present an analytical study on the characteristics of $(\epsilon, \delta)^k$-dissimilarity. Specifically, we discuss the satisfiability of $(\epsilon, \delta)^k$-dissimilarity, and expose the implicit trade-off in setting the parameters ϵ and δ.

4.3.1 Satisfiability of (Epsilon, Delta)^k- Dissimilarity

For the given microdata table T and privacy parameters (k, ϵ, δ), the first question arises as "does there exist a partition $\mathcal{G}_T$ for T that satisfies both k-anonymity and (ϵ, δ)-dissimilarity?", i.e., the *satisfiability* of $(\epsilon, \delta)^k$-dissimilarity with respect to T. Unfortunately, in general, no efficient solution exists to answer this question unless P = NP, as shown in the next theorem.

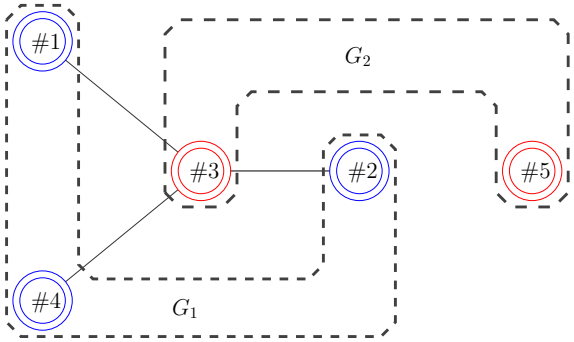


Figure 25: Abstract graph and coloring.

Theorem 9. *In general, for the given microdata table T and parameters (ϵ, δ, k), deciding the satisfiability of $(\epsilon, \delta)^k$-dissimilarity for T is* NP*-hard.*

Before presenting the proof, we first introduce two fundamental concepts, *abstract graph* and *proper coloring*.

Definition 23 (Abstract Graph)**.** *For a microdata table T and a proximity threshold ϵ, the abstract graph $\Psi_T^\epsilon = (\mathcal{V}_T^\epsilon, \mathcal{E}_T^\epsilon)$ is defined as follows: $\mathcal{V}_T^\epsilon$ denotes the set of vertices, with each vertex corresponding to a SA-value in T; $\mathcal{E}_T^\epsilon$ represents the set of edges over $\mathcal{V}_T^\epsilon$, and two vertices are adjacent if and only if their corresponding SA-values are ϵ-neighbors.*

Definition 24 (Proper Coloring)**.** *Given a graph $\Psi = (\mathcal{V}, \mathcal{E})$, a (proper) m-coloring of Ψ is an assignment of no more than m colors to the vertices $\mathcal{V}$, such that no two adjacent vertices share the same color.*

Example 6. Figure 25 illustrates the abstract graph corresponding to the SA-values appearing in the first QI-group of Table 5, under the setting of $\epsilon = 0.1$. One possible 2-coloring scheme is to assign (#1, #2, #4) color 1 and (#3, #5) color 2.

Sketchily, our proof to Theorem 9 is constructed by mapping the satisfiability of $(\epsilon, \delta)^k$-dissimilarity to a proper coloring of the abstract graph corresponding to T and ϵ.

Theorem 9. It suffices to prove that the problem under a specific setting is NP-hard. Let us consider a stringent version of $(\epsilon, \delta)^k$-dissimilarity with $\delta = 1$; that is, all SA-values in a same QI-group are required to be dissimilar.

For the given parameter ϵ and microdata T (of cardinality n), we construct an abstract graph $\Psi_T^\epsilon = (\mathcal{V}_T^\epsilon, \mathcal{E}_T^\epsilon)$. Without loss of generality, consider a partition $\mathcal{G}_T$ of T comprising m ($m \leq \lfloor n/k \rfloor$) QI-groups: $\mathcal{G}_T = \{G_i\}_{i=1}^m$. In Ψ_T^ϵ, the vertices corresponding to each G_i are assigned with one distinct color.

Clearly, under this setting, $\mathcal{G}_T$ satisfies $(\epsilon, \delta)^k$-dissimilarity, only if every two adjacent vertices in $\Psi_T(\epsilon)$ have distinct colors, i.e., a proper m-coloring. Nevertheless, it is known that determining for a general graph if a proper m-coloring exists is NP-complete [49], which implies that deciding the satisfiability of $(\epsilon, \delta)^k$-dissimilarity for given T is NP-hard. □

Therefore, instead of attempting to seek the exact answer to whether an $(\epsilon, \delta)^k$-dissimilarity-satisfying partition exists for given T, we are more interested in developing approximate solutions that can 1) provide explicit and intuitive guidance for the setting of privacy parameters, and 2) efficiently find high-quality partitions of the microdata.

4.3.2 Trade-off between Epsilon and Delta

In our framework, the privacy requirement is specified as a triple of parameters k, ϵ, and δ, which however demonstrate implicit conflicts. In this section, we assume that

k is fixed, and discuss the trade-off between ϵ and δ; in Section 4.5, we reveal the quantitative relationship among ϵ, δ, and k.

Specifically, ϵ specifies the upper bound of semantic distance between two SA-values to be considered as semantically proximate. Meanwhile, δ controls the alarm threshold of proximity breach; a larger δ implies a lower tolerance of proximity-privacy risk. Clearly, by increasing ϵ or δ, one can achieve better privacy protection against association attack, though along different dimensions. An inherent trade-off, however, exists between ϵ and δ. Here, we expose this trade-off in an informal manner, and provide a quantitative modeling in Section 4.4.

Consider two extreme settings of ϵ. 1) $\epsilon = 0$, which amounts to saying that all distinct SA-values are considered dissimilar. By evenly distributing tuples sharing an identical SA-value into different QI-groups, a majority of the SA-values in each QI-group tend to be pairwise dissimilar; therefore, one expects to achieve high δ. 2) $\epsilon = \infty$, which means that all SA-values in the domain are considered similar. In this case, no partition can achieve any $\delta < 1$; that is, the risk of proximity-privacy breach is always 1. Intuitively, as ϵ increases, the number of pairs of similar SA-values grows, rendering it harder to achieve high dissimilarity (large δ in each QI-group), and vise versa.

4.4 Theory

As discussed in Section 4.2, the key to anonymizing a microdata table T through generalization is to determine a partition $\mathcal{G}_T$ of T. It is however shown in Theorem 9 that deciding exactly if T is "anonymizable" for given parameters (k, ϵ, δ) is NP-hard. In this section, we establish the theoretical foundation for an approximate solution that allows intuitive and flexible tuning of the multiple privacy parameters, and finds high-quality partitions with polynomial complexity.

More concretely, we re-formulate the problem of finding an $(\epsilon, \delta)^k$-dissimilarity-satisfying partition in the framework of *defect graph coloring*; we map it to a novel *relaxed equitable coloring* problem that embeds all the privacy parameters. We then conduct an analytical study on the sufficient conditions (in terms of ϵ, δ, and k) for the existence of a valid coloring. The constructive nature of the proofs naturally leads to algorithms that are efficient in both theory and practice.

4.4.1 Problem Re-formulation

It is shown in Section 4.3 that given a microdata table[2] T and a proximity threshold ϵ, one can construct an abstract graph $\Psi^\epsilon = (\mathcal{V}^\epsilon, \mathcal{E}^\epsilon)$. A partition $\mathcal{G}$ of T corresponds to a m-coloring of Ψ^ϵ (may not be proper), which partitions the vertices $\mathcal{V}^\epsilon$ into m *color classes*, defined as below.

Definition 25 (Color Class). *A m-coloring of a graph $\Psi = (\mathcal{V}, \mathcal{E})$ partitions $\mathcal{V}$ into m disjoint subsets (color classes) $\{V_i\}_{i=1}^m$, each corresponding to one distinct color.*

Next, we intend to re-formulate the problem of finding a $(\epsilon, \delta)^k$-dissimilarity-satisfying partition $\mathcal{G}$ in the framework of graph coloring. Sufficiently and necessarily, if a partition $\mathcal{G} = \{G_1\}_{i=1}^m$ of T satisfies $(\epsilon, \delta)^k$-dissimilarity, then there must exist a corresponding coloring of Ψ^ϵ that satisfies the following conditions: 1) Ψ^ϵ is colored using m colors ($\{V_i\}_{i=1}^m$ represent the m color classes); 2) the size of every color class is at least k, i.e., $|V_i| \geq k$ $(1 \leq i \leq m)$; and 3) for any $v \in V_i$ $(1 \leq i \leq m)$, at most $(1 - \delta) \cdot (|V_i| - 1)$ vertices in V_i are adjacent to v.

We note that the coloring problem above can be considered as a "relaxed" version of the classic proper coloring (Definition 24), in the sense that it allows a constrained

[2]Without ambiguity, in the rest of the paper, we omit this referred microdata table in the notations.

number of monochromatic edges (called *defects*). It however deviates from the conventional setting of *defect coloring* as studied in graph theory, e.g., [32], in the sense that it imposes constraints on the size of every color class.

Therefore, in developing our solution, we target the following *relaxed equitable coloring* problem.

Definition 26 (Relaxed Equitable $(\lfloor \frac{n}{k} \rfloor, \delta)$-Coloring)**.** *We define a relaxed equitable* $(\lfloor \frac{n}{k} \rfloor, \delta)$*-coloring of a graph* Ψ^ϵ *as the following conditions:*

(i) Ψ^ϵ *is colored using* $m = \lfloor \frac{n}{k} \rfloor$ *colors, with the corresponding color classes denoted by* $\{V_i\}_{i=1}^m$*;*

(ii) the sizes of any two color classes differ by at most 1;

(iii) for any $v \in V_i$ $(1 \leq i \leq m)$*, at most* $\lfloor (1-\delta) \cdot (|V_i|-1) \rfloor$ *vertices in* V_i *are adjacent to* v*.*

Clearly, this coloring scheme incorporates both k-anonymity (condition (ii)) and (ϵ, δ)-dissimilarity (condition (iii)). Note that for ease of presentation, here we limit the size of every color class to be either k or $k+1$ (i.e., equitable coloring); our results however can be readily extended to support different group sizes.

To the best of our knowledge, there is no previous study on such relaxed equitable coloring problem; therefore, the solution presented here is interesting in its own right from the perspective of graph theory.

4.4.2 Rationale

Following, we present the theoretical rationale of our equitable $(\lfloor \frac{n}{k} \rfloor, \delta)$-coloring scheme. We begin with introducing the fundamental concepts. The complete list of notations used in the presentation can be found in Table 6.

Among the properties of Ψ^ϵ, we are particularly interested in its maximum degree, which is formally defined as below.

Table 6: List of symbols and notations.

notation	definition
ϵ	threshold of semantic proximity
δ	threshold of breach
k	parameter of k-anonymity
n	cardinality of microdata table T
m	number of color classes $\lfloor n/k \rfloor$
t	$\lfloor (1-\delta)\cdot(k-1) \rfloor$
$\mathcal{G}_T^\epsilon$	abstract graph for given ϵ and T
Θ_T^ϵ	maximum degree of $\mathcal{G}_T^\epsilon$
g	lower bound of the number of movable classes

Definition 27 (Maximum Degree)**.** *The maximum degree Θ^ϵ of a graph Ψ^ϵ is defined as $\Theta^\epsilon = \max_{v\in\mathcal{V}^\epsilon} \mathcal{D}_{\Psi^\epsilon}(v)$, where $\mathcal{D}_{\Psi^\epsilon}(v)$ denotes the degree of v in Ψ^ϵ.*

For the sake of clarity, we assume that n is divisible by k; thus, every color class is of identical size k. We use $m = \frac{n}{k}$ to denote the number of color classes, and $t = \lfloor (1-\delta)\cdot(k-1) \rfloor$ to represent the maximum number of neighbors that a vertex is allowed to have in its self-colored class.

Let $\mathcal{D}_V(v)$ denote the number of neighbors of a vertex v in a color class V. Clearly, if v has overlarge $\mathcal{D}_V(v)$ in its self-colored class V, it violates (ϵ,δ)-dissimilarity, formally

Definition 28 (Violation/Movable Class)**.** *Given a color class V and a vertex v, if $v \in V$ and $\mathcal{D}_V(v) \geq (t+1)$, v is called a violation; if $v \notin V$ and $\mathcal{D}_{V'}(v) \leq t$, V is called a movable class for v, or v is movable to V.*

We have the following lemma that establishes a lower bound on the number of movable classes for any v.

Lemma 1. *Given a graph $\Psi^\epsilon = (\mathcal{V}^\epsilon, \mathcal{E}^\epsilon)$, and a m-coloring of Ψ^ϵ, $\mathcal{C} = \{V_i\}_{i=1}^m$, for any $v \in \mathcal{V}^\epsilon$, at least $g = m - \lfloor \Theta^\epsilon/(t+1) \rfloor$ color classes $V \in \mathcal{C}$ satisfy $\mathcal{D}_V(v) \leq t$.*

Lemma 1. Summing the degrees of v over all the color classes, we have $\sum_{i=1}^m \mathcal{D}_{V_i}(v) \leq$

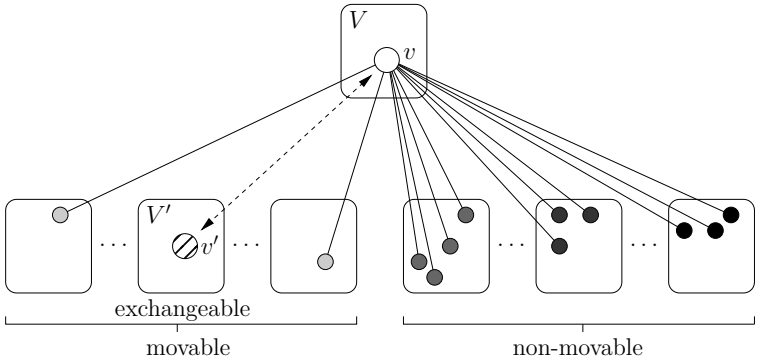


Figure 26: Movable-classes and exchangeable classes.

Θ^{ϵ}. According to the *pigeon-hole* principle, one can derive that at most $\lfloor \Theta^{\epsilon}/(t+1) \rfloor$ classes contain more than t neighbors of v, from which follows this lemma. □

Assume that an initial coloring $\mathcal{C} = \{V_i\}_{i=1}^{m}$ violates (ϵ, δ)-dissimilarity. The key idea of transforming $\mathcal{C}$ to an (ϵ, δ)-dissimilarity-satisfying coloring $\mathcal{C}' = \{V_i'\}_{i=1}^{m}$ is to move every violation $v \in V$ $(V \in \mathcal{C})$ to a movable class V' with $\mathcal{D}_{V'}(v) \leq t$. In order to satisfy the requirement that all color classes are of identical size, it is necessary to move a vertex $v' \in V'$ back to V, i.e., v and v' are exchanged. The movement of adding v' to V, however, could potentially create more violations in V, thereby making this transformation process never converge.

To remedy this, we introduce the concept of *global potential* as an indication of the convergence of this process. Specifically, the global potential of a coloring with respect to a graph is defined as the total number of monochromatic edges. Informally, if every move of the transformation makes the potential decrease, the process will converge in polynomial time (the maximum possible potential of a coloring with respect to Ψ^{ϵ} is $|\mathcal{E}^{\epsilon}|$). Now, we proceed to formulating the impact of each move over the global potential.

Definition 29 (Potential Change)**.** *The change in potential resulted from moving a vertex v from color class V to V', $\Lambda(V \xrightarrow{v} V')$, is calculated as $\Lambda(V \xrightarrow{v} V') = \mathcal{D}_{V'}(v) - \mathcal{D}_{V}(v)$, i.e., the change in the number of monochromatic edges.*

We demand that the move of switching two vertices $v \in V$ and $v' \in V'$ is allowed only if the global potential is decreased, formally

Definition 30 (Exchangeable Class)**.** *A vertex v is exchangeable to a color class V' (or V' is an exchangeable class for v) only if (i) v is movable to V', and (ii) $\exists v' \in V'$, $\Lambda(V \xrightarrow{v} V') + \Lambda(V' \cup \{v\} \xrightarrow{v'} V \setminus \{v\}) < 0$.*

This scenario is illustrated in Figure 26: among the family of movable classes exists an exchangeable class V' that contains a vertex v' such that switching v and v' results in the decrease of the global potential.

We are thus interested in investigating the existence of exchangeable class among the family of movable classes for v. In the following lemma, we show that if the maximum degree Θ^ϵ is bounded by certain threshold, there must exist at least one exchangeable class for v.

Lemma 2. *If $\Theta^\epsilon \leq \frac{m\cdot(t+1)}{2}$, for an arbitrary coloring $\mathcal{C} = \{V_i\}_{i=1}^m$ and any $v \in V$ with $\mathcal{D}_V(v) \geq (t+1)$, there exists at least one exchangeable class V' for v.*

Lemma 2. Otherwise, assume that all the color classes of $\mathcal{C}$ are non-exchangeable for v. Consider the family of movable classes for v. Without loss of generality, assume that $\{V_i\}_{i=1}^g$ are movable for v. Applying the assumption of $\mathcal{D}_V(v) \geq (t+1)$, we have the following two facts:

1) $\sum_{i=1}^g \mathcal{D}_{V_i}(v) \leq \Theta^\epsilon - (m-g)\cdot(t+1)$, derived from Definition 28;

2) $\Lambda(V \xrightarrow{v} V_i) \leq \mathcal{D}_{V_i}(v) - (t+1)$ $(1 \leq i \leq g)$, derived from Definition 29.

According to the assumption, none of the movable classes are exchangeable for v; therefore, any vertex $v' \in V_i$ $(1 \leq i \leq g)$ should satisfy the next condition:

$$\Lambda(V_i \cup \{v\} \xrightarrow{v'} V \setminus \{v\}) \geq -\Lambda(V \xrightarrow{v} V_i)$$

We thus have the following inequality:

$$\begin{aligned}
\mathcal{D}_{V\setminus\{v\}}(v') &= \Lambda(V_i \cup \{v\} \xrightarrow{v'} V \setminus \{v\}) + \mathcal{D}_{V_i\cup\{v\}}(v') \\
&\geq \Lambda(V_i \cup \{v\} \xrightarrow{v'} V \setminus \{v\}) \\
&\geq -\Lambda(V \xrightarrow{v} V_i) \\
&\geq (t+1) - \mathcal{D}_{V_i}(v)
\end{aligned}$$

Summing $\mathcal{D}_{V\setminus\{v\}}(v')$ over all the vertices in the family of movable classes $\{V_i\}_{i=1}^{g}$ for v, we can obtain

$$\begin{aligned}
\sum_{i=1}^{g}\sum_{v'\in V_i} \mathcal{D}_{V\setminus\{v\}}(v') &\geq \sum_{i=1}^{g}\sum_{v'\in V_i} [(t+1) - \mathcal{D}_{V_i}(v)] \\
&= g\cdot k\cdot(t+1) - k\sum_{i=1}^{g}\mathcal{D}_{V_i}(v) \\
&\geq g\cdot k\cdot(t+1) \\
&\quad -k\cdot[\Theta^{\epsilon} - (m-g)\cdot(t+1)] \\
&= m\cdot k\cdot(t+1) - k\cdot\Theta^{\epsilon} \\
&\geq k\cdot\Theta^{\epsilon}
\end{aligned}$$

It is thus derived that the maximum degree of the vertices in $V \setminus \{v\}$ is at least $\frac{\sum_{i=1}^{g}\sum_{v'\in V_i}\mathcal{D}_{V\setminus\{v\}}(v')}{k-1} > \Theta^{\epsilon}$, which is a contradiction to the maximality of Θ^{ϵ}. □

Based on Lemma 2, we are ready to introduce the following theorem, which can be considered as a significant extension of the classic result of Lovász [96] along the dimension of equitable coloring.

Theorem 10. *For given m, a graph $\Psi = (\mathcal{V}, \mathcal{E})$ with maximum degree Θ can be equitably colored using m colors, with each color class of degree at most $(\frac{2\Theta}{m} - 1)$, in time $O(|\mathcal{E}|\cdot|\mathcal{V}|)$.*

Theorem 10. Start with an arbitrary initial coloring wherein all the color classes are of identical size. Consider a vertex v in a class V with more than $(\frac{2\Theta}{m} - 1)$ self-colored

neighbors. As proved in Lemma 2, there must exist at least one vertex v' in an exchangeable class V' for v such that switching v and v' decreases the potential of the graph. We exchange the colors of v and v', thereby decreasing the overall number of monochromatic edges in the graph by at least 1. Repeat this process until all the violations are removed. This takes at most $|\mathcal{E}|$ steps, with the cost of each step at most $|\mathcal{V}|$, leading to the overall complexity of $O(|\mathcal{E}| \cdot |\mathcal{V}|)$. □

4.5 XCOLOR *Algorithm*

Theorem 10 paves the way for designing efficient solutions to fulfill $(\epsilon, \delta)^k$-dissimilarity. In this section, we bridge the gap between the theoretical methodology and the practical generalization algorithm.

Concretely, we aim at addressing three key challenges. 1) The setting of the privacy parameters ϵ, δ, and k. As exposed in Section 4.3, their inherent conflicts make it necessary to carefully balance these parameters in order to achieve the best quality of privacy protection. 2) The utility of the anonymized data. So far we have been solely focusing on providing adequate protection against proximity breach; while in real applications, it is imperative to take account of the resulted data utility when designing the anonymization algorithm. 3) The efficiency of the algorithm. Although it is difficult to further improve the asymptotic complexity of the basic algorithm in Theorem 10, one can construct a high-quality initial coloring that leads to significant performance gains over unattended initial configurations.

Following, we first reveal how to set the privacy parameters to achieve the best quality of protection, then show how to optimize the data utility along the process of anonymization, and finally discuss how to speed up the anonymization process.

4.5.1 Setting of K, Epsilon, and Delta

Within our framework, the privacy requirement is specified as a triple of parameters k, ϵ, and δ. Intuitively, k represents the lower bound of the QI-group size, thereby

guaranteeing that an attacker is unable to associate a victim with a single SA-value with high confidence; ϵ indicates the threshold of semantic proximity, and two SA-values with distance below ϵ are considered as semantically similar; δ denotes the alarm threshold of proximity breach, and any breach with risk above $(1-\delta)$ needs to be eliminated.

Clearly, increasing k, ϵ, or δ all improves the quality of protection, but along different dimensions. The adjustment, however, is not arbitrary, as constrained by their inherent trade-offs. Our framework provides an explicit and quantitative modeling of such trade-offs using the following inequality (derived from Lemma 2).

$$\Theta^{\epsilon} \leq \frac{n}{2} \cdot [1 - (1 - \frac{1}{k}) \cdot \delta] \tag{6}$$

At the first glance, it may seem that the parameter ϵ is not reflected in this inequality. we note, however, that for given microdata T, there is a surjective mapping from ϵ to the abstract graph Ψ^{ϵ}; that is, ϵ uniquely determines the structure of Ψ^{ϵ}. In particular, the density of Ψ^{ϵ} is directly correlated with ϵ; therefore, the maximum degree Θ^{ϵ} is a proper indicator of the underlying parameter ϵ: a larger ϵ implies a denser Ψ^{ϵ} and thus a higher maximum degree.

Different users tend to possess varying preferences for the importance of these three parameters. Below, using the case of fixed ϵ as an example, we show how to set δ and k to achieve the optimal protection. Note that stricter privacy protection is obtained at the cost of reduced data utility. Here, we concentrate our discussion on the maximum achievable protection, and temporarily ignore the utility issue.

For fixed ϵ (fixed Θ^{ϵ}), one can increase both δ and k when Inequality 6 holds. When it reaches the bottleneck, i.e., $(1 - 1/k) \cdot \delta = (1 - 2\Theta^{\epsilon}/n)$, depending on the user's preference, one can trade k (or δ) in order to increase δ (or k) further.

4.5.2 Incorporation of Data Utility

A key consideration missing in the basic algorithm derived from Theorem 10 is the utility of the resulted anonymized data. We intend to incorporate the optimization of data utility in the anonymization process.

As discussed in Section 4.2, generalization transforms the QI-values in each QI-group to a uniform format. Evidently, this operation is performed at the cost of information loss; and the objective therefore is to minimize the global information loss in the process of anonymization.

Various metrics (a detailed survey in [77]) have been proposed to measure the information loss incurred by the generalization operation. Although our framework makes no specific assumption regarding the metrics in use, in our experiments, we employ the following model [91, 142, 143] to take the gauge of the information loss for a QI-group G:

$$\mathsf{loss}(G) = \sum_{i=1}^{d} \frac{|G.A_i^{qi}|}{|A_i^{qi}|}$$

where $|A_i^{qi}|$ represents the domain length of an QI-attribute A_i^{qi}, $|G.A_i|$ represents the length of the generalized QI-value of G, and d is the number of QI-attributes. Note that since all the QI-groups are of identical size, we omit the factor of group size in this model.

In our framework, we adopt a greedy strategy to minimize the global information loss. Specifically, when exchanging a vertex $v \in V$ with another one $v' \in V'$, the change of information loss can be formulated as follows:

$$\begin{aligned} \mathsf{loss}'(V \overset{v,v'}{\longleftrightarrow} V') \;=\; & \mathsf{loss}(V \setminus \{v\} \cup \{v'\}) \\ & + \mathsf{loss}(V' \setminus \{v'\} \cup \{v\}) \\ & - \mathsf{loss}(V) - \mathsf{loss}(V') \end{aligned}$$

We seek the pair of vertices $(\mathring{v}, \mathring{v}')$ for exchange that lead to the maximum decrease

in the overall information loss, formally

$$(\mathring{v}, \mathring{v}') = \arg\min_{v,v'} \mathsf{loss}'(V \overset{v,v'}{\longleftrightarrow} V')$$

At the same time instance, however, a significant number of vertices might all violate (ϵ, δ)-dissimilarity, and each might correspond to a considerable number of exchangeable vertices, resulting in the prohibitive complexity of finding the optimal pair $(\mathring{v}, \mathring{v}')$. In our implementation, we make further approximation by dividing this operation into two steps: in the first step, we find the vertex $\mathring{v}$ (in a class V) whose departure minimizes the information loss of V, formally

$$\mathring{v} = \arg\min_{v} \mathsf{loss}(V \setminus \{v\}) - \mathsf{loss}(V)$$

In the second step, we seek the vertex $\mathring{v}'$ (in a class V') whose exchange with $\mathring{v}$ minimizes the overall information loss of V and V', formally

$$\begin{aligned}\mathring{v}' = \arg\min_{v'} \quad & \mathsf{loss}(V \setminus \{\mathring{v}\} \cup \{v'\}) + \mathsf{loss}(V' \cup \{\mathring{v}\} \setminus \{v'\}) \\ & - \mathsf{loss}(V) - \mathsf{loss}(V')\end{aligned}$$

Assuming that at the time instance there are n_v instances of v, each corresponding to $n_{v'}$ instances of v', the approximation here reduces the complexity from $O(n_v \cdot n_{v'})$ to $O(n_v + n_{v'})$.

4.5.3 Optimization of Initial Coloring

As proved in Theorem 10, for an arbitrary initial configuration, the basic algorithm can converge to an equitable (m, δ)-coloring with time complexity of $O(|\mathcal{V}| \cdot |\mathcal{E}|)$, where the term $|\mathcal{E}|$ follows the upper bound of the number of monochromatic edges. We deem it as the key of decreasing the computational complexity to minimize the initial number of monochromatic edges. The following lemma provides the rationale for our initial configuration construction procedure.

Lemma 3. *Assume that the probability that two vertices are adjacent is proportional to the product of their degrees. The overall number of monochromatic edges is minimized if all color classes have identical sum of degrees.*

Lemma 3. Let $\{d_i\}_{i=1}^n$ denote the degrees of the vertices $\{v_i\}_{i=1}^n$, correspondingly, and $\mathsf{edge}(v_i, v_j)$ be the probability that two vertices v_i and v_j share an edge. According to the assumption, we have $\mathsf{edge}(v_i, v_j) \propto d_i \cdot d_j$.

We intend to partition the vertices into m color classes of identical size k, $V_c = \{v_i^c\}_{i=1}^k$ $(1 \leq c \leq m)$, such that the overall number of monochromatic edges is minimized:

$$\min \sum_{c=1}^{m} \sum_{v_i^c, v_j^c \epsilon V_c} \mathsf{edge}(v_i^c, v_j^c)$$

Given the assumption that $\mathsf{edge}(v_i^c, v_j^c) \propto d_i^c \cdot d_j^c$, one can derive that this problem is equivalent to minimizing the following simplified version.

$$\min \sum_{c=1}^{m} \left(\sum_{i=1}^{k} d_i^c \right)^2 - \sum_{i=1}^{n} d_i^2$$

For fixed $\{d_i\}_{i=1}^n$, we can omit $\sum_{i=1}^n d_i^2$. Now, let X_c denote $\sum_{i=1}^k d_i^c$. We have the following equivalent formulation.

$$\min \sum_{c=1}^{m} X_c^2 \quad s.t. \quad \sum_{c=1}^{m} X_c = \sum_{i=1}^{n} d_i$$

It is well known that under this setting the minimum is achieved when $X_i = X_j$ for every pair of i and j. □

Unfortunately, reductible from the *equal-subset-sum* problem [138], constructing an initial configuration with all the color classes of identical sum of degrees is NP-hard.

Following, we present a heuristic solution to constructing the initial coloring, which is empirically proved to lead to fairly small number of monochromatic edges. Algorithm 5 outlines the INITIALIZE procedure. It takes as input a graph Ψ (of cardinality

Algorithm 5: INITIALIZE (Ψ, k)

Input: graph $\Psi = (\mathcal{V}, \mathcal{E})$, k
Output: m color classes ($m = \lfloor n/k \rfloor$, $n = |\mathcal{V}|$)

1 sort $\{v | v \in \mathcal{V}\}$ in descending order of degrees as $v_1, v_2, \ldots, v_n$ (with degrees $d_1, d_2, \ldots, d_n$, respectively);
2 initialize m empty buckets $\{V_j\}_{j=1}^{m}$;
3 **for** *i from* 1 *to* $k \cdot m$ **do**
4 $\quad j^* \leftarrow \arg\min_j d(V_j)$ s.t. $h(V_j) < k$;
5 $\quad$ add v_i to V_{j^*};
// process the remaining vertices
6 **for** *i from* $(k \cdot m + 1)$ *to* n **do**
7 $\quad j^* \leftarrow \arg\min_j d(V_j)$ s.t. $h(V_j) < k + 1$;
8 $\quad$ add v_i to V_{j^*};
9 return $\{V_j\}_{j=1}^{m}$

n) and a parameter k, and generates a coloring of Ψ with m color classes ($m = \lfloor n/k \rfloor$). For each class V_j, it keeps track of its sum of degrees $d(V_j)$ and its height $h(V_j)$ (the number of vertices assigned to V_j). INITIALIZE balances $d(V_j)$ ($1 \leq j \leq m$) in a greedy manner: at each iteration, from the pool of unassigned vertices, it picks the one with the maximum degree, and adds it to a non-full class with the minimum sum of degrees (lines 3-5). After all the classes are filled with k vertices, it assigns remaining vertices (if n is not divisible by k) to classes with the minimum sum of degrees (line 6-8).

4.5.4 A Complete Framework

Incorporating the suite of multi-folded optimizations into the basic algorithm derived from Theorem 10, we are now ready to present the complete anonymization framework, XCOLOR, as sketched in Algorithm 6. Given the microdata table T and the parameters ϵ, δ, and k, XCOLOR produces an anonymized table T^* that satisfies $(\epsilon, \delta)^k$-dissimilarity.

Specifically, for the given T and ϵ, XCOLOR first constructs an abstract graph Ψ_T^ϵ (line 2), and invokes INITIALIZE to obtain an initial coloring of Ψ_T^ϵ (line 3). It then identifies and eliminates all the violations of (ϵ, δ)-dissimilarity (line 4-11): at each

iteration, XCOLOR switches a violation $\mathring{v}$ (in class V) with an exchangeable vertex $\mathring{v}'$ (in class V'), meanwhile minimizing the information loss. Finally, the coloring is mapped to a partition of the microdata table T (line 12); and by generalizing all the corresponding QI-groups, an anonymized table T^* is obtained (line 13).

Algorithm 6: XCOLOR (T, k, ϵ, δ)

Input: microdata table T, parameters k, ϵ, δ
Output: anonymized table T^*

1 $n = |T|$, $m = \lfloor n/k \rfloor$, $t = \lfloor (1-\delta) \cdot (k-1) \rfloor$;
2 construct an abstract graph $\Psi_T^\epsilon = (\mathcal{V}_T^\epsilon, \mathcal{E}_T^\epsilon)$;
`// create an initial coloring`
3 $\{V_i\}_{i=1}^m$ = INITIALIZE (Ψ_T^ϵ, k);
`// vertices exchanges`
4 $\mathcal{V} = \{v | v \in V_i$ and $\mathcal{D}_{V_i}(v) \geq t+1$ $(1 \leq i \leq m)\}$;
5 **while** $\mathcal{V} \neq \varnothing$ **do**
6 find $\mathring{v} = \min_{v \in \mathcal{V}} \mathsf{loss}(V \setminus \{v\}) - \mathsf{loss}(V)$;
7 $\mathcal{V}' \leftarrow$ exchangeable vertices for $\mathring{v}$;
8 find $\mathring{v}' = \min_{v' \in \mathcal{V}'} \mathsf{loss}(V \setminus \{\mathring{v}\} \cup \{v'\})$
9 $+\mathsf{loss}(V' \cup \{\mathring{v}\} \setminus \{v'\}) - \mathsf{loss}(V) - \mathsf{loss}(V')$;
10 switch $\mathring{v}$ and $\mathring{v}'$;
11 update $\mathcal{V}$;
`// map to partition`
12 map $\{V_i\}_{i=1}^m$ to a partition $\mathcal{G}_T = \{G_i\}_{i=1}^m$;
13 generalize G_i $(1 \leq i \leq m)$ and return T^*;

4.6 Experiments

In this section, we perform an empirical evaluation to validate the analytical models and the efficacy of the proposed countermeasure. The experiments comprise two main parts: 1) we intend to study the impact of general proximity breach over the anonymized data generated according to alternative privacy definitions; 2) we aim to investigate the practical performance of the XCOLOR method, in terms of proximity-privacy protection, utility preservation and operation efficiency.

4.6.1 Experimental Setting

Our experiments use a real dataset SAL (http://ipums.org), which has now become a de facto benchmark for evaluating anonymized data publishing algorithms, e.g., [91, 143, 142]. The dataset contains 50k valid tuples, each corresponding to the personal information of an American adult, collected from the US census. The attributes used in the experiments, their domain lengths (the number of distinct values), and the heights of their domain generalization taxonomies (for categorical attributes) are listed in Table 7.

Table 7: Attributes of the SAL dataset.

attribute	**type**	**d. length**	**d. height**
Age	numeric	85	N/A
Sex	categoric	2	2
Marital Status	categoric	6	2
Race	categoric	9	2
Education	sensitive	17	N/A
Work Class	sensitive	10	4
Income	sensitive	50	N/A

We use the first four attributes as QI-attributes, and regard the last three as a composite sensitive attribute. The semantic proximity between two sensitive values is defined as their normalized L^p space distance (within the interval [0,1]); the weight of each component in the attribute is set according to a health report [137] on the influence of education, income, and occupation to cardiovascular disease.

All the algorithms are implemented in C++, and the experiments are conducted on a Linux workstation running 1.7GHz Pentium III and 2GB memory.

4.6.2 Experimental Results

Attack Vulnerability

In the first set of experiments, we intend to evaluate the impact of general proximity breach over the "publishable" data generated according to alternative privacy definitions. Specifically, we deploy an implementation of Mondrian [84], a state-of-the-art anonymizaton algorithm, to generate l-diverse tables. To show that even strong l-diversity does not guarantee sufficient protection against general proximity breach, we fix the value of l to be 20, which implies that in every QI-group, no more than 5% tuples can share identical sensitive values.

We then measure the risk of proximity breach (Definition 21) for the published data by counting the number of QI-groups which violate (ϵ, δ)-dissimilarity. Given a generalized table, we define its vulnerability to association attack as the proportion of QI-groups that contain proximity breach over the total number of QI-groups in the table. Moreover, we measure the vulnerability under varying settings of semantic proximity metric, i.e., L^1- and L^2-norm.

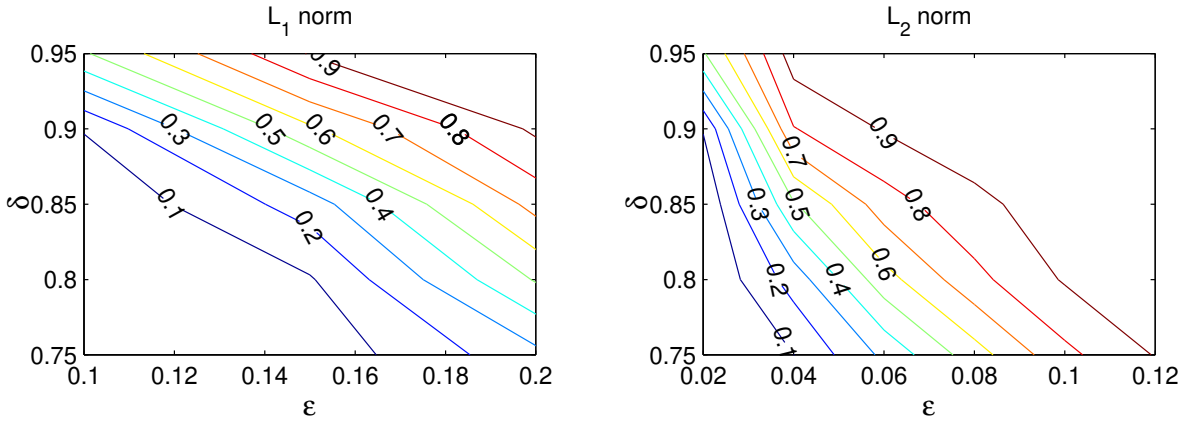


Figure 27: Vulnerability of the published table (generated by Mondrian with $l = 20$) to general proximity breach.

Figure 27 gives a contour view of the vulnerability (with respect to ϵ and δ) of the

published table generated by Mondrian with $l = 20$. The left and right plots correspond to L^1- and L^2-norm, respectively. It is clear that in both cases the vulnerability increases significantly as ϵ or δ grows. For example, in the left plot (L^1-norm), for any $\epsilon \geq 0.18$ and $\delta \geq 0.85$, over half of the QI-groups are subjected to general proximity breach. One can also notice the "mutual enforcement" of the two parameters ϵ and δ with respect to the vulnerability: for a large ϵ (or δ), a trivial increase in δ (or ϵ) leads to a considerable growth of the attack vulnerability.

Because of the similar characteristics of L^p-norms ($p = 1, 2$ in the example above), in the following, we primarily use L^1-norm as the semantic proximity metric.

Data Quality

We measure the utility of the generalized data using the approach described in [91, 143, 142]. Consider the *count queries* of the form

$$\texttt{select count (*) from } \text{generalized-data}$$
$$\texttt{where } A_1 \in I_1 \texttt{ and } A_2 \in I_2 \texttt{ and } \ldots \texttt{ and } A_q \in I_q$$

where $A_1, \ldots, A_q$ are q distinct random attributes, comprising qd QI-attributes and qs SA-attributes ($qd + qs = q$), and each I_i ($1 \leq i \leq q$) is a randomly selected interval in the domain of A_i. The length of the interval is controlled by a parameter s ($0 \leq s \leq 1$), called the *query selectivity*. Specifically, the length $|I_i|$ of I_i is given by $|I_i| = |A_i| \cdot s^{\frac{1}{q}}$, where $|A_i|$ is the domain length of A_i. Clearly, a larger s implies a wider selection interval. In our experiments, qs is fixed to 2, and qd is called the *query dimensionality*.

The quality of the resulted data is measured by the average relative error of answering such count queries using the generalized table. Specifically, for each query, we evaluate it over the generalized table using the approach described in [84], and calculate the relative error as $|gen - raw|/raw$, where gen and raw represent the evaluation results over the generalized table and the microdata table, respectively.

In each set of experiments, we fix three of the parameters, ϵ, δ, k, and s, and evaluate the impact of the rest one over the quality of the generalized table. Each set of experiments contain a workload of 1k queries. The default setting of the parameters is: $\epsilon = 0.1$, $\delta = 0.8$, $k = 10$, and $s = 0.1$.

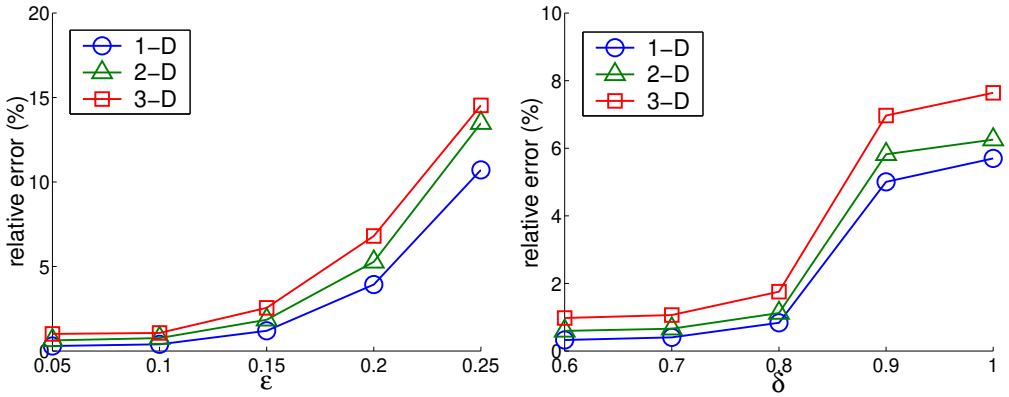


Figure 28: Average relative error with respect to parameters ϵ and δ.

Figure 28 plots the data quality (measured by average relative error) with respect to ϵ and δ. First notice that the data utility is a decreasing function of both ϵ and δ, though demonstrating fairly different patterns. This is expected, since a larger ϵ or δ indicates stricter privacy requirement, at the cost of reduced data utility. Meanwhile, the sharp increase of the relative error for δ from 0.8 to 0.9 is contributed to the fact that a majority of (ϵ, δ)-dissimilarity violations fall in this interval; for the given ϵ and initial configuration, the data utility is usually reversely correlated with the number of operations needed to remedy these violations. Nevertheless, note that in both cases, the relative error is always below 15%, even in the extreme case of query dimensionality $qd = 3$.

In Figure 29, the influence of the parameter k over the data utility shows more interesting patterns. The relative error decreases as k varies from 5 to 15, reaches its minimum at $k = 15$, and then slightly grows afterwards. Here is an intuitive explanation: for fixed ϵ and δ, we claim that smaller k usually results in a larger number of (ϵ, δ)-dissimilarity violations. To see this, consider the following simple

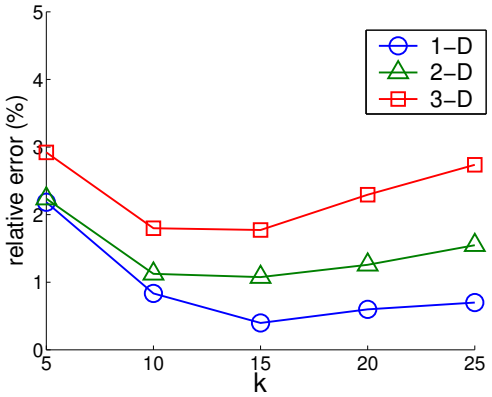


Figure 29: Average relative error with respect to parameters k.

example, which can be readily generalized to support our claim:

Example 7. Assume a random initial partition, i.e., a tuple is assigned to every QI-group with identical probability, a total number of 8 tuples, and $\delta = 1/2$. Consider a tuple x with two ϵ-neighbors x_1 and x_2 for given ϵ. If $k = 4$, x causes a violation if x_1 and x_2 are assigned to the same QI-group as x, i.e., with probability $(1/2)^2 = 1/4$; if $k = 2$, x results in a violation if either x_1 or x_2 is assigned to x's group, thus with a larger probability 1-$(3/4)^2 = 7/16$.

As we have pointed out, the data utility is usually reversely correlated with the number of operations needed to remedy the violations. This explains the high relative error for a small k. Meanwhile, recall that the QI-attribute values in a QI-group is generalized to their minimum bounding interval (for quantitative attribute) or their lowest common ancestor on the hierarchy (for categorical attribute), a larger k implies more significant distortions, which explains the high relative error for a large k.

Figure 30 shows the data utility as a function of the query selectivity s, which varies from 0.05 to 0.25. It is noticed that the average workload error decreases as s grows, which is consistent with the analysis in [84]: the generalized data enjoys higher accuracy for queries with larger selection intervals. We note that even for s as low as 0.05, our approach guarantees high data utility, with the relative error below 5%.

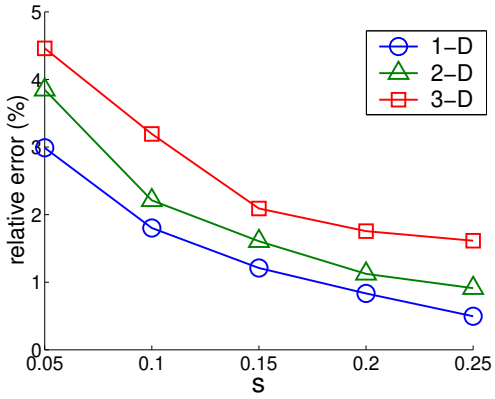


Figure 30: Average relative error with respect to parameter s.

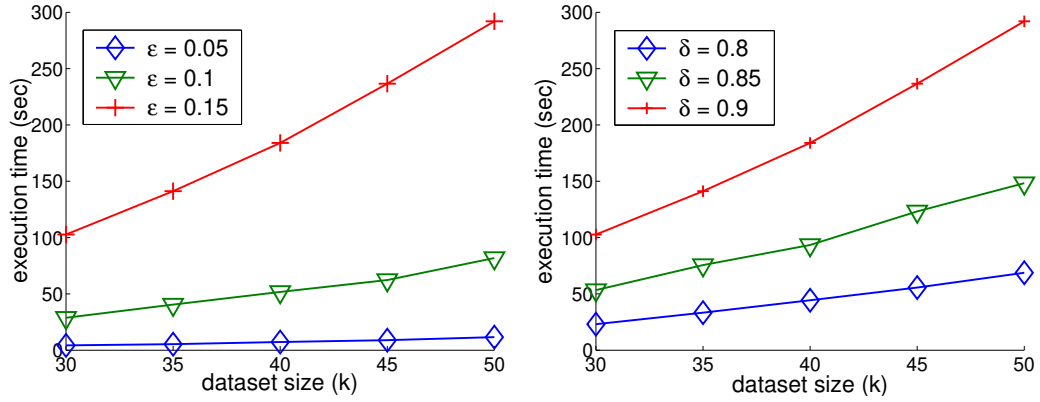


Figure 31: Average execution time with respect to ϵ, δ and the size of the dataset.

Execution Efficiency

Figure 31 and 32 plot the average execution time of our generalization algorithm with respect to the dataset size which varies from 30k to 50k. Furthermore, in each set of experiments, we fix two of the three parameters ϵ, δ, and k, and measure the impact of the remaining one on the running time. By default, the parameter setting is: $\epsilon = 0.15$, $\delta = 0.9$, and $k = 20$.

One can notice that in all these plots, the average execution time grows approximately linearly with respect to the dataset size, and all the experiments terminate within minutes. As expected, the computation cost appears as an increasing function of ϵ and δ. This is contributed to that for a given initial partition, a larger ϵ or δ

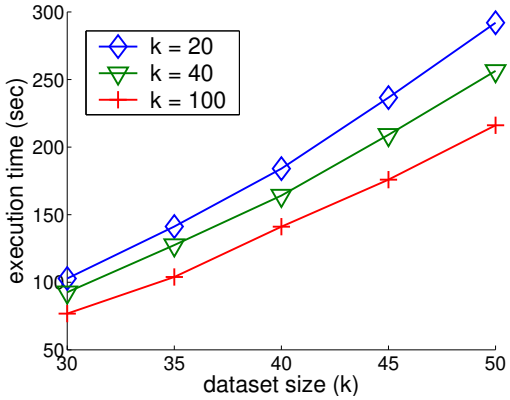


Figure 32: Average execution time with respect to k and the size of the dataset.

results in a more significant number of (ϵ, δ)-dissimilarity violations, and the computational cost is usually proportional to the number of operations needed to remedy these breaches. It is also interesting to notice that in the rightmost plot, the cost decreases as k grows, which empirically validates our analysis regarding the impact of k over the number of (ϵ, δ)-dissimilarity violations.

4.7 Related Work

The problem of centralized publication has attracted intensive research recently. The existing literatures can be classified mainly into two categories. The first one aims at devising privacy definitions and principles, as the criteria to measure the quality of the protection provided by an anonymization model. As mentioned in our discussion, k-anonymity [119], l-diversity [98], and (α, k)-anonymity [139] target association attack based on exact QI-SA association, while t-closeness [92], (k, e)-anonymity [149], and (ϵ, m)-anonymity [91] take into consideration the attack leveraging proximate association. For scenarios with different background knowledge assumptions, a collection of principles have been proposed: δ-presence [104] assumes that the adversary has no prior knowledge regarding the presence of individuals in the microdata; (c, k)-safety [99] and privacy skyline [24] consider the case that the adversary possesses external knowledge regarding the target individual, other individuals, or the family

of individuals sharing a same SA-value; m-invariance [143] is designed for sequential releases of microdata; while differential privacy [40] measures the quality of protection from the perspective of the perturbation mechanism itself.

The second category of work explores the possibility of fulfilling the proposed anonymization principles, meanwhile preserving the data utility to the maximum extent. In [2], Aggarwal showed that due to the curse of dimensionality, it is hard to enforce even 2-anonymity for high-dimensional microdata. In [2, 84, 101], it was proved that finding the optimal k-anonymization with minimum information loss is NP-Hard for the suppression, multi-dimensional, and attribute models, respectively. Despite these negative results, it was shown in [15, 83] that the optimal relation can be found efficiently by systematically enumerating the possible generalizations, in conjunction of effective pruning using heuristics. Efficient greedy-manner solutions have also be considered [48, 84, 127]. Besides the heuristic methods above, a set of approximation algorithms have been developed [15, 101, 109], which provide theoretical guarantees on the quality of the resulting data. Another direction of work attempts to optimize the data utility, without compromising the hard privacy requirement. Instead of anonymizing the whole microdata table, Kifer and Gehrke [77] advocated anonymizing and publishing a set of marginals, to ameliorate the curse of dimensionality. In [142, 149], it was proposed to publish the QI-attributes and SA-attributes separately, so as to preserve the utility of the QI-values. LeFevre et al. [85] differentiated the generalization level for different subsets of the microdata according to their importance, and propose a workload-aware anonymization scheme.

Graph coloring has been a prominent topic in graph theory for a long history. An (ordinary vertex) coloring is a partition of the vertices of a graph into independent sets. It is known that determining if a general graph can be colored with less than k colors (its chromatic number) is NP-Hard [49]. Many variants and generalizations have been considered, particularly in relation to practical applications. Cowen et

al. [32] considered a relaxation of coloring in which the color classes partition the vertices into subgraphs of degree at most d, called (k, d)-coloring, following the classic work of Lovász [96]. Erdös considered the problem of equitable coloring, imposing the constraint that each color class should be of identical size, and made the famous conjecture that the chromatic number of a graph with maximum degree Θ is at most $(\Theta + 1)$, which was later proved. However, to the best of our knowledge, no previous work exists on the problem of equitable coloring with defect.

CHAPTER V

BUTTERFLY: PRIVACY-AWARE DATA MINING

5.1 Introduction

Privacy of personal information has been arising as a vital requirement in designing and implementing data mining and management systems; individuals were usually unwilling to provide their personal information if they knew that the privacy of their data could be compromised. To this end, a plethora of work has been done on preserving *input privacy* for static data [6, 119, 42, 25, 98], which assumes untrusted data recipients and enforces privacy regulations by sanitizing the raw data before sending it to the recipients. The mining process is then performed over the sanitized data, and produce output (patterns or models) with accuracy comparable to, if not identical to that constructed over the raw data. This scenario is illustrated as the first four steps of the grand framework of privacy-preserving data mining in Figure 33.

Nevertheless, in a strict sense, privacy preservation not only requires to prevent unauthorized access to raw data that leads to exposure of sensitive information, but also includes eliminating unwanted disclosure of sensitive patterns through inference attacks over mining output. By sensitive patterns, we refer to those properties possessed uniquely by a small number of individuals participating in the input data. At the first glance, it may seem sufficient to sanitize input data in order to address such threat; however, as will be revealed, even though the patterns (or models) are built over the sanitized data, the published mining output could still be leveraged to infer sensitive patterns. Intuitively, this can be explained by the fact that input-privacy protection techniques are designed to make the constructed models close to, if not identical to that built over the raw data, in order to guarantee the utility of

the result. Such "no-outcome-change" property is considered as a key requirement of privacy-preserving data mining [21]. Given that the significant statistical information of the raw data is preserved, there exists the risk of disclosure of sensitive information. Therefore, the preservation of input privacy may not necessarily lead to that of output privacy, while it is necessary to introduce another unique layer of output-privacy protection into the framework, as shown in Figure 33. A concrete example is given as follows.

Example 8. Consider a nursing-care records database that collects the observed symptoms of the patients in a hospital. By mining such database, one can discover valuable information regarding syndromes characterizing particular diseases. However, the released mining output can also be leveraged to uncover some combinations of symptoms that are so special that only rare people match them (we will show how to achieve this in the following sections), which qualifies as a severe threat to individuals' privacy.

Assume that Alice knows that Bob has certain symptoms a, b but not c ($\overline{c}$), and by analyzing the mining output, she finds that only one person in the hospital matching the specific combination of $\{a, b, \overline{c}\}$, and only one having all $\{a, b, \overline{c}, d\}$. She can safely conclude that the victim is Bob, who also suffers symptom d. Further more, by studying other medical databases, she may learn that the combination of $\{a, b, d\}$ is linked to a rare disease with fairly high chance.

The output-privacy issue is more complicated in stream mining, wherein the mining output usually needs to be published in a continuous and in-time manner. Not only a single-time release may contain privacy breaches, but also multiple releases can potentially be exploited in combination, given the overlap of the corresponding input data. Consider the sliding window model [12] as an example, arguably the most popular stream processing model, where queries are not evaluated over the entire history of the stream, but rather over a sliding window of the most recent data from the

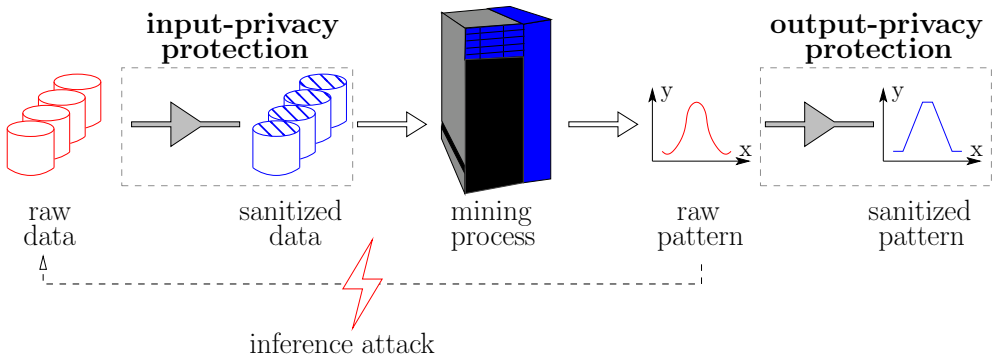


Figure 33: Grand framework of privacy-preserving data mining.

stream. The window may be defined over data items or timestamps, i.e., item-based or time-based window, respectively. Besides the leakage in the output of a single window (*intra-window* breach), the output of multiple overlapping windows can also be combined to infer sensitive information (*inter-window* breach), even each window itself contains no breach per se. Moreover, the characteristics of data streams typically evolve over time, which precludes the feasibility of global data analysis-based techniques, due to the strict processing time and memory limitations. Hence, one needs to consider addressing output-privacy vulnerabilities in stream mining systems as a unique problem.

Surprisingly, in contrast of the wealth of work on protecting input privacy, output privacy has received fairly limited attention so far in both stream data mining and privacy-preserving data mining in general. This work, to our best knowledge, represents the most systematic study to date of output-privacy vulnerabilities in the context of data stream mining.

5.1.1 State of the Art

The first naturally arising question might be: is it sufficient to apply input-privacy protection techniques to address output vulnerabilities? Unfortunately, most existing techniques fail to satisfy the requirement of countering inference attacks over mining output. They differ from one to another in terms of concrete mechanisms to provide

attack-resilient protection while minimizing utility loss of mining output incurred by sanitization; however, the adversarial attacks over input data (raw records) is significantly different from that over mining output (patterns or models).

As a concrete case, in Example 8, one conceivable solution to controlling the inference is to block or perturb those sensitive records, e.g., the one corresponding to Bob, in the mining process; however, such record-level perturbation suffers from a number of drawbacks. First, the utility of mining output is not guaranteed. Since the perturbation directly affects the mining output, it is usually difficult to guarantee both that the valuable knowledge (the intended result) is preserved and that the sensitive patterns are disguised. Among these, one significant issue is the possible large amount of false knowledge. For instance, in Example 8, if the dataset is prepared for frequent pattern mining, blocking or perturbing sensitive records may make frequent patterns become non-frequent, or vice versa; if the dataset is prepared for learning classification tree, modifying sensitive records may result in significant deviation of the cut points, crucial for decision making. Second, unlike the scenarios considered in certain existing work, in real applications, the sensitive patterns may not be predefined or directly observable; rather, sophisticated analysis over the entire dataset is typically necessary to detect potential privacy leakage in the mining output. For example, as we will show in Section 5.3, in the case of frequent pattern mining, the number of potential breaches needed to be checked is exponential in terms of the number of items. The situation is even more complicated for the case of stream mining wherein multiple windows can be exploited together for inference. Such complexity imposes efficiency issues for record-level perturbation. Third, in a broad range of computation-intensive applications, e.g., neural network-based models, the mining output is typically not directly observable; thus the effect of applying record-level perturbation cannot be evaluated without running the mining process. In all theses cases, record-level perturbation is ineffective to protect sensitive patterns.

Meanwhile, one might draw a comparison between our work and the disclosure control techniques in statistical and census databases. Both concern about providing statistical information without compromising sensitive information regarding individuals; however, they also exhibit significant distinctions. First, the queries of statistical databases typically involve only simple statistics, e.g., MIN, MAX, AVG, etc., while the output (patterns or models) of data mining applications usually feature much more complex structures, leading to more complicated requirements for output utility. Second, compared with that in statistical databases, the output-privacy protection in data mining faces much stricter constraints over processing time and space, which is especially true for the case of stream mining.

5.1.2 Overview of Our Solution

A straightforward yet inefficient solution to preserving output privacy is to detect and eliminate all potential breaches, i.e., the *detecting-then-removing* paradigm as typically adopted by inference control in statistical databases. However, the detection of breaches usually requires computation-intensive analysis of the entire dataset, which is negative in tone [27] for stream mining systems. Further, even at such high cost, the concrete operations of removing the identified breaches, e.g., suppression and addition [11], tend to result in considerable decrease in the utility of mining output.

Instead, we propose a novel proactive model to counter inference attacks over output. Analogous to sanitizing raw data from leaking sensitive information, we introduce the concept of "sanitized pattern", arguing that by intelligently modifying the "raw patterns" produced by mining process, one is able to significantly reduce the threat of malicious inference, while maximally preserving the utility of raw patterns. This scenario is shown as the last step in Figure 33.

In contrary to record-level perturbation, pattern-level perturbation demonstrates advantages in both protecting sensitive patterns and preserving output utility. First,

the utility of mining output is guaranteed, e.g., it is feasible to precisely control the amount of false knowledge. For instance, in Example 8, all the valuable frequent patterns regarding symptom-disease relationships can be preserved, while no false frequent patterns are introduced. Also, as we will show in Section 5.5 and 5.6, in the case of frequent pattern mining, not only the accuracy of each frequent itemset can be controlled, but also their semantic relationships can be preserved to the maximum extent, which is hard to achieve with record-level perturbation. Second, it is possible to devise effective yet efficient pattern-level perturbation schemes that can be performed either online or offline, without affecting the efficiency of (stream) mining process. Finally, since the target of perturbation, the mining output, is directly observable to the perturbation process, it is possible to analytically gauge the perturbation schemes.

Specifically, we present BUTTERFLY*, a light-weighted countermeasure against malicious inference over mining output. It possesses a series of desirable features that make it suitable for (stream) mining applications: 1) it needs no explicit detection of (either intra-window or inter-window) privacy breaches; 2) it requires no reference to previous output when publishing the current result; 3) it provides flexible control over the balance of multiple utility metrics and privacy guarantee.

Following a two-phase paradigm, BUTTERFLY* achieves attack-resilient protection and output-utility preservation simultaneously. In the first phase, it counters malicious inference by amplifying the uncertainty of sensitive patterns, at the cost of trivial accuracy loss of individual patterns. In the second phase, while guaranteeing the required privacy, it maximally optimizes the output utility by taking account of several model-specific semantic constraints.

Our contributions can be summarized as follows: (i) we articulate the problem and the importance of preserving output privacy in (stream) data mining; (ii) we expose a general inferencing attack model that exploits the privacy breaches existing

in current (stream) mining systems; (iii) we propose a two-phase framework that effectively addresses attacks over mining output; (iv) we provide both theoretical analysis and experimental evaluation to validate our approach in terms of privacy guarantee, output utility, and execution efficiency.

5.1.3 Roadmap

We begin in Section 5.2 with introducing the preliminaries of frequent pattern mining over data streams, exemplifying with which, we formalize the problem of addressing output-privacy vulnerabilities in (stream) data mining. In Section 5.3, after introducing a set of basic inferencing techniques, we present two general attack models that exploit intra-window and inter-window breaches in stream mining output, respectively. Section 5.4 outlines the motivation and design objectives of BUTTERFLY*, followed by Section 5.5 and 5.6 detailing the two phases of BUTTERFLY* and discussing the implicit trade-offs among privacy guarantee and multiple utility metrics. An empirical evaluation of the analytical models and the efficacy of BUTTERFLY* is presented in Section 5.7. Finally, section 5.8 surveys relevant literature.

5.2 Problem Formalization

To expose the output-privacy vulnerabilities in existing mining systems, we exemplify with the case of frequent pattern mining over data streams. We first introduce the preliminary concepts of frequent pattern mining and pattern categorization, and then formalize the problem of protecting output privacy in such mining task.

5.2.1 Frequent Pattern Mining

Consider a finite set of *items* $\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$. An *itemset* I is a subset of $\mathcal{I}$, i.e., $I \subseteq \mathcal{I}$. A *database* $\mathcal{D}$ consists of a set of records, each corresponding to a non-empty itemset. The *support* of an itemset I with respect to $\mathcal{D}$, denoted by $T_{\mathcal{D}}(I)$, is defined as the number of records containing I as a subset in $\mathcal{D}$. Frequent pattern mining

aims at finding all itemsets with supports exceeding a predefined threshold C, called *minimum support*.

A *data stream* $\mathcal{S}$ is modeled as a sequence of records, $(r_1, r_2, \ldots, r_N)$, where N is the current size of $\mathcal{S}$, and grows as time goes by. The *sliding window* model is introduced to deal with the potential of N going to infinity. Concretely, at each N, one considers only the window of most recent H records, $(r_{N-H+1}, \ldots, r_N)$, denoted by $\mathcal{S}(N, H)$, where H is the window size. The problem is therefore to find all the frequent itemsets in each window.

Example 9. Consider a data stream with current size $N = 12$, window size $H = 8$, as shown in Figure 34, where $a \sim d$ and $r_1 \sim r_{12}$ represent the set of items and records, respectively. Assuming minimum support $C = 4$, then within window $\mathcal{S}(11, 8)$, $\{c, bc, ac, abc\}$ is a subset of frequent itemsets.

One can further generalize the concept of itemset by introducing the negation of an item. Let $\overline{i}$ denote the negation of item i. A record is said to contain $\overline{i}$ if it does not contain i. Following, we will use the term *pattern* to denote a set of items or negation of items, e.g., $a\overline{bc}$. We use $\overline{I}$ to denote the negation of itmeset I, i.e., $\overline{I} = \{\overline{i} | i \in I\}$.

Analogously, we say that a record r satisfies a pattern P if it contains all the items and negation of items in P and the support of P with respect to database $\mathcal{D}$ is defined as the number of records containing P in $\mathcal{D}$.

Example 10. In Figure 34, r_{10} contains $a\overline{b}$, but not $a\overline{c}$. The pattern $\overline{a}bc$ has support 2 with respect to $\mathcal{S}(12, 8)$, because only records r_8 and r_{11} match it.

5.2.2 Pattern Categorization

Loosely speaking, output privacy refers to the requirement that the output of mining process does not disclose any sensitive information regarding individuals participating in the input data.

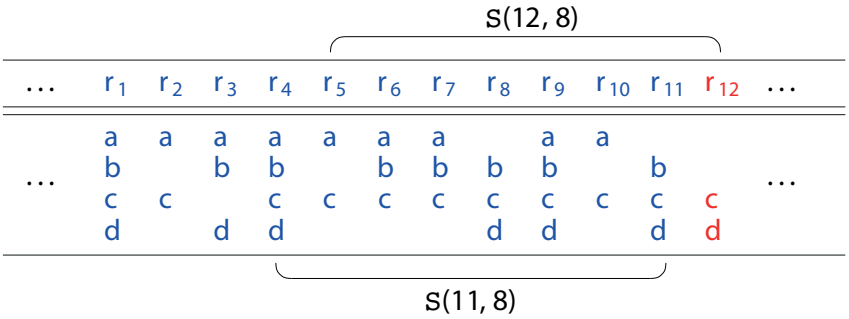


Figure 34: Data stream and sliding window model.

In the context of frequent pattern mining, such sensitive information can be instantiated as patterns with extremely low supports, which correspond to properties uniquely possessed by few records (individuals), as shown in Example 8. We capture this intuition by introducing a threshold K ($K \ll C$), called *vulnerable support*, and consider patterns with (non-zero) supports below K as *vulnerable patterns*. We can then establish the following classification system.

Definition 31. (Pattern Categorization) *Given a database $\mathcal{D}$, let $\mathcal{P}$ be the set of patterns appearing in $\mathcal{D}$, then all $P \in \mathcal{P}$ can be classified into three disjoint classes, for the given threshold K and C.*

$$\begin{cases} \textit{Frequent Pattern}: & \mathcal{P}_f = \{P | T_{\mathcal{D}}(P) \geq C\} \\ \textit{Hard Vulnerable Pattern}: & \mathcal{P}_{hv} = \{P | 0 < T_{\mathcal{D}}(P) \leq K\} \\ \textit{Soft Vulnerable Pattern}: & \mathcal{P}_{sv} = \{P | K < T_{\mathcal{D}}(P) < C\} \end{cases}$$

Intuitively, frequent pattern ($\mathcal{P}_f$) is the set of patterns with supports above minimum support; they expose the significant statistics of the underlying data, and are often the candidate in the mining process. Actually the frequent itemsets found by frequent pattern mining are a subset of $\mathcal{P}_f$. Hard vulnerable pattern ($\mathcal{P}_{hv}$) is the set of patterns with supports below vulnerable support; they represent the properties possessed by only few individuals, so it is unacceptable that they are disclosed or inferred from the mining output. Finally, soft vulnerable pattern ($\mathcal{P}_{sv}$) neither demonstrates

the statistical significance, nor violates the privacy of individual records; such patterns are not contained in the mining output, but it is usually tolerable that they are learned from the output.

Example 11. As shown in Figure 34, given $K = 1$ and $C = 4$, ac and bc are both $\mathcal{P}_f$, and $\overline{ab}c$ is $\mathcal{P}_{hv}$ with respect to $\mathcal{S}(12, 8)$, while bcd is $\mathcal{P}_{sv}$ since its support lies between K and C.

5.2.3 Problem Definition

We are now ready to formalize the problem of preserving output privacy in the context of frequent pattern mining over streams: *For each sliding window $\mathcal{S}(N, H)$, output-privacy preservation prevents the disclosure or inference of any hard vulnerable patterns with respect to $\mathcal{S}(N, H)$ from the mining output.*

It may seem at the first glance that no breach exists at all in frequent pattern mining, if it only outputs frequent itemsets (recall $C \gg K$); however, as shown in the next example (with detailed discussion in Section 5.3), from the released frequent patterns and their associated supports, the adversary may still be able to infer certain hard vulnerable patterns.

Example 12. Recall Example 11. Given the supports of $\{c, ac, bc, abc\}$, based on the inclusion-exclusion principle [106], $T(\overline{ab}c) = T(c) - T(ac) - T(bc) + T(abc)$, one is able to infer the support of $\overline{ab}c$, which is $\mathcal{P}_{hv}$ in $\mathcal{S}(12, 8)$.

5.3 Attack over Mining Output

In this section, we reveal the privacy breaches existing in current (stream) mining systems, and present a general attack model that exploits these breaches.

5.3.1 Attack Model

For simplicity of presentation, we will use the following notations: given two itemsets I and J, $I \oplus J$ denotes their union, $I \odot J$ their intersection, $J \ominus I$ the set difference of

Table 8: Symbols and notations.

notation	description
$\mathcal{S}(N,H)$	stream window of $(r_{N-H+1} \sim r_N)$
$T_{\mathcal{D}}(X)$	support of X in database $\mathcal{D}$
K	vulnerable support
C	minimum support
$\mathcal{X}_I^J$	set of itemsets $\{X \mid I \subseteq X \subseteq J\}$
W_p	previous window
W_c	current window
Δ_X^+	number of inserted records containing X
Δ_X^-	number of deleted records containing X

J and I, and $|I|$ the size of I. The notations used in the rest of the paper are listed in Table 8.

As a special case of multi-attribute aggregation, computing the support of I ($I \subseteq J$) can be considered as generalization of J over all the attributes of $J \ominus I$; therefore, one can apply the standard tool of multi-attribute aggregation, a lattice structure, based on which we construct the attack model.

Lattice Structure

Consider two itemsets I, J that satisfy $I \subset J$. All the itemsets $\mathcal{X}_I^J = \{X \mid I \subseteq X \subseteq J\}$ form a lattice structure: each node corresponds to an itemset X, and each edge represents the generalization relationship between two nodes X_s and X_t such that $X_s \subset X_t$ and $|X_t \ominus X_s| = 1$. Namely, X_s is the generalization of X_t over the item $X_t \ominus X_s$.

Example 13. A lattice structure is shown in Figure 35, where $I = c$, $J = abc$, and $J \ominus I = ab$.

For simplicity, in what follows, we use $\mathcal{X}_I^J$ to represent both the set of itemsets and their corresponding lattice structure. Next, we introduce the basis of our inferencing model, namely, *deriving pattern support* and *estimating itemset support*. These two techniques have been introduced in [11] and [23], respectively, with usage or purpose

different from ours. In [11], deriving pattern support is considered as the sole attack model to uncover sensitive patterns; in [23], estimating itemset support is used to mine non-derivable patterns, and thus saving the storage of patterns. The novelty of our work, however, lies in constructing a general inferencing model that exploits the privacy breaches existing in single or multiple releases of mining output, with these two primitives as building blocks.

Deriving Pattern Support

Consider two itemsets $I \subset J$, if the supports of all the lattice nodes of $\mathcal{X}_I^J$ are accessible, one is able to derive the support of pattern P, $P = I \oplus (\overline{J \ominus I})$, according to the inclusion-exclusion principle [106]:

$$T(I \oplus (\overline{J \ominus I})) = \sum_{I \subseteq X \subseteq J} (-1)^{|X \ominus I|} T(X)$$

Example 14. Recall the example illustrated in Figure 35. Given the supports of the lattice nodes of $\mathcal{X}_c^{abc}$ in $\mathcal{S}(12,8)$, the support of pattern $P = \overline{ab}c$ is derived as: $T_{\mathcal{S}(12,8)}(\overline{ab}c) = T_{\mathcal{S}(12,8)}(c)$ - $T_{\mathcal{S}(12,8)}(ac)$ - $T_{\mathcal{S}(12,8)}(bc)$ + $T_{\mathcal{S}(12,8)}(abc) = 1$.

Essentially, the adversary can use this technique to infer vulnerable patterns with respect to one specific window from the mining output.

Estimating Itemset Support

For the support of any itemset is non-negative, according to the inclusion-exclusion principle, if the supports of the itemsets $\{X | I \subseteq X \subset J\}$ are available, one is able to bound the support of J as follows:

$$\begin{cases} T(J) \leq \sum_{I \subseteq X \subset J} (-1)^{|J \ominus X|+1} T(X) & |J \ominus I| \text{ is odd} \\ T(J) \geq \sum_{I \subseteq X \subset J} (-1)^{|J \ominus X|+1} T(X) & |J \ominus I| \text{ is even} \end{cases}$$

Example 15. Given the supports of c, ac, and bc in $\mathcal{S}(12,8)$, one is able to establish the lower and upper bounds of $T_{\mathcal{S}(12,8)}(abc)$ as: $\leq T_{\mathcal{S}(12,8)}(ac) = 5$, $\leq T_{\mathcal{S}(12,8)}(bc) = 5$, $\geq T_{\mathcal{S}(12,8)}(ac) + T_{\mathcal{S}(12,8)}(bc) - T_{\mathcal{S}(12,8)}(c) = 2$.

Figure 35: Privacy breaches in stream mining output.

When the bounds are tight, i.e., the lower bound meets the upper bound, one can exactly determine the actual support. In our context, the adversary can leverage this technique to uncover the information regarding certain unpublished itemsets.

5.3.2 Intra-Window Inference

In stream mining systems without output-privacy protection, the released frequent itemsets over one specific window may contain intra-window breaches, which can be exploited via the technique of deriving or estimating pattern support.

Example 16. As shown in Example 14, $\overline{ab}c$ is $\mathcal{P}_{hv}$ with respect to $\mathcal{S}(12,8)$ if $K = 1$; however, one can easily derive its support if the supports of c, ac, bc, abc are known.

Formally, if J is a frequent itemset, then according to the Apriori rule [5], all $X \subseteq J$ must be frequent, which are supposed to be reported with their supports. Therefore, the information is complete to compute the support of $P = I \oplus (\overline{J \ominus I})$ for all $I \subset J$. This also implies that the number of breaches needed to be checked is potentially exponential in terms of the number of items.

Even if the support of J is unavailable, i.e, the lattice of $\mathcal{X}_I^J$ is incomplete to infer $P = I \oplus (\overline{J \ominus I})$, one can first apply the technique of estimating itemset support to complete certain missing "mosaics", then derive the supports of vulnerable patterns. Possibly the itemsets under estimation themselves may be vulnerable. Following, we assume that estimating itemset support is performed as a preprocessing step of the attack.

5.3.3 Inter-Window Inference

The intra-window inference attack is only a part of the story. In stream mining, privacy breaches may also exist in the output of overlapping windows. Intuitively, the output of a previous window can be leveraged to infer the vulnerable patterns within the current window, and vice versa, even though no vulnerable patterns can be inferred from the output of each window per se.

Example 17. Consider two windows $W_p = \mathcal{S}(11, 8)$ and $W_c = \mathcal{S}(12, 8)$ as shown in Figure 34, with frequent itemsets summarized in Figure 35. Assume $C = 4$ and $K = 1$. In window W_p, no $\mathcal{P}_{hv}$ exists; while in W_c, abc is unaccessible (shown as dashed box). From the available information of W_c, the best guess about abc is $[2, 5]$, as discussed in Example 15. Clearly, this bound is not tight enough to infer that $\overline{ab}c$ is $\mathcal{P}_{hv}$. Both windows are thus immune to intra-window inference.

However, if one is able to derive that the support of abc decreases by 1 between W_p and W_c, then based on the information released in W_p, which is $T_{W_p}(abc) = 4$, the exact value of abc in W_c can be inferred, and $\overline{ab}c$ is uncovered.

The main idea of inter-window inference is to exactly estimate the transition of the supports of certain itemsets between the previous and current windows. We below discuss how to achieve accurate estimation of such transition over two consecutive windows.

Without loss of generality, consider two overlapping windows $W_p = \mathcal{S}(N - L, H)$ and $W_c = \mathcal{S}(N, H)$ $(L < H)$, i.e., W_c is lagging W_p by L records (in the example above, $N = 12$, $H = 8$ and $L = 1$). Assume that the adversary attempts to derive the support of pattern $P = I \oplus (\overline{J \ominus I})$ in W_c. Let $\mathcal{X}_p$ and $\mathcal{X}_c$ be the subsets of $\mathcal{X}_I^J$ that are released or estimated from the output of W_p and W_c, respectively. We assume that $\mathcal{X}_p \oplus \mathcal{X}_c = \mathcal{X}_I^J$ $(\mathcal{X}_I^J \ominus \mathcal{X}_c = \mathcal{X}_p \ominus \mathcal{X}_c)$, i.e., the missing part in $\mathcal{X}_c$ can be obtained in $\mathcal{X}_p$. In Figure 35, $\mathcal{X}_p = \{c, ac, bc, abc\}$, while $\mathcal{X}_c = \{c, ac, bc\}$.

For itemset X, let Δ_X^+ and Δ_X^- be the number of records containing X in the

windows $\mathcal{S}(N, L)$ and $\mathcal{S}(N - H, L)$, respectively. Thus, the support change of X over W_p and W_c can be modeled as inserting Δ_X^+ records and deleting Δ_X^- ones, i.e., $T_{W_c}(X) = T_{W_p}(X) + \Delta_X^+ - \Delta_X^-$.

Example 18. Recall our running example, with $N = 12$, $H = 8$, and $L = 1$. $\mathcal{S}(N, L)$ corresponds to record r_{11} while $\mathcal{S}(N - H, L)$ refers to record r_4. Clearly, r_4 contains ac, while r_{11} does not; therefore, $T_{\mathcal{S}(12,8)}(ac) = T_{\mathcal{S}(11,8)}(ac) + \Delta_{ac}^+ - \Delta_{ac}^- = 5$.

The adversary is interested in estimating $T_{W_c}(X^*)$ for $X^* \in \mathcal{X}_p \ominus \mathcal{X}_c$. The bound (min, max) of $T_{W_c}(X^*)$ can be obtained by solving the following integer programming problem:

$$\max(\min) \quad T_{W_p}(X^*) + \Delta_{X^*}^+ - \Delta_{X^*}^-$$

satisfying the constraints:

$$\begin{aligned}
&\mathsf{R}_1 : 0 \le \Delta_X^+, \Delta_X^- \le L \\
&\mathsf{R}_2 : \Delta_X^+ - \Delta_X^- = T_{W_c}(X) - T_{W_p}(X) \qquad X \in \mathcal{X}_p \odot \mathcal{X}_c \\
&\mathsf{R}_3 : \Delta_X^+(\Delta_X^-) \le \textstyle\sum_{I \subseteq Y \subset X} (-1)^{|X \ominus Y|+1} \Delta_Y^+(\Delta_Y^-) \qquad |X \ominus I| \text{ is odd} \\
&\mathsf{R}_4 : \Delta_X^+(\Delta_X^-) \ge \textstyle\sum_{I \subseteq Y \subset X} (-1)^{|X \ominus Y|+1} \Delta_Y^+(\Delta_Y^-) \qquad |X \ominus I| \text{ is even}
\end{aligned}$$

Here, R_1 stems from that W_p differs from W_c by L records. When transiting from W_p to W_c, the records containing X that are deleted or added cannot exceed L. R_2 amounts to saying that the support change $(\Delta_X^+ - \Delta_X^-)$ for those itemsets $X \in \mathcal{X}_c \odot \mathcal{X}_p$ is known. R_3 and R_4 are the application of estimating itemset support for itemsets in windows $\mathcal{S}(N, L)$ and $\mathcal{S}(N - H, L)$.

Sketchily, the inference process runs as follows: starting from the change of $X \in \mathcal{X}_p \odot \mathcal{X}_c$ (R_2), by using rules R_1, R_3, and R_4, one attempts to estimate $\Delta_X^+(\Delta_X^-)$ for $X \in \mathcal{X}_p \ominus \mathcal{X}_c$. It is noted that when the interval L between W_p and W_c is small enough, the estimation can be fairly tight.

Example 19. Consider our running example, $L = 1$, and $\mathcal{X}_p \odot \mathcal{X}_c = \{c, ac, bc\}$. One

can first observe the following facts based on R_1 and R_2:

$$\Delta^+_{ac} - \Delta^-_{ac} = -1, 0 \leq \Delta^+_{ac}, \Delta^-_{ac} \leq 1 \quad \Rightarrow \quad \Delta^+_{ac} = 0, \Delta^-_{ac} = 1$$
$$\Delta^+_{bc} - \Delta^-_{bc} = -1, 0 \leq \Delta^+_{bc}, \Delta^-_{bc} \leq 1 \quad \Rightarrow \quad \Delta^+_{bc} = 0, \Delta^-_{bc} = 1$$

Take ac as an instance. Its change over W_p and W_c is Δ^+_{ac} - $\Delta^-_{ac} = -1$, and both Δ^+_{ac} and Δ^-_{ac} are bounded by 0 and 1; therefore, the only possibility is that $\Delta^+_{ac} = 0$ and $\Delta^-_{ac} = 1$. Further, by applying R_3 and R_4, one has the following facts:

$$\Delta^+_{abc} \leq \Delta^+_{ac} = 0 \quad \Rightarrow \quad \Delta^+_{abc} = 0$$
$$\Delta^-_{abc} \geq \Delta^-_{ac} + \Delta^-_{bc} - \Delta^-_{c} = 1 \quad \Rightarrow \quad \Delta^-_{abc} = 1$$

Take abc as an instance. Following the inclusion-exclusion principle, one knows that Δ^+_{abc} should be no greater than $\Delta^+_{ac} = 0$; hence, $\Delta^+_{abc} = 0$. Meanwhile, Δ^-_{abc} has tight upper and lower bounds as 1. The estimation of abc over W_c is thus given by $T_{W_c}(abc) = T_{W_p}(abc) + \Delta^+_{abc}$ - $\Delta^-_{abc} = 3$, and the $\mathcal{P}_{hv}$ $\overline{ab}c$ is uncovered.

The computation overhead of inter-window inference is dominated by the cost of solving the constrained integer optimization problems. The available fast off-the-shelf tools make such attack feasible even with moderate computation power.

*5.4 Overview of Butterfly**

Motivated by the inferencing attack model above, we outline BUTTERFLY*, our solution to protecting output privacy for (stream) mining applications.

5.4.1 Design Objective

Alternative to the reactive, detecting-then-removing scheme, we intend to use a proactive approach to tackle both intra-window and inter-window inference in a uniform manner. Our approach is motivated by two key observations. First, in certain mining applications, the utility of mining output is measured by metrics other than the exact supports of individual itemsets, e.g., the semantic relationship of their supports (e.g.,

the order or ratio of supports). It is thus acceptable to trade the accuracy of individual itemsets for boosting the output-privacy guarantee, provided that the desired output utility is maintained. Second, both intra-window and inter-window inferencing attacks are based on the inclusion-exclusion principle, which involves multiple frequent itemsets. Trivial randomness injected into each frequent itemset can accumulate into considerable uncertainty in inferred patterns. The more complicated the inference (i.e., harder to be detected), the more considerable such uncertainty.

We therefore propose BUTTERFLY*, a light-weighted output-privacy preservation scheme based on pattern perturbation. By sacrificing trivial accuracy of individual frequent itemsets, it significantly amplifies the uncertainty of vulnerable patterns, and thus blocking both intra-window and inter-window inference.

5.4.2 Mining Output Perturbation

Data perturbation refers to the process of modifying confidential data while preserving its utility for intended applications [1]. This is arguably the most important technique used to date for protecting original input data. In our scheme, we employ perturbation to inject uncertainty into mining output. The perturbation over output pattern significantly differs from that over input data. In input perturbation, the data utility is defined by the overall statistical characteristics of the dataset. The distorted data is fed as input into the following mining process. Typically no utility constraints are attached to individual data values. While in output perturbation, the perturbed results are directly presented to end-users, and the data utility is defined over each individual value.

There are typically two types of utility constraints for the perturbed results. First, each reported value should have enough accuracy, i.e., the perturbed value should not deviate wildly from the actual value. Second, the semantic relationships among the results should be preserved to the maximum extent. There exist non-trivial trade-offs

among these utility metrics. To our best knowledge, this work is the first to consider such multiple trade-offs in mining output perturbation.

Concretely, we consider two perturbation techniques, with their roots at statistics literature [27, 1]. *Value distortion* perturbs the support by adding a random value drawn from certain probabilistic distribution. *Value bucketization* partitions the range of support into a set of disjoint, mutually exclusive intervals; instead of reporting the exact support, it returns the interval which the support belongs to.

Both techniques can be applied to output perturbation. However, value bucketization leads to fairly poor utility compared with value distortion, since all supports within an interval are modified to the same value, and any semantic constraints, e.g., order or ratio, can hardly be enforced in this model. We thus focus on value distortion in the following discussion. Moreover, in order to guarantee the accuracy of individual frequent itemsets, we are more interested in probabilistic distributions with bounded intervals. We thus exemplify with discrete uniform distributions over integers, although our discussion is applicable for other distributions.

5.4.3 Operation of Butterfly*

On releasing the mining output of a stream window, we perturb the support of each frequent itemset X, $T(X)$[1] by adding a random variable r_X drawn from a discrete uniform distribution over the integers within an interval $[l_X, u_X]$. The sanitized support $T'(X) = T(X) + r_X$ is hence a random variable, which can be specified by its bias $\beta(X)$ and variance $\sigma^2(X)$. Intuitively, the bias indicates the difference of the expected value $E[T'(X)]$ and actual value $T(X)$, while the variance represents the average deviation of $T'(X)$ from $E[T'(X)]$. Note that compared with $T(X)$, r_X is non-significant, i.e., $|r_X| \ll T(X)$.

While this operation is simple, the setting of $\beta(X)$ and $\sigma^2(X)$ is non-trivial, in

[1] In what follows, without ambiguity, we omit the referred database $\mathcal{D}$ in the notations.

order to achieve sufficient privacy protection and utility guarantee simultaneously, which is the focus of our following discussion. Specifically, we will address the trade-off between privacy guarantee and output utility in Section 5.5, and the trade-offs among multiple utility metrics in Section 5.6.

*5.5 Basic Butterfly**

We start with defining the metrics to quantitatively measure the precision of individual frequent itemsets, and the privacy protection for vulnerable patterns.

5.5.1 Precision Measure

The precision loss of a frequent itemset X incurred by perturbation can be measured by the *mean square error* (mse) of the perturbed support $T'(X)$:

$$\mathsf{mse}(X) = E[(T'(X) - T(X))^2] = \sigma^2(X) + \beta^2(X)$$

Intuitively, $\mathsf{mse}(X)$ measures the average deviation of perturbed support $T'(X)$ with respect to actual value $T(X)$. A smaller mse implies higher accuracy of the output. Also, it is conceivable that the precision loss should take account of the actual support. The same mse may indicate sufficient accuracy for an itmeset with large support, but may render the output of little value for another itemset with lower support. Therefore, we have the following precision metric:

Definition 32. (Precision Degradation) *For each frequent itemset X, its precision degradation, denoted by $\mathsf{pred}(X)$, is defined as the relative mean squared error of $T'(X)$:*

$$\mathsf{pred}(X) = \frac{E[(T'(X) - T(X))^2]}{T^2(X)} = \frac{\sigma^2(X) + \beta^2(X)}{T^2(X)}$$

5.5.2 Privacy Measure

Distorting the original supports of frequent itemsets is only a part of the story, it is necessary to ensure that the distortion cannot be filtered out. Hence, one needs to

consider the adversary's power in estimating the real supports of vulnerable patterns through the protection.

Without loss of generality, assume that the adversary desires to estimate the support of pattern P of the form $I \oplus (\overline{J \ominus I})$, and has full access to the sanitized support $T'(X)$ of all $X \in \mathcal{X}_I^J$. Let $T''(P)$ represent the adversary's estimation regarding $T(P)$. The privacy protection should be measured by the error of $T''(P)$. Following let us discuss such estimation from the adversary's perspective. Along the discussion, we will show how various prior knowledge possessed by the adversary may impact the estimation.

Recall that $T(p)$ is estimated following the inclusion-exclusion principle: $T(p) = \sum_{X \in \mathcal{X}_I^J} (-1)^{|X \ominus I|} T(X)$. From the adversary's view, each support $T(X)(X \in \mathcal{X}_I^J)$ is now a random variable; therefore $T(P)$ is also a random variable. Thus, the estimation accuracy of $T''(P)$ with respect to $T(P)$ (by the adversary) can be measured by the mean square error, defined as $\mathsf{mse}(P) = E[(T(P) - T''(P))^2]$. We consider the worst case (the best case for the adversary) wherein $\mathsf{mse}(P)$ is minimized, and define the privacy guarantee based on this lower bound. Intuitively, larger $\min \mathsf{mse}(P)$ indicates more significant estimation error by the adversary, i.e., better privacy protection. Also it is noted that the privacy guarantee should account for actual support $T(P)$: if $T(P)$ is close to zero, trivial variance makes it hard for the adversary to infer if pattern P exists. Such "zero-indistinguishability" decreases as $T(P)$ grows. Therefore, we introduce the following privacy metric for vulnerable pattern P.

Definition 33. (Privacy Guarantee) *For each vulnerable pattern P, its privacy guarantee, denoted by $\mathsf{prig}(P)$, is defined as its minimum relative estimation error (by the adversary):*

$$\mathsf{prig}(P) = \frac{\min \mathsf{mse}(P)}{T^2(P)}$$

In the following, we show how various assumptions regarding the adversary's prior knowledge impact this privacy guarantee. We start the analysis by considering each

itemset independently, then take account of the interrelations among them.

Prior Knowledge 11. *The adversary may have full knowledge regarding the applied perturbation, including its distribution and parameters.*

In our case, the parameter of r_X specifies the interval $[l_X, r_X]$ from which the random variable r_X is drawn; therefore, from the adversary's view, of each $X \in \mathcal{X}_I^J$, its actual support $T(X) = T'(X) - r_X$, is a random variable following a discrete uniform distribution over interval $[l'_X, u'_X]$, where $l'_X = T'(X) - u_X$, $u'_X = T'(X) - l_X$ and has expectation $T'(X) - (l_X + u_X)/2$ and variance $\sigma^2(X)$. Recalling that $|r_X| \ll T(X)$, this is a bounded distribution over positive integers. Given the expectation of each $T(X)$, we have the following theorem that gives the lower bound of $\mathsf{mse}(P)$.

Theorem 12. *Given the distribution $f(x)$ of a random variable x, the mean square error of an estimator e of x, $\mathsf{mse}(e) = \int_{-\infty}^{\infty}(x-e)^2 f(x)dx$ reaches its minimum value $Var[x]$, when $e = E[x]$.*

(Theorem 12). We have the following derivation:

$$\begin{aligned}\mathsf{mse}(x) &= \int_{-\infty}^{\infty}(x-e)^2 f(x)\mathrm{d}x \\ &= E[x^2] + e^2 - 2e \cdot E[x] \\ &= (e - E[x])^2 + Var[x]\end{aligned}$$

Hence, $\mathsf{mse}(e)$ is minimized when $e = E[x]$. □

Therefore, $\mathsf{mse}(P)$ is minimized when $T''(P) = E[T(P)]$, which is the best guess that the adversary can achieve (note that the optimality is defined in terms of average estimation error, not the semantics, e.g., $E[T(P)]$ is possibly negative). In this case, the lowest estimation error is reached as $Var[T(P)]$.

In the case that each itemset is considered independently, the fact that $T(p)$ is a linear combination of all involved $T(X)$ implies that $Var[T(p)]$ can be approximated by the sum of the variance of all involved $T(X)$, i.e., $\min \mathsf{mse}(p) = \sum_{X \in \mathcal{X}_I^J} \sigma^2(X)$.

Prior Knowledge 13. *The supports of different frequent itemsets are interrelated by a set of inequalities, derived from the inclusion-exclusion principle.*

Here, we take into consideration the dependency among the involved itemsets. As we have shown, each itemset X is associated with an interval $[l'_X, u'_X]$ containing its possible support. Given such itemset-interval pairs, the adversary may attempt to apply these inequalities to tighten the intervals, and thus obtaining better estimation regarding the support. Concretely, this idea can be formalized in the entailment problem [22]:

Definition 34 (ENTAILMENT)**.** *A set of itemset-interval pairs $\mathcal{C}$ entail a constraint $T(X) \in [l_X, u_X]$, denoted by $\mathcal{C} \models T(X) \in [l_X, u_X]$ if every database $\mathcal{D}$ that satisfies $\mathcal{C}$, also satisfies $T(X) \in [l_X, u_X]$. The entailment is tight if for every smaller interval $[l'_X, u'_X] \subset [l_X, u_X]$, $\mathcal{C} \not\models T(X) \in [l'_X, u'_X]$, i.e., $[l_X, u_X]$ is the best interval that can be derived for $T(X)$ based on $\mathcal{C}$.*

Clearly, the goal of the adversary is to identify the tight entailment for each $T(X)$ based on the rest; however, we have the following complexity result.

Theorem 14. *Deciding whether $T(X) \in [l_X, u_X]$ is entailed by a set of itemset-interval pairs $\mathcal{C}$ is DP-Complete.*

(Theorem 14-sketch). Deciding whether $\mathcal{C} \models T(X) \in [l_X, u_X]$ is equivalent to the entailment problem in the context of probabilistic logic programming with conditional constraints [97], which is proved to be DP-Complete. □

This theorem indicates that it is hard to leverage the dependency among the involved itemsets to improve the estimation regarding each individual itemset; therefore, we can approximately consider the supports of frequent itemsets as independent varaibles in measuring the adversary's power. The privacy guarantee $\mathsf{prig}(P)$ can thus be expressed as $\mathsf{prig}(P) = \sum_{X \in \mathcal{X}_I^J} \sigma^2(X)/T^2(P)$.

Prior Knowledge 15. *The adversary may have access to other forms of prior knowledge, e.g., published statistics of the dataset, samples of a similar dataset, or supports of the top-k frequent itemsets, etc.*

All these forms of prior knowledge can be captured by the notion of *knowledge point*: a knowledge point is a specific frequent itemset X, for which the adversary has estimation error less than $\sigma^2(X)$. Note that following Theorem 14, the introduction of knowledge points in general does not influence the estimation of other itemsets. Our definition of privacy guarantee can readily incorporate this notion. Concretely, let $\mathcal{K}_I^J$ denote the set of knowledge points in the set of $\mathcal{X}_I^J$, and $\kappa^2(X)$ be the average estimation error of $T(X)$ for $X \in \mathcal{K}_I^J$. We therefore have the refined definition of privacy guarantee.

$$\mathsf{prig}(P) = \frac{\sum_{X \in \mathcal{K}_I^J} \kappa^2(X) + \sum_{X \in \mathcal{X}_I^J \setminus \mathcal{K}_I^J} \sigma^2(X)}{T^2(P)}$$

Another well-known uncertainty metric is entropy. Both variance and entropy are important and independent measures of privacy protection. However, as pointed out in [61], variance is more appropriate in measuring individual-centric privacy wherein the adversary is interested in determining the precise value of a random variable. We therefore argue that variance is more suitable for our purpose, since we are aiming at protecting the exact supports of vulnerable patterns.

Prior Knowledge 16. *The sanitized supports of the same frequent itemsets may be published in consecutive stream windows.*

Since our protection is based on independent random perturbation, if the same support is repeatedly perturbed and published in multiple windows, the adversary can potentially improve the estimation by averaging the observed output (the law of large numbers). To block this type of attack, once the perturbed support of a frequent itemset is released, we keep publishing this sanitized value if the actual support remains unchanged in consecutive windows.

Discussion

In summary, the effectiveness of BUTTERFLY* is evaluated in terms of its resilience against both intra-window and inter-window inference over stream mining output. We note three key implications.

First, the uncertainty of involved frequent itemsets is accumulated in the inferred vulnerable patterns. Moreover, more complicated inferencing attacks (i.e., harder to be detected) face higher uncertainty.

Second, the actual supports of vulnerable patterns are typically small (only a unique or less than K records match vulnerable patterns), and thus adding trivial uncertainty can make it hard to tell the existence of such patterns in the dataset.

Third, inter-window inference follows a two-stage strategy, i.e., first deducing the transition between contingent windows, then inferring the vulnerable patterns. The uncertainty associated with both stages provides even stronger protection.

5.5.3 Trade-off between Precision and Privacy

In our BUTTERFLY* framework, the trade-off between privacy protection and output utility can be flexibly adjusted by the setting of variance and bias for each frequent itemset. Specifically, variance controls the overall balance between privacy and utility, while bias gives a finer control over the balance between precision and other utility metrics, as we will show later. Here, we focus on the setting of variance. Intuitively, smaller variance leads to higher output precision, however also decreases the uncertainty of inferred vulnerable patterns, thus lower privacy guarantee.

To ease the discussion, we assume that all the frequent itemsets are associated with the same variance σ^2 and bias β. In Section 5.6 when semantic constraints are taken into account, we will lift this simplification, and consider more sophisticated settings.

Let C denote the minimum support for frequent itemsets. From the definition of

precision metrics, it can be derived that for each frequent itemset X, its precision degradation $\mathsf{pred}(X) \leq (\sigma^2 + \beta^2)/C^2$, because $T(X) \geq C$. Let $P_1(C) = (\sigma^2 + \beta^2)/C^2$, i.e., the upper bound of precision loss for frequent itemsets. Meanwhile, for a vulnerable pattern $P = I(\overline{J \setminus I})$, it can be proved that its privacy guarantee $\mathsf{prig}(P) \geq (\sum_{X \epsilon \mathcal{X}_I^J} \sigma^2)/K^2 \geq (2\sigma^2)/K^2$, because $T(P) \leq K$ and the inference involves at least two frequent itemsets. Let $P_2(C, K) = (2\sigma^2)/K^2$, i.e., the lower bound of privacy guarantee for inferred vulnerable patterns.

P_1 and P_2 provide convenient representation to control the trade-off. Specifically, setting an upper bound ϵ over P_1 guarantees sufficient accuracy of the reported frequent itemsets; while setting a lower bound δ over P_2 provides enough privacy protection for the vulnerable patterns. One can thus specify the precision-privacy requirement as a pair of parameters (ϵ, δ), where $\epsilon, \delta > 0$. That is, the setting of β and σ should satisfy $P_1(C) \leq \epsilon$ and $P_2(C, K) \geq \delta$, as

$$\sigma^2 + \beta^2 \leq \epsilon C^2 \tag{7}$$

$$\sigma^2 \geq \delta K^2/2 \tag{8}$$

To make both inequalities hold, it should be satisfied that $\epsilon/\delta \geq K^2/(2C^2)$. The term ϵ/δ is called the *precision-privacy ratio* (PPR). When precision is a major concern, one can set PPR as its minimum value $K^2/(2C^2)$ for given K and C, resulting in the minimum accuracy loss for given privacy requirement. The minimum PPR also implies that $\beta = 0$ and the two parameters ϵ and δ are coupled. We refer to the perturbation scheme with the minimum PPR as the basic Butterfly*.

5.6 Optimized Butterfly*

The basic Butterfly* scheme attempts to minimize the accuracy loss of individual frequent itemsets, without taking account of their semantic relationships. Although easy to implement and resilient against attacks, this simple scheme may easily violate these semantic constraints directly related to the specific applications of the mining

output, and thus decreasing the overall utility of the results. In this section, we refine this basic scheme by taking semantic constraints into our map, and develop constraint-aware BUTTERFLY* schemes. For given precision and privacy requirement, the optimized scheme preserves the utility-relevant semantics to the maximum extent.

In this work, we specifically consider two types of constraints, *absolute ranking* and *relative frequency*. By absolute ranking, we refer to the order of frequent itemsets according to their supports. In certain applications, users pay special attention to the ranking of patterns, rather than their actual supports, e.g., querying the top-ten most popular purchase patterns. By relative frequency, we refer to the pair-wise ratio of the supports of frequent itemsets. In certain applications, users care more about the ratio of two frequent patterns, instead of their absolute supports, e.g., computing the confidence of association rules.

To facilitate the presentation, we first introduce the concept of *frequency equivalent class* (FEC).

Definition 35. (FREQUENT EQUIVALENT CLASS). *A frequent equivalent class (FEC) is a set of frequent itemsets that feature equivalent support. Two itemsets I, J belong to the same FEC if and only if $T(I) = T(J)$. The support of a FEC fec, $T(fec)$, is defined as the support of any of its member.*

A set of frequent itemsets can be partitioned into a set of disjoint FECs, according to their supports. Also note that a set of FECs are a strictly ordered sequence: we define two FECs fec_i and fec_j as $fec_i < fec_j$ if $T(fec_i) < T(fec_j)$. Following we assume that the given set of FECs $\mathcal{FEC}$ are sorted according to their supports, i.e., $T(fec_i) < T(fec_j)$ for $i < j$.

Example 20. In our running example as shown in Fig. 35, given $C = 4$, there are three FECs, $\{cd\}$, $\{ac, bc\}$, $\{c\}$, with support 4, 5, and 8, respectively.

Apparently, to comply with the constraints of absolute ranking or relative frequency, the equivalence of itemsets in a FEC should be preserved to the maximum

extent in the perturbed output. Thus, in our constraint-aware schemes, the perturbation is performed at the level of FECs, instead of specific itemsets.

We argue that this change does not affect the privacy guarantee as advertised, provided the fact that the inference of a vulnerable pattern involves at least two frequent itemsets with different supports, i.e., at least two FECs. Otherwise assuming that the involved frequent itemsets belong to the same FEC, the inferred vulnerable pattern would have support zero, which is a contradiction. Therefore, as long as each FEC is associated with uncertainty satisfying Eq.(8), the privacy preservation is guaranteed to be above the advertised threshold.

5.6.1 Order Preservation

When the order of itemset support is an important concern, the perturbation of different FECs cannot be uniform, since that would easily invert the order of two itemsets, especially when their supports are close. Instead, we need to maximally separate the perturbed supports of different FECs, under the given constraints of Eq.(7) and Eq.(8). To capture this intuition, we first introduce the concept of *uncertainty region* of FEC.

Definition 36. (Uncertainty Region) *The uncertainty region of FEC fec is the set of possible values of its perturbed support:* $\{x|Pr(T'(fec) = x) > 0\}$.

For instance, when adding to FEC fec a random variable drawn from a discrete uniform distribution over interval $[a, b]$, the uncertainty region is all the integers within interval $[a + T(fec), b + T(fec)]$. To preserve the order of FECs with overlapping uncertainty regions, we maximally reduce their intersection, by adjusting their bias setting.

Example 21. As shown in Fig. 36, three FECs have intersected uncertainty regions, and their initial biases are all zero. After adjusting the biases properly, they share no

overlapping uncertainty region; thus, the order of their supports is preserved in the perturbed output.

Note that the order is not guaranteed to be preserved if certain FECs still have overlapping regions after adjustment, due to the constraints of given precision and privacy parameters (ϵ, δ). We intend to achieve the maximum preservation under the given requirement.

Minimizing Overlapping Uncertainty Region

Below we formalize the problem of order preservation. Without loss of generality, consider two FECs fec_i, fec_j with $T(fec_i) < T(fec_j)$. To simplify the notation, we use the following short version: let $t_i = T(fec_i)$, $t_j = T(fec_j)$, t'_i and t'_j be their perturbed supports, and β_i and β_j the bias setting, respectively.

The order of fec_i and fec_j can be possibly inverted if their uncertainty regions intersect; that is, $Pr[t'_i \geq t'_j] > 0$. We attempt to minimize this inversion probability $Pr[t'_i \geq t'_j]$ by adjusting β_i and β_j. This adjustment is not arbitrary, constrained by the precision and privacy requirement. We thus introduce the concept of *maximum adjustable bias*:

Definition 37. (Maximum Adjustable Bias) *For each FEC fec, its bias is allowed to be adjusted within the range of $[-\beta_{max}(fec), \beta_{max}(fec)]$, $\beta_{max}(fec)$ is called the maximum adjustable bias. For given ϵ and δ, it is defined as*

$$\beta_{max}(fec) = \lfloor \sqrt{\epsilon T^2(fec) - \delta K^2/2} \rfloor$$

derived from Eq.(7) and Eq.(8).

Wrapping up the discussion above, the problem of preserving absolute ranking can be formalized as: *given a set of FECs $\{fec_1, \ldots, fec_n\}$, find the optimal bias setting for each FEC fec within its maximum adjustable bias $[-\beta_{max}(fec), \beta_{max}(fec)]$ to minimize the sum of pair-wise inversion probability: min $\sum_{i<j} Pr[t'_i \geq t'_j]$.*

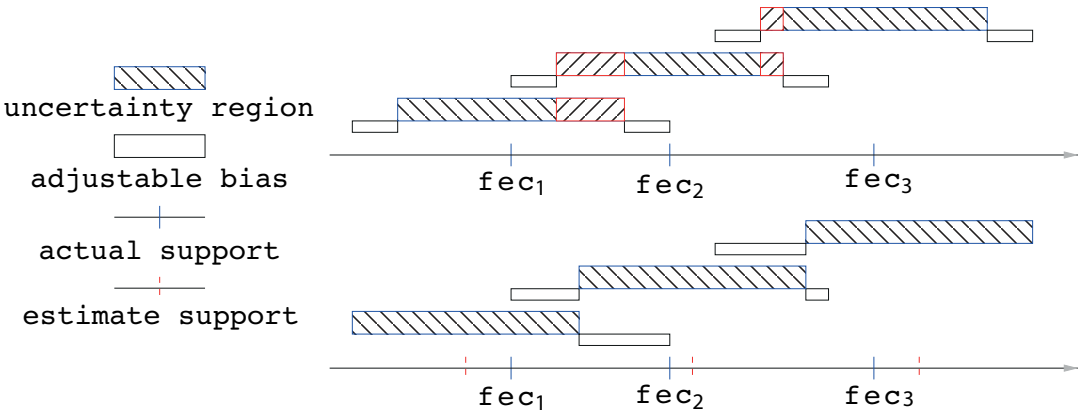


Figure 36: Adjusting bias to minimize overlapping uncertainty regions.

Exemplifying with discrete uniform distribution, we now show how to compute $Pr[t'_i \geq t'_j]$. Consider a discrete uniform distribution over interval $[a, b]$, with $\alpha = b - a$ as the interval length. The variance of this distribution is given by $\sigma^2 = [(\alpha+1)^2 - 1]/12$. According to Eq.(8), we have $\alpha = \lceil \sqrt{1 + 6\delta K^2} \rceil - 1$. Let d_{ij} be the distance of their estimators $e_i = t_i + \beta_i$ and $e_j = t_j + \beta_j$[2], i.e., $d_{ij} = e_j - e_i$.

The intersection of uncertainty regions of fec_i and fec_j is a piece-wise function, with four possible types of relationships: 1) $e_i < e_j$, fec_i and fec_j do not overlap; 2) $e_i \leq e_j$, fec_i and fec_j intersect; 3) $e_i > e_j$, fec_i and fec_j intersect; 4) $e_i > e_j$, fec_i and fec_j do not overlap. Correspondingly, the inversion probability $Pr[t'_i \geq t'_j]$ is computed as follows:

$$Pr[t'_i \geq t'_j] = \begin{cases} 0 & d_{ij} \geq \alpha + 1 \\ \frac{(\alpha+1-d_{ij})^2}{2(\alpha+1)^2} & 0 < d_{ij} < \alpha + 1 \\ 1 - \frac{(\alpha+1+d_{ij})^2}{2(\alpha+1)^2} & -\alpha - 1 < d_{ij} \leq 0 \\ 1 & d_{ij} \leq -\alpha - 1 \end{cases}$$

Following we use C_{ij} (or C_{ji}) to denote $Pr[t'_i \geq t'_j]$, the cost function of the pair of fec_i and fec_j. The formulation of C_{ij} can be considerably simplified based on the next key observation: for any pair of fec_i and fec_j with $i < j$, the solution of the optimization problem contains no configuration of $d_{ij} < 0$, as proved in the next

[2]Following we will use the setting of bias and estimator exchangeably.

lemma.

Lemma 4. *In the optimization solution of min* $\sum_{i<j} C_{ij}$*, any pair of FECs* fec_i *and* fec_j *with* $i < j$ *must have* $e_i \leq e_j$*, i.e.,* $d_{ij} \geq 0$*.*

(Lemma 4). Assume that the estimators $\{e_1, \ldots, e_n\}$ corresponding to the optimal setting, and there exists a pair of FECs, fec_i and fec_j with $i < j$ and $e_i > e_j$. By switching their setting, i.e., let e'_i (β'_i), and $e'_j(\beta'_j)$ be their new setting, and $e'_i = e_j$, and $e'_j = e_i$, the overall cost is reduced, because $\sum_{k\neq i,j} C_{ki} + C_{kj}$ remains the same, and C_{ij} is reduced, which is contradictory to the optimality assumption.

We need to prove that the new setting is feasible, that is $|\beta'_i| \leq \beta_{max}(fec_i)$ and $|\beta'_j| \leq \beta_{max}(fec_j)$. Here, we prove the feasibility of β'_i, and a similar proof applies to β'_j. First, according to the assumption, we know that

$$e_j = t_j + \beta_j < t_i + \beta_i = e_i \quad \text{and} \quad t_i < t_j$$

therefore, we have the next fact

$$\beta'_i = \beta_j + t_j - t_i < \beta_i \leq \beta_{max}(fec_i)$$

We now just need to prove that $\beta'_i \geq -\beta_{max}(fec_i)$, equivalent to $\beta_j + t_j - t_i \geq -\beta_{max}(fec_i)$, which is satisfied if

$$t_j - t_i \geq \beta_{max}(fec_j) - \beta_{max}(fec_i)$$

By substituting the maximum adjustable bias with its definition, and considering the fact $\epsilon \leq 1$, this inequality can be derived. □

Therefore, it is sufficient to consider the case $d_{ij} \geq 0$ for every pair of fec_i and fec_j when computing $Pr[t'_i \geq t'_j]$. The optimization problem is thus simplified as: $\sum_{i<j}(\alpha + 1 - d_{ij})^2$.

One flaw of the discussion so far is that we treat all FECs uniformly without considering their characteristics, i.e., the number of frequent itemsets within each

FEC. The inversion of FECs containing more frequent itemsets is more serious than that of FECs with less members. Quantitatively, let s_i be the number of frequent itemsets in FEC fec_i, the inversion of two FECs fec_i and fec_j means the order of $(s_i + s_j)$ itemsets are disturbed.

Therefore, our aim now is to solve the weighted optimization problem:

$$\begin{aligned} \min \quad & \sum_{i<j} (s_i + s_j)(\alpha + 1 - d_{ij})^2 \\ \text{s.t.} \quad & d_{ij} = \begin{cases} \alpha + 1 & e_j - e_i \geq \alpha + 1 \\ e_j - e_i & e_j - e_i < \alpha + 1 \end{cases} \\ & \forall i < j, e_i \leq e_j \\ & \forall i, e_i \in \mathbb{Z}^+, |e_i - t_i| \leq \beta_{max}(fec_i) \end{aligned}$$

This is a quadratic integer programming (QIP) problem, with piece-wise cost function. In general, QIP is NP-Hard, even without integer constraints [124]. This problem can be solved by first applying quadratic optimization techniques, such like simulated annealing, and then using random rounding techniques to impose the integer constraints. However, we are more interested in online algorithms that can flexibly trade between efficiency and accuracy. Following we present one such solution based on dynamic programming.

A Near Optimal Solution

By relaxing the constraint that $\forall i < j, e_i \leq e_j$ to $e_i < e_j$, we obtain the following key properties: (i) the estimators of all the FECs are in strict ascending order, i.e., $\forall i < j$, $e_i < e_j$; (ii) the uncertainty regions of all the FECs have the same length α. Each FEC can thus intersect with at most α of its previous ones. These properties lead to an *optimal substructure*, crucial for our solution.

Lemma 5. *Given that the biases of the last α FECs $\{fec_{n-\alpha+1} : fec_n\}$[3] are fixed as $\{\beta_{n-\alpha+1} : \beta_n\}$, and $\{\beta_1 : \beta_{n-\alpha}\}$ are optimal w.r.t. $\{fec_1 : fec_n\}$, then for given $\{\beta_{n-\alpha} : \beta_{n-1}\}$, $\{\beta_1 : \beta_{n-\alpha-1}\}$ must be optimal w.r.t. $\{fec_1 : fec_{n-1}\}$.*

(Lemma 5). Suppose that there exists a better setting $\{\beta'_1 : \beta'_{n-\alpha-1}\}$ leading to lower cost w.r.t. $\{fec_1 : fec_{n-1}\}$. Since fec_n does not intersect with any of $\{fec_1 : fec_{n-\alpha-1}\}$, the setting of $\{\beta'_1 : \beta'_{n-\alpha-1}, \beta_{n-\alpha} : \beta_n\}$ leads to lower cost w.r.t. $\{fec_1 : fec_n\}$, contradictory to our optimality assumption. □

Based on this optimal substructure, we propose a dynamic programming solution, which adds FECs sequentially according to their order. Let $C_{n-1}(\beta_{n-\alpha} : \beta_{n-1})$ represent the minimum cost that can be achieved by adjusting FECs $\{fec_1 : fec_{n-\alpha-1}\}$ with the setting of the last α FECs fixed as $\{\beta_{n-\alpha} : \beta_n\}$. When adding fec_n, the minimum cost $C_n(\beta_{n-\alpha+1} : \beta_n)$ is computed using the rule:

$$C_n(\beta_{n-\alpha+1} : \beta_n) = \min_{\beta_{n-\alpha}} C_{n-1}(\beta_{n-\alpha} : \beta_{n-1}) + \sum_{i=n-\alpha}^{n-1} (s_i + s_n)(\alpha + 1 - d_{in})^2$$

The optimal setting is the one with the minimum cost among all the combinations of $\{\beta_{n-\alpha+1} : \beta_n\}$.

Now, let us analyze the complexity of this scheme. Let β^*_{max} denote the upper bound of maximum adjustable biases of all FECs: $\beta^*_{max} = \max_i \beta_{max}(fec_i)$. For each fec, its bias can be chosen from at most $(2\beta^*_{max} + 1)$ integers. Computing $C_n(\beta_{n-\alpha+1} : \beta_n)$ for each combination of $\{\beta_{n-\alpha+1} : \beta_n\}$ from $C_{n-1}(\beta_{n-\alpha} : \beta_{n-1})$ takes at most $(2\beta^*_{max} + 1)$ steps, and the number of combinations is at most $(2\beta^*_{max} + 1)^\alpha$. The time complexity of this scheme is thus bounded by $(2\beta^*_{max} + 1)^{\alpha+1} n$, i.e., $O(n)$ where n is the total number of FECs. Meanwhile, the space complexity is also bounded by the number of cost function values needed to be recorded for each FEC, i.e., $(2\beta^*_{max} + 1)^\alpha$. In addition, at each step, we need to keep track of the bias setting for the added FECs so far for each combination, thus $(2\beta^*_{max} + 1)^\alpha (n - \alpha)$ in total.

[3] *In the following we use $\{x_i : x_j\}$ as a short version of $\{x_i, x_{i+1}, \ldots, x_j\}$.*

In practice, the complexity is typically much lower than this bound, given that (i) under the constraint $\forall i < j, e_i < e_j$, a number of combinations are invalid, and (ii) β^*_{max} is an over-estimation of the average maximum adjustable bias.

It is noted that as α or β^*_{max} grows, the complexity increases sharply, even though it is linear in terms of the total number of FECs. In view of this, we develop an approximate scheme that allows trading between efficiency and accuracy. The basic idea is that on adding each FEC, we only consider its intersection with its previous γ FECs, instead of α ones ($\gamma < \alpha$). This approximation is tight when the distribution of FECs is not extremely dense, which is usually the case, as verified by our experiments. Formally,

$$C_n(\beta_{n-\gamma+1} : \beta_n) = \min_{\beta_{n-\gamma}} C_{n-1}(\beta_{n-\gamma} : \beta_{n-1}) + \sum_{i=n-\gamma}^{n-1} (s_i + s_n)(\alpha + 1 - d_{in})^2$$

Now the complexity is bounded by $(2\beta^*_{max} + 1)^{\gamma+1} n$. By properly adjusting γ, one can control the balance between accuracy and efficiency.

The complete algorithm is sketched in Algorithm 7: one first initializes the cost function for the first γ FECs; then by running the dynamic programming procedure, one computes the cost function for each newly added FEC. The optimal configuration is the one with the global minimum value $C_n(\beta_{n-\gamma+1} : \beta_n)$.

5.6.2 Ratio Preservation

In certain applications, the relative frequency of the supports of two frequent itemsets carries important semantics, e.g., the confidence of association rules. However, the random perturbation may easily render the ratio of the perturbed supports considerably deviate from the original value. Again, we achieve the maximum ratio preservation by intelligently adjust the bias setting of FECs. First, we formalize the problem of ratio preservation.

Algorithm 7: Order-preserving bias setting

Input: $\{t_i, \beta_{max}(fec_i)\}$ for each $fec_i \in \mathcal{FEC}$, α, γ.
Output: β_i for each $fec_i \in \mathcal{FEC}$.

```
begin
    /*initialization*/;
    for β_1 = −β_max(fec_1) : β_max(fec_1) do
        C_1(β_1) = 0;
    for i = 2 : γ do
        for β_i = −β_max(fec_i) : β_max(fec_i) do
            /*e_i < e_j*/;
            if β_i + t_i > β_{i−1} + t_{i−1} then
                C_i(β_1 : β_i) = C_{i−1}(β_1 : β_{i−1}) + Σ_{j=1}^{i−1}(s_j + s_i)(α + 1 − d_ji)^2;
    /* dynamic programming */;
    for i = γ + 1 : n do
        for β_i = −β_max(fec_i) : β_max(fec_i) do
            if β_i + t_i > β_{i−1} + t_{i−1} then
                C_i(β_{i−γ+1} : β_i) = min_{β_{i−γ}} C_{i−1}(β_{i−γ} : β_{i−1}) +
                Σ_{j=i−γ}^{i−1}(s_j + s_i)(α + 1 − d_ji)^2;
    /*find the optimal setting*/;
    find the minimum C_n(β_{n−γ+1} : β_n);
    backtrack and output β_i for each fec_i ∈ FEC;
end
```

Maximizing $(k, 1/k)$ Probability of Ratio

Consider two FECs fec_i and fec_j with $t_i < t_j$. To preserve the ratio of fec_i and fec_j, one is interested in making the ratio of perturbed supports t'_i/t'_j appear in the proximate area of original value t_i/t_j with high probability, e.g., interval $[k\frac{t_i}{t_j}, \frac{1}{k}\frac{t_i}{t_j}]$, where $k \in (0, 1)$ indicates the tightness of this interval. We therefore introduce the concept of $(k, 1/k)$ probability.

Definition 38. ($(k, 1/k)$ Probability) *The $(k, 1/k)$ probability of the ratio of two random variables t'_i and t'_j, $Pr_{(k,1/k)}\left[\frac{t'_i}{t'_j}\right]$ is defined as*

$$Pr_{(k,1/k)}\left[\frac{t'_i}{t'_j}\right] = Pr\left[k\frac{t_i}{t_j} \leq \frac{t'_i}{t'_j} \leq \frac{1}{k}\frac{t_i}{t_j}\right]$$

This $(k, 1/k)$ probability quantitatively describes the proximate region of original

ratio t_i/t_j. A higher probability that t'_i/t'_j appears in this region indicates better ratio preservation. We are therefore interested in solving the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{i<j} Pr_{(k,1/k)}\left[\frac{t'_i}{t'_j}\right] \\ \text{s.t} \quad & \forall i, e_i \in \mathbb{Z}^+, |e_i - t_i| \leq \beta_{max}(fec_i) \end{aligned}$$

It is not hard to see that in the case of discrete uniform distribution, the $(k, 1/k)$ probability of the ratio of two random variables is a non-linear piece-wise function, i.e., a non-linear integer optimization problem. In general, non-linear optimization problem is NP-Hard, even without integer constraints. Instead of applying off-the-shelf non-linear optimization tools, we are more interested in efficient heuristics that can find near-optimal configurations with linear complexity in terms of the number of FECs. Following, we present one such scheme that performs well in practice.

A Near Optimal Solution

We construct our bias setting scheme based on Markov's inequality. To maximize the $(k, 1/k)$ probability $Pr\left[k\frac{t_i}{t_j} \leq \frac{t'_i}{t'_j} \leq \frac{1}{k}\frac{t_i}{t_j}\right]$, we can alternatively minimize $Pr\left[\frac{t'_i}{t'_j} \geq \frac{1}{k}\frac{t_i}{t_j}\right]$ + $Pr\left[\frac{t'_j}{t'_i} \geq \frac{1}{k}\frac{t_j}{t_i}\right]$. From Markov's inequality, we knows that $Pr\left[\frac{t'_i}{t'_j} \geq \frac{1}{k}\frac{t_i}{t_j}\right]$ is bounded by

$$Pr\left[\frac{t'_i}{t'_j} \geq \frac{1}{k}\frac{t_i}{t_j}\right] \leq \frac{E\left[\frac{t'_i}{t'_j}\right]}{\frac{1}{k}\frac{t_i}{t_j}} = k\frac{t_j}{t_i}E\left[\frac{t'_i}{t'_j}\right]$$

The maximization of the $(k, 1/k)$ probability of t'_i/t'_j is therefore simplified as the following expression (k is omitted since it does not affect the optimization result):

$$\min \frac{t_j}{t_i}E\left[\frac{t'_i}{t'_j}\right] + \frac{t_i}{t_j}E\left[\frac{t'_j}{t'_i}\right] \tag{9}$$

The intuition here is that neither expectation $\frac{t_j}{t_i}E\left[\frac{t'_i}{t'_j}\right]$ nor $\frac{t_i}{t_j}E\left[\frac{t'_j}{t'_i}\right]$ should deviate wildly from one.

According to its definition, the expectation of $\frac{t'_i}{t'_j}$, $E\left[\frac{t'_i}{t'_j}\right]$, is computed as

$$E\left[\frac{t'_i}{t'_j}\right] = \frac{1}{(\alpha+1)^2}\sum_{e_j-\alpha/2}^{e_j+\alpha/2}\frac{1}{t'_j}\sum_{e_i-\alpha/2}^{e_i+\alpha/2}t'_i = \frac{e_i}{\alpha+1}(H_{e_j+\frac{\alpha}{2}} - H_{e_j-\frac{\alpha}{2}})$$

where H_n is the nth harmonic number. It is known that $H_n = \ln n + \Theta(1)$, thus

$$E\left[\frac{t'_i}{t'_j}\right] \approx \frac{e_i}{\alpha+1}\ln\frac{e_j+\alpha/2}{e_j-\alpha/2} = \frac{e_i}{\alpha+1}\ln(1+\frac{\alpha}{e_j-\alpha/2}) \tag{10}$$

This form is still not convenient for computation. We are therefore looking for a tight approximation for the logarithm part of the expression. It is known that $\forall x, y \in \mathbb{R}^+$, $(1+x/y)^{y+x/2}$ is a tight upper bound for e^x. We have the following approximation by applying this bound: $1+\alpha/(e_j-\alpha/2) = e^{\frac{\alpha}{e_j-\alpha/2+\alpha/2}} = e^{\frac{\alpha}{e_j}}$.

Applying this approximation to computing $E\left[\frac{t'_i}{t'_j}\right]$ in Eq.(10), it is derived that

$$E\left[\frac{t'_i}{t'_j}\right] = \frac{e_i}{\alpha+1}\ln e^{\frac{\alpha}{e_j}} = \frac{\alpha}{\alpha+1}\frac{e_i}{e_j}$$

The optimization of Eq.(9) is thus simplified as

$$\min \frac{t_j}{t_i}\frac{e_i}{e_j} + \frac{t_i}{t_j}\frac{e_j}{e_i} \tag{11}$$

Assuming that e_i is fixed, by differentiating Eq.(11) w.r.t. e_j, and setting the derivative as 0, we get the solution of e_j as $e_j/e_i = t_j/t_i$, i.e., $\beta_j/\beta_i = t_j/t_i$.

Following this solution is our bottom-up bias setting scheme: for each FEC fec_i, its bias β_i should be set in proportion to its support t_i. Note that the larger $(t_i+\beta_i)$ compared with α, the more accurate the applied approximation; hence, β_i should be set as its maximum possible value.

Algorithm 8 sketches the bias setting scheme: one first sets the bias of the minimum FEC fec_1 as its maximum $\beta_{max}(fec_1)$, and for each remaining FEC fec_i, its bias β_i is set in proportion to t_i/t_{i-1}. In this scheme, for any pair of fec_i and fec_j, their biases satisfy $\beta_i/\beta_j = t_i/t_j$. Further, we have the following lemma to prove the feasibility of this scheme. By feasibility, we mean that for each FEC fec_i, β_i falls within the allowed interval $[-\beta_{max}(fec_i), \beta_{max}(fec_i)]$.

Algorithm 8: Ratio-preserving bias setting

Input: $\{t_i\}$ for each $fec_i \in FEC$, ϵ, δ, K.
Output: β_i for each $fec_i \in FEC$.

```
1 begin
2     /* setting of the minimum FEC */;
3     set β_1 = ⌊√(εt_1^2 − δK^2/2)⌋;
4     /* bottom-up setting */;
5     for i = 2 : n do
6         set β_i = ⌊β_{i−1} t_i/t_{i−1}⌋;
7 end
```

Lemma 6. *For two FECs fec_i and fec_j with $t_i < t_j$, if the setting of β_i is feasible for fec_i, then the setting $\beta_j = \beta_i \frac{t_j}{t_i}$ is feasible for fec_j.*

(Lemma 6). Given that $0 < \beta_i \leq \beta_{max}(fec_i)$, then according to the definition of maximum adjustable bias, β_j has the following property

$$\begin{aligned} \beta_j &= \beta_i \frac{t_j}{t_i} \leq \beta_{max}(fec_i)\frac{t_j}{t_i} = \lfloor \sqrt{\epsilon t_i^2 - \frac{\delta K^2}{2}} \rfloor \frac{t_j}{t_i} \\ &= \lfloor \sqrt{\epsilon t_j^2 - \frac{\delta K^2}{2}\frac{t_j^2}{t_i^2}} \rfloor \leq \lfloor \sqrt{\epsilon t_j^2 - \frac{\delta K^2}{2}} \rfloor = \beta_{max}(fec_j) \end{aligned}$$

□

Thus if β_1 is feasible for fec_1, β_i is feasible for any fec_i with $i > 1$, since $t_i > t_1$.

5.6.3 A Hybrid Scheme

While order-preserving and ratio-preserving bias settings achieve the maximum utility at their ends, in certain applications wherein both semantic relationships are important, it is desired to balance the two quality metrics in order to achieve the overall optimal quality.

We thus develop a hybrid bias setting scheme that takes advantage of the two schemes, and allows to flexibly adjust the trade-off between the two factors. Specifically, for each FEC fec, let $\beta_{op}(fec)$ and $\beta_{rp}(fec)$ denote its bias setting obtained

by the order-preserving and frequency-preserving schemes, respectively. We have the following setting based on a linear combination:

$$\forall fec \in \mathcal{FEC} \quad \beta(fec) = \lambda\beta_{op}(fec) + (1-\lambda)\beta_{rp}(fec)$$

The parameter λ is a real number within the interval of $[0, 1]$, which controls the trade-off between the two quality metrics. Intuitively, a larger λ tends to indicate more importance over order information, but less over ratio information, and vise versa. Particularly, the order-preserving and ratio-preserving schemes are the special cases when $\lambda = 1$ and 0, respectively.

5.7 Experimental Analysis

In this section, we investigate the efficacy of the proposed BUTTERFLY* approaches. Specifically, the experiments are designed to measure the following three properties: 1) privacy guarantee: the effectiveness against both intra-window and inter-window inference; 2) result utility: the degradation of the output accuracy, the order and ratio preservation, and the trade-off among these utility metrics; 3) execution efficiency: the time taken to perform our approaches. We start with describing the datasets and the setup of the experiments.

5.7.1 Experimental Setting

We tested our solutions over both synthetic and real datasets. The synthetic dataset T20I4D50K is obtained by using the data generator as described in [5], which mimics transactions from retail stores. The real datasets used include: 1) BMS-WebView-1, which contains a few months of clickstream data from an e-commerce web site; 2) BMS-POS, which contains several years of point-of-sale from a large number of electronic retailers; 3) Mushroom in UCI KDD archive[4], which is used widely in

[4]http://kdd.ics.uci.edu/

machine learning research. All these datasets have been used in frequent pattern mining over streams [26].

We built our BUTTERFLY* prototype on top of *Moment* [26], a streaming frequent pattern mining framework, which finds closed frequent itemsets over a sliding window model. By default, the minimum support C and vulnerable support K are set as 25 and 5, respectively, and the window size is set as 2K. Note that the setting here is designed to test the effectiveness of our approaches with high ratio of vulnerable/minimum threshold (K/C). All the experiments were performed over a workstation with Intel Xeon 3.20GHz and 4GB main memory, running Red Hat Linux 9.0 operating system. The algorithm is implemented in C++ and compiled using g++ 3.4.

5.7.2 Experimental Results

To provide an in-depth understanding of our output-privacy protection schemes, we evaluated four different versions of BUTTERFLY*: the basic version, the optimized version with $\lambda = 0$, 0.4, and 1, respectively, over both synthetic and real datasets. Note that $\lambda = 0$ corresponds to the ratio-preserving scheme, while $\lambda = 1$ corresponds to the order-preserving one.

Privacy and Precision

To evaluate the effectiveness of BUTTERFLY* in terms of output-privacy protection, we need to find all potential privacy breaches in the mining output. This is done by running an analysis program over the results returned by the mining algorithm, and finding all possible vulnerable patterns that can be inferred through either intra-window or inter-window inference.

Concretely, given a stream window, let $\mathcal{P}_{hv}$ denote all the hard vulnerable patterns that are inferable from the mining output. After the perturbation, we evaluate the relative deviation of the inferred value and the estimator for each pattern $P \in \mathcal{P}_{hv}$ for

100 continuous windows. we use the following average privacy (avg_prig) metric to measure the effectiveness of privacy preservation:

$$\text{avg\_prig} = \sum_{P \epsilon \mathcal{P}_{hv}} \frac{(T'(P) - E[T'(P)])^2}{T^2(P)|\mathcal{P}_{hv}|}$$

The decrease of output precision is measured by the average precision degradation (avg_pred) of all frequent itemsets $\mathcal{I}$:

$$\text{avg\_pred} = \sum_{I \epsilon \mathcal{I}} \frac{(T'(I) - T(I))^2}{T^2(I)|\mathcal{I}|}$$

In this set of experiments, we fix the precision-privacy ratio as $\epsilon/\delta = 0.04$, and measure avg_prig and avg_pred for different settings of ϵ (δ).

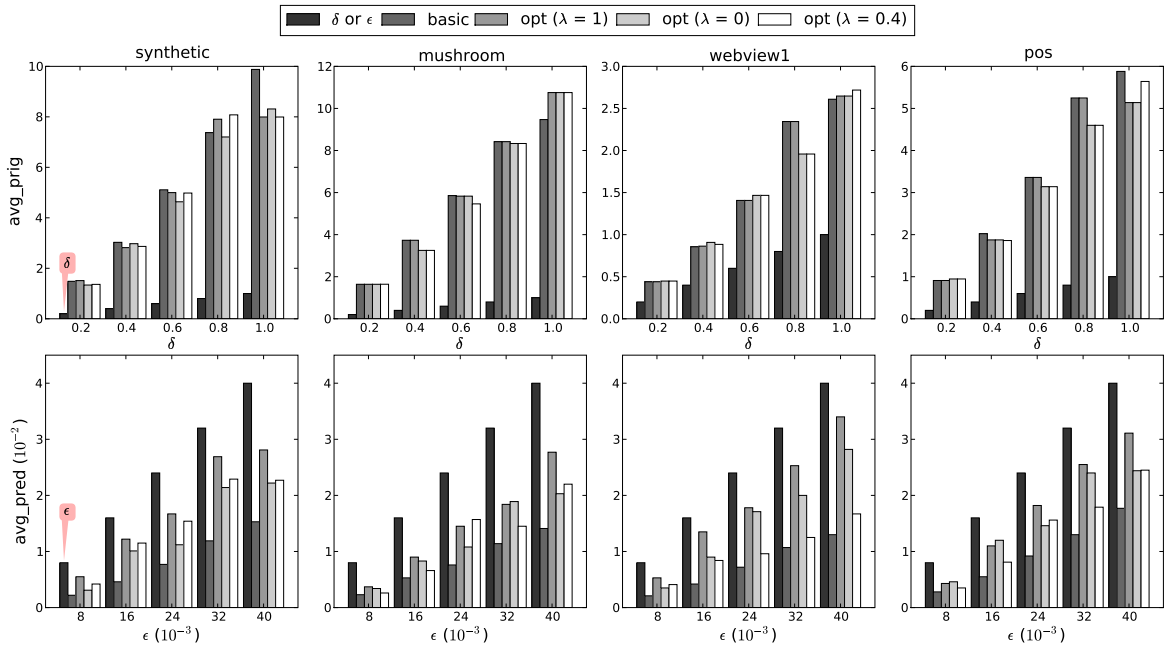


Figure 37: Average privacy guarantee (*avg_prig*) and precision degradation (*avg_pred*).

Specifically, the four plots in the top tier of Figure 37 show that as the value of δ increases, all four versions of Butterfly* provide similar amount of average privacy protection for all the datasets, far above the minimum privacy guarantee δ. The four plots in the lower tier show that as σ increases from 0 to 0.04, the output precision decreases; however, all four versions of Butterfly* have average

precision degradation below the system-supplied maximum threshold ϵ. Also note that among all the schemes, basic BUTTERFLY* achieves the minimum precision loss, for given privacy requirement. This can be explained by the fact that the basic scheme considers no semantic relationships, and sets all the biases as zero, while optimized BUTTERFLY* trades precision for other utility-related metrics. Although the basic scheme maximally preserves the precision, it may not be optimal in the sense of other utility metrics, as shown next.

Order and Ratio

For given privacy and precision requirement (ϵ, δ), we measure the effectiveness of BUTTERFLY* in preserving order and ratio of frequent itemsets.

The quality of order preservation is evaluated by the proportion of order-preserved pairs over all possible pairs, referred to as the rate of order preserved pairs (ropp):

$$\mathsf{ropp} = \frac{\sum_{I,J \in \mathcal{I} \text{ and } T(I) \leq T(J)} \mathbf{1}_{T'(I) \leq T'(J)}}{C^2_{|\mathcal{I}|}}$$

where $\mathbf{1}_x$ is the indicator function, returning 1 if condition x holds, and 0 otherwise.

Analogously, the quality of ratio preservation is evaluated by the fraction of the number of $(k, 1/k)$ probability-preserved pairs over the number of possible pairs, referred to as the rate of ratio preserved pairs (rrpp) (k is set 0.95 in all the experiments):

$$\mathsf{rrpp} = \frac{\sum_{I,J \in \mathcal{I} \text{ and } T(I) \leq T(J)} \mathbf{1}_{k \frac{T(I)}{T(J)} \leq \frac{T'(I)}{T'(J)} \leq \frac{1}{k} \frac{T(I)}{T(J)}}}{C^2_{|\mathcal{I}|}}$$

In this set of experiments, we vary the precision-privacy ratio ϵ/δ for fixed $\delta = 0.4$, and measure the ropp and rrpp for four versions of BUTTERFLY* (the parameter $\gamma = 2$ in all the experiments), as shown in Figure 38.

As predicted by our theoretical analysis, the order-preserving ($\lambda = 1$) and ratio-preserving ($\lambda = 0$) bias settings are fairly effective, both outperform all other schemes at their ends. The ropp and rrpp increase as the ratio of ϵ/δ grows, due to the fact

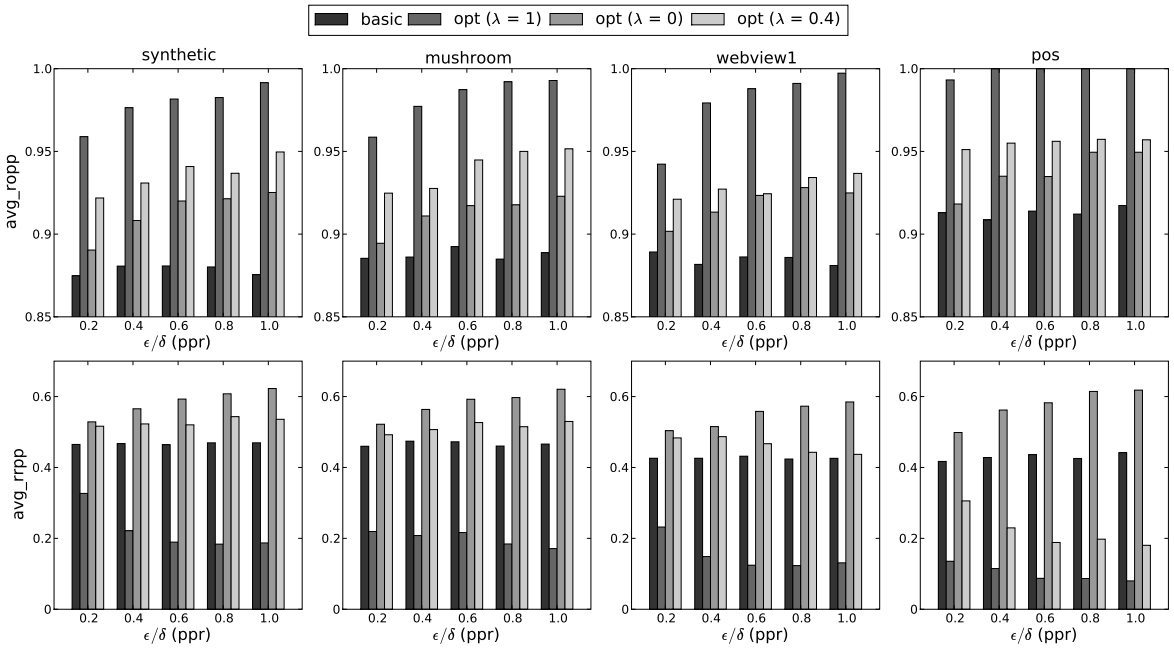


Figure 38: Average order preservation (avg_ropp) and ratio preservation (avg_rrpp).

that larger ϵ/δ offers more adjustable bias therefore leading to better quality of order or ratio preservation.

It is also noticed that order-preserving scheme has the worst performance in terms of avg_rrpp, even worse than the basic approach. This is explained by that in order to distinguish overlapping FECs, the order-preserving scheme may significantly distort the ratio of pairs of FECs. In all these cases, the hybrid scheme $\lambda = 0.4$ achieves the second best in terms of avg_rrpp and avg_ropp, and an overall best quality when order and ratio information is equally important.

Tuning of Parameters γ and λ

Next we give a detailed discussion on the setting of the parameters γ and λ.

Specifically, γ controls the depth of dynamic programming in the order-preserving bias setting. Intuitively, a larger γ leads to better quality of order preservation, but also higher time and space complexity. We desire to characterize the gain of the quality of order preservation with respect to γ, and find the setting that balances the

quality and efficiency.

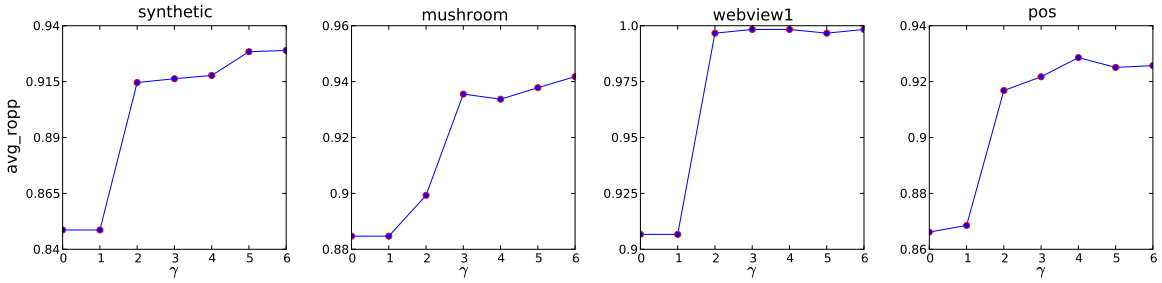


Figure 39: Average rate of order-preserved pairs with respect to setting of γ.

For all four datasets, we measure the ropp with respect to the setting of γ, with result shown in Figure 39. It is noted that the quality of order preservation increases sharply at certain points $\gamma = 2$ or 3, and the trend becomes much flatter after that. This is explained by that in most real datasets, the distribution of FECs is not extremely dense; under proper setting of (ϵ, δ), a FEC can intersect with only 2 or 3 neighboring FECs on average. Therefore, the setting of small γ is usually sufficient for most datasets.

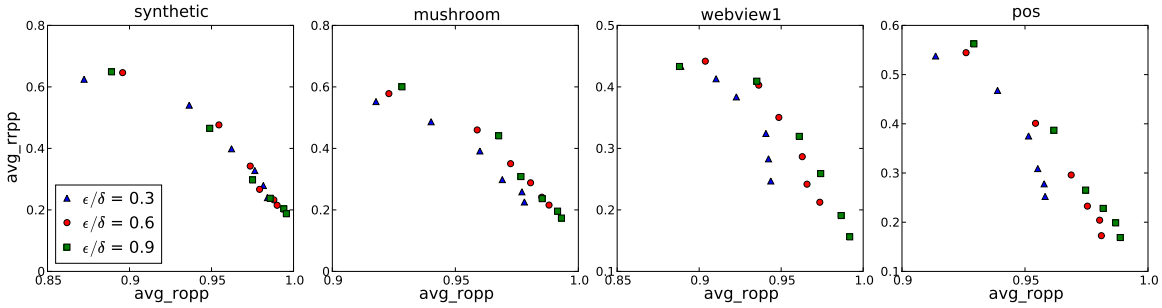


Figure 40: Trade-off between order preservation and ratio preservation.

The setting of λ balances the quality of order and ratio preservation. For each dataset, we evaluate ropp and rrpp with different settings of λ (0.2, 0.4, 0.6, 0.8 and 1) and precision-privacy ratio ϵ/δ (0.3, 0.6 and 0.9), as shown in Figure 40.

These plots give good estimation of the gain of order preservation, for given cost of ratio preservation that one is willing to sacrifice. A larger ϵ/δ gives more room for this adjustment. In most cases, the setting of $\lambda = 0.4$ offers a good balance between

the two metrics. The trade-off plots could be made more accurate by choosing more settings of λ and ϵ/δ to explore more points in the space.

Execution Efficiency

In the last set of experiments, we measure the computation overhead of BUTTERFLY* over the original mining algorithm for different settings of minimum support C. We divide the execution time into two parts contributed by the mining algorithm (mining algorithm) and BUTTERFLY* algorithm (butterfly), respectively. Note that we do not distinguish basic and optimized BUTTERFLY*, since basic BUTTERFLY* involves simple perturbation operations, with unnoticeable cost. The window size is set 5K for all four datasets.

The result plotted in Figure 41 shows clearly that the overhead of BUTTERFLY* is much less significant than the mining algorithm; therefore, it can be readily implemented atop existing stream mining algorithms. Further, while the current versions of BUTTERFLY* are window-based, it is expected that an incremental version of BUTTERFLY* can achieve even lower overhead.

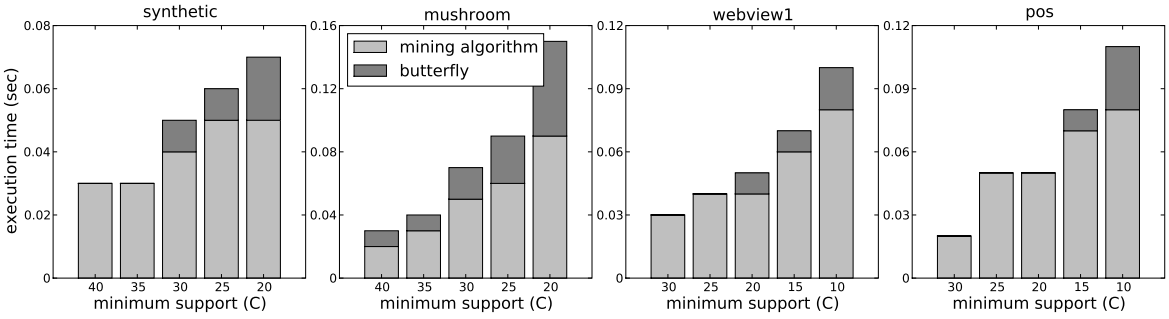


Figure 41: Overhead of BUTTERFLY* algorithms in stream mining systems.

It is noted that in most cases, the running time of both mining algorithm and BUTTERFLY* algorithm grows significantly as C decreases; however, the growth of the overhead of BUTTERFLY* is much less evident compared with the mining algorithm itself. This is expected since as the minimum support decreases, the number of

frequent itemsets increases super-linearly, but the number of FECs has much lower growth rate, which is the most influential factor for the performance of BUTTERFLY*.

5.8 Related Work

5.8.1 Disclosure Control in Statistical Database

The most straightforward solution to preserving output privacy is to detect and eliminate all potential privacy breaches, i.e., the *detecting-then-removing* strategy, which stemmed from the inference control in statistical and census databases from 1970's. Motivated by the need of publishing census data, the statistics literature focuses mainly on identifying and protecting the privacy of sensitive data entries in contingency tables, or tables of counts corresponding to cross-classification of the microdata.

Extensive research has been done in statistical databases to provide statistical information without compromising sensitive information regarding individuals [27, 113, 1]. The techniques, according to their application scenarios, can be broadly classified as *query restriction* and *data perturbation*. The query restriction family includes controlling the size of query results [46], restricting the overlap between successive queries [38], suppressing the cells of small size [33], and auditing queries to check privacy compromises [27]; the data perturbation family includes sampling microdata [37], swapping data entries between different cells [34], and adding noises to the microdata [121] or the query results [37]. Data perturbation by adding statistical noise is an important method of enhancing privacy. The idea is to perturb the true value by a small amount ϵ where ϵ is a random variable with mean = 0 and a small variance = σ^2. While we adopt the method of perturbation from statistical literature, one of our key technical contributions is the generalization of the basic scheme by adjusting the mean to accommodate various semantic constraints in the applications of mining output.

5.8.2 Input Privacy Preservation

Intensive research efforts have been directed to addressing the input-privacy issues. The work of [6, 3] paved the way for the rapidly expanding field of privacy-preserving data mining; they established the main theme of privacy-preserving data mining so as to provide sufficient privacy guarantee while minimizing the information loss in the mining output. Under this framework, a variety of techniques have been developed.

The work of [6, 3, 42, 25] applied data perturbation, specifically random noise addition, to association rule mining, with the objective of maintaining sufficiently accurate estimation of frequent patterns while preventing disclosure of specific transactions (records). In the context of data dissemination and publication, group-based anonymization approaches have been considered. The existing work can be roughly classified as two categories: the first one aims at devising anonymization models and principles, as the criteria to measure the quality of privacy protection, e.g., k-anonymity [119], l-diversity [98], $(\epsilon, \delta)^k$-dissimilarity [132], etc.; the second category of work explores the possibility of fulfilling the proposed anonymization principles, meanwhile preserving the data utility to the maximum extent, e.g., [84, 109]. Cryptographic tools have also been used to construct privacy-preserving data mining protocols, e.g., secure multi-party computation [94, 123]. Nevertheless, all these techniques focus on protecting input privacy for static datasets, Quite recently, the work [90] addresses the problem of preserving input privacy for streaming data, by online analysis of correlation structure of multivariate streams. The work [21] distinguishes the scenario of data custodian, where the data collector is entrusted, and proposes a perturbation scheme that guarantees no change in the mining output. In [73, 65], it is shown that a hacker can potentially employ spectral analysis to separate the random noise from the real values for multi-attribute data.

5.8.3 Output Privacy Preservation

Compared with the wealth of techniques developed for preserving input privacy, the attention given to protecting mining output is fairly limited. The existing literature can be broadly classified as two categories. The first category attempts to propose general frameworks for detecting potential privacy breaches. For example, the work [71] proposes an empirical testing scheme for evaluating if the constructed classifier violates the privacy constraint. The second category focuses on proposing algorithms to address the detected breaches for specific mining tasks. For instance, it is shown in [11] that the association rules can be exploited to infer information about individual transactions; while the work [126] proposes a scheme to block the inference of sensitive patterns satisfying user-specified templates by suppressing certain raw transactions.

CHAPTER VI

XSTAR: PRIVACY-AWARE LOCATION DATA MANAGEMENT

6.1 Introduction

With ubiquitous wireless connectivity and continued advance in mobile positioning technologies (e.g., cellular phones, GPS-like devices), recent years have witnessed the explosive growth of location-based services (LBS). Examples include location-based store finders ("*Where is the nearest gas station to my current location?*"), traffic condition tracking ("*What is the traffic condition on Highway 85 North?*"), and spatial alarms ("*Remind me to drop off a letter when I am near a post office.*"). The mobile users obtain such services by issuing queries together with their location information to the LBS service providers. Nevertheless, while offering great convenience and business opportunities, LBS also opens the door for misuse of mobile users' private location information [136, 72]. For example, the collected location information can be exploited to spam users with unwanted advertisements, execute physical stalking [47, 122], or perform inference about personal medical records by knowing user's frequent visits to specific clinics.

A plethora of work has been done on the anonymization of location information of mobile users [14, 17, 44, 56, 51, 50, 70, 69, 102]. While most existing solutions target spatial-temporal obfuscation techniques under the *random waypoint* mobility model [20, 66], where mobile users can move in arbitrary directions at random speed, they fail to address the vulnerabilities of mobile users traveling over roads, where both the user mobility and the location-based service processing are constrained by the underlying road networks.

More specifically, the protection enough under the random waypoint model may be insufficient under the *network-constrained* mobility model. For example, the spatial cloaking techniques protect users' privacy by blurring their exact location information with cloaked spatial areas, and measure the amount of protection as the area size. Such measurement however, is inapplicable under the road network model, since a large area may contain a single road segment, which enables the adversary to track down the mobile user fairly easily. Furthermore, the condition of the road networks, e.g., the network topology, has considerable impact on the query processing and communication efficiency, which should be a critical concern for developing location privacy solutions. For instance, the complexity of computing the network distance of two objects, a most fundamental operation in location-based query processing, varies significantly with the underlying network structures.

In this work, we present a general framework for location privacy protection under the network-constrained mobility model. Compared with prior work, our framework highlights three distinct features.

First, we argue that the protection for mobile users' privacy should be provided along two orthogonal dimensions: 1) *location anonymity*, which advocates that it should be difficult to identify a specific user among a set of users, the anonymous set, based on their location information; and 2) *location diversity*, which promotes that it should be hard to link a specific user with a specific location (such as a road segment) with high certainty. Furthermore, such privacy requirements should be customizable and supported on a per query basis. In this following, we refer to the process of achieving location-anonymity and location-diversity as *location anonymization*.

Second, we regard the attack resilience of the performed anonymization and the cost of processing queries with anonymous location information (including both computation and communication costs) as two critical measures for designing location privacy solutions. We propose corresponding analytical models to quantitatively

evaluate the efficacy of location anonymization models. In particular, we reveal the inherent tradeoff between these two metrics through a formal study of two basic anonymization models.

Third and most importantly, we present XStar, a novel star graph based location anonymization model, to achieve the optimal tradeoff between high query processing efficiency and strong inference attack resilience. To the best of our knowledge, this is the first model that takes account of both measures. In implementing XStar, we further introduce a suite of novel optimizations to enhance its performance. Extensive experimental evaluation is carried out to validate the analytical models and the efficacy of XStar.

The remainder of the chapter is organized as follows. In Section 6.2, we introduce fundamental concepts and models, and discuss the design objectives of location privacy solutions; Section 6.3 describes in detail the design of XStar, a novel star graph based anonymization model; The theoretical analysis of XStar in terms of query processing cost and inference attack resilience are presented in Section 6.4 and Section 6.5, respectively; Section 6.6 addresses detailed issues of implementing XStar and proposes multi-folded optimization strategies; The proposed solution is empirically evaluated in Section 6.7; Section 6.8 surveys relevant literatures.

6.2 Overview

In this section, we intend to give an overview of the design of XStar. We start with introducing the concept of road network aware location privacy, and then present our model of anonymous query processing; through an analysis of two extreme anonymization models in terms of attack resilience and query processing cost, we highlight our design objectives of location privacy solutions; finally, we outline the design of the XStar location anonymization model.

6.2.1 Road Network Model

In this chapter, we model a road network as an un-directed graph $G = (\mathcal{V}_G, \mathcal{E}_G)$, with the node set $\mathcal{V}_G$ and the edge set $\mathcal{E}_G$ representing road junctions and direct road links, respectively. An example of the road network model is shown in Figure 42. We use $\mathsf{d}_G(n)$ to denote the degree of a node n with respect to the graph G. Specifically, n is called an *intersection node* if $\mathsf{d}_G(n) \geq 3$, an *intermediate node* if $\mathsf{d}_G(n) = 2$, and an *end node* if $\mathsf{d}_G(n) = 1$.

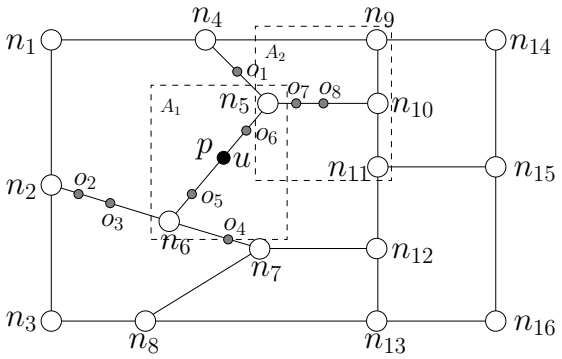


Figure 42: A road network model.

To model the restriction of users' mobility by the underlying road network, we introduce the concept of *segment*: a segment s is a sequence of edges $(\overline{n_0n_1}, \overline{n_1n_2}, \ldots, \overline{n_{L-1}n_L})$[1] where $\{n_i\}_{i=0}^{L}$ are all distinct, and the degrees of the nodes satisfy $\mathsf{d}_G(n_i) \geq 3$ for $i = 0$ or L, and $\mathsf{d}_G(n_i) = 2$ otherwise, i.e., n_0 and n_L are intersection or end nodes, all others are intermediate nodes.

Note that each edge is either a segment itself, or belongs to a unique segment, i.e., a road network can be uniquely partitioned into a set of segments. Hence, we assume the following scenario: every mobile user registered with LBS is moving along certain road segment, and sends her location-based query together with the information of her current position to the LBS provider, which then executes the query based on the provided location information.

[1]Without ambiguity, below we use the sequence of nodes $\overline{n_0 \ldots n_L}$ to denote the segment $(\overline{n_0n_1}, \overline{n_1n_2}, \ldots, \overline{n_{L-1}n_L})$.

6.2.2 Location Privacy Model

We consider two types of privacy concerns arising in LBS under the road network mobility model, namely location anonymity and location diversity. The first requirement ensures the indistinguishability of a specific mobile user among a set of users (anonymous set), and is usually captured by the *location k-anonymity* model [56, 50].

Definition 39 (Location k-anonymity)**.** *The published location of a user is said to be k-anonymous, if there at at least* $(k-1)$ *other active users with the same published location.*

However, ensuring location k-anonymity alone (the goal of most prior work [14, 56, 51, 102]) does not provide sufficient protection when the underlying road network is taken into consideration. For example, in Figure 42, assume that users $\mathtt{u}_1$ and $\mathtt{u}_2$ publish their k-anonymous location as A_1 and A_2, respectively. Given that A_1 and A_2 are of same size and contain same number of active users, $\mathtt{u}_1$ and $\mathtt{u}_2$ are considered to enjoy equivalent amount of privacy protection, under the criterion of location k-anonymity. However, it is much easier for the adversary to track down $\mathtt{u}_1$ than $\mathtt{u}_2$, since $\mathtt{u}_1$ is associated with a single road segment $\overline{n_5 n_6}$, while $\mathtt{u}_2$ is possibly associated with three. Intuitively, from the view of the adversary, the difficulty of tracking a user is in proportion to the number of segments that she is possibly associated with. This motivates us to introduce location diversity [98] as the second dimension of privacy measurement.

Definition 40 (Segment l-diversity)**.** *The published location of a user is said to be l-diverse, if it satisfies location k-anonymity, and contains at least l different road segments.*

In this framework, every mobile user $\mathtt{u}$ specifies customized privacy requirements as $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}})$ in terms of k-anonymity and l-diversity on a per query level. Besides, to guarantee the quality of received services, e.g., response time, each user $\mathtt{u}$ may also

specify customized QoS requirements as maximum spatial tolerance $\sigma_s^{\mathtt{u}}$ and temporal tolerance $\sigma_t^{\mathtt{u}}$: the spatial tolerance bounds the expansion of the anonymized location, while the temporal tolerance specifies the tolerable delay due to the anonymization operation (if a request could not be honored within $\sigma_t^{\mathtt{u}}$, it is typically discarded).

To fulfill such requirements, we introduce the operation of *location anonymization*:

Definition 41 (Location Anonymization)**.** *Let* $\mathtt{q}$ *denote a location-based query issued by a mobile user* $\mathtt{u}$*. Location anonymization transforms the exact location information associated with* $\mathtt{q}$ *to some approximate version that satisfies both privacy and QoS requirements as specified by* $\mathtt{u}$*.*

With road networks as the background context, we assume that the anonymization operation is performed on the basis of road segments, and an anonymous location is composed by a set of segments.

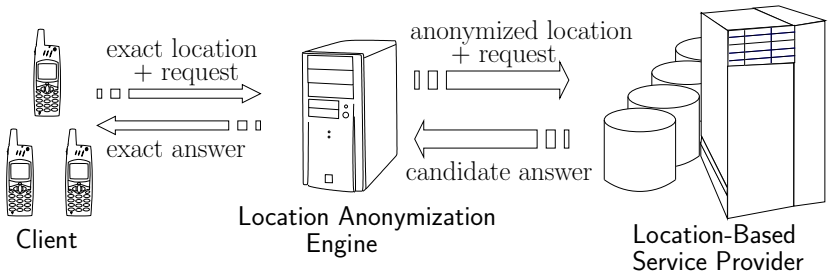


Figure 43: Overall architecture of XStar.

Furthermore, in this work, we assume a trusted, third party *location anonymization engine* that acts as a middle layer between mobile users and the LBS provider, and performs location anonymization. Specifically, it is responsible for (1) receiving the query and the exact position information from the mobile user; (2) anonymizing the location information according to the user's privacy requirements, and relaying it to the LBS provider; (3) generating the exact query result from the candidate answer returned by the service provider by properly filtering false positive information, and (4) delivering the exact answer to the client. Figure 43 sketches the overall

architecture of this framework.

6.2.3 Anonymous Query Processing Model

Now we consider the processing of queries with anonymous location information (a set of segments). Without loss of generality, we concentrate our discussion on k nearest neighbors (k-NN) style queries, in which the user requests for the k objects of interest with the minimum distance to her current position, called the *query focal point*. The distance between two points on the road network is defined as the length of their shortest path.

Intensive research has been directed to the query processing for road network databases recently [28, 63, 78, 103, 108]. While the proposed approaches differ in assumptions and techniques, we abstract two fundamental operations underlying these approaches to build our model, which is therefore generally applicable for most state-of-the-art frameworks: 1) segment-based operation, which takes the query $\mathtt{q}$ and a segment s as input, and returns the set of objects on s that satisfy the query condition, denoted by $\mathcal{O}(\mathtt{q}, s)$; 2) node-based operation, which takes $\mathtt{q}$ and a node n as input, and returns the set of objects in the vicinity of n that satisfy the query condition, denoted by $\mathcal{O}(\mathtt{q}, n)$.

We base our basic query processing model on the following theorem (the proof is omitted due to the space constraint).

Theorem 17. *For a k-NN query $\mathtt{q}$ with query point p on a segment s, with n_0^s and n_1^s as the two ends of s, the query result $\mathcal{R}(\mathtt{q}, p)$ satisfies the following condition:*

$$\mathcal{R}(\mathtt{q}, p) \subseteq \mathcal{O}(\mathtt{q}, s) \cup \mathcal{O}(\mathtt{q}, n_0^s) \cup \mathcal{O}(\mathtt{q}, n_1^s)$$

This theorem amounts to saying that the result of $\mathtt{q}$ must be included in the union of the following two sets of objects: (1) the objects of interest on s, and (2) the k nearest objects of interest to the two nodes n_0^s and n_1^s. An example is given in

Figure 42. The user u issues a k-NN query $\mathtt{q}$ with $k = 3$ while moving on the segment $\overline{n_5 n_6}$. It is clear that the exact answer $\mathcal{R}(\mathtt{q}, p) = \{o_5, o_6, o_7\}$ appears in the union of $O(\mathtt{q}, \overline{n_5 n_6}) = \{o_5, o_6\}$, $\mathcal{O}(\mathtt{q}, n_5) = \{o_1, o_6, o_7\}$ and $\mathcal{O}(\mathtt{q}, n_6) = \{o_5, o_3, o_4\}$.

Hence, given a location-based query $\mathtt{q}$ with its query point on a segment s, the processing of $\mathtt{q}$ comprises one segment-based operation with respect to s, and two node-based operations with respect to n_0^s and n_1^s. We now extend this model to the case of anonymous queries involving multiple segments. We first introduce the concept of boundary nodes.

Definition 42 (Boundary Node)**.** *Given a set of segments S in the road network G, the set of boundary nodes of S, denoted by $\mathcal{BV}_S$, is defined as:*

$$\mathcal{BV}_S = \{n | n \in \mathcal{V}_S, \mathtt{d}_G(n) > \mathtt{d}_S(n)\}$$

That is $\mathcal{BV}_S$ are those nodes in S that are connected to the rest of the network. For instance, for the set of segments $S = \{\overline{n_2 n_1 n_4}, \overline{n_2 n_6}, \overline{n_2 n_3 n_8}\}$ in Figure 42, its boundary node set is given as $\mathcal{BV}_S = \{n_4, n_6, n_8\}$.

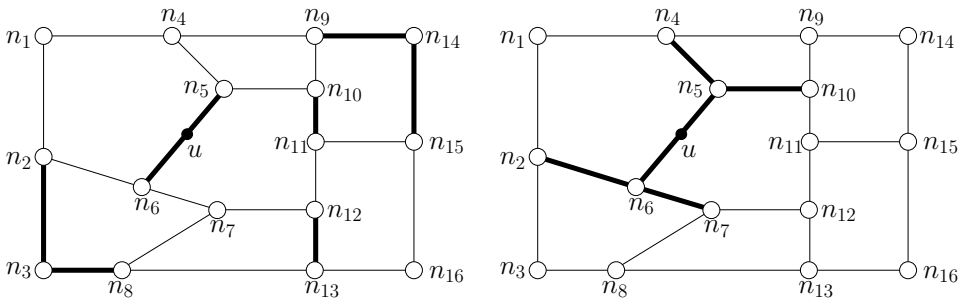
For a query $\mathtt{q}$ with associated anonymous location as a set of segments S, the evaluation of $\mathtt{q}$ consists of two parts: (1) the objects of interest on the segments of S, i.e., $\cup_{s \in S} \mathcal{O}(\mathtt{q}, s)$; and (2) the results as $\mathtt{q}$ issued on the boundary nodes of S, i.e., $\cup_{n \in \mathcal{BV}_S} \mathcal{O}(\mathtt{q}, n)$. Formally,

$$\mathcal{R}(\mathtt{q}, S) \subseteq (\cup_{s \in S} \mathcal{O}(\mathtt{q}, s)) \cup (\cup_{n \in \mathcal{BV}_S} \mathcal{O}(\mathtt{q}, n))$$

6.2.4 Two Motivating Schemes

The focus of this work is to develop robust and scalable location anonymization model. While multiple models are available to perform location anonymization to meet users' privacy requirements, we are interested in the optimal one that leads to (1) low query processing cost and (2) high attack resilience. Below we present two motivating models, based on *random sampling* and *network expansion*, respectively,

and achieve either extreme of the spectrum, which motivate our star-graph based model.



(a) Random Sampling (b) Network Expansion

Figure 44: Two naïve location anonymization models.

Random Sampling

Given a query $\mathtt{q}$ issued by a user $\mathtt{u}$ with privacy profile $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}})$, and maximum spatial tolerance $\sigma_s^{\mathtt{u}}$, at each iteration, this scheme samples one segment at random from the spatial region as defined by $\sigma_s^{\mathtt{u}}$ and adds it to $\mathtt{u}$'s anonymous location. The process continues until both requirements $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}})$ are satisfied. As an example, consider a user $\mathtt{u}$ in Figure 44(a) with $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}}) = (5, 5)$. With the RANDOM SAMPLING scheme, four segments are randomly selected (containing four active users $\{\mathtt{u}_i\}_{i=1}^4$), in addition to the original one that $\mathtt{q}$ is associated with, as represented by bold lines.

Network Expansion

At the other extreme of the spectrum, one can perturb $\mathtt{u}$'s location based on the network expansion scheme [108]: starting from the original segment which $\mathtt{u}$ is on, following Dijkstra's algorithm, one incrementally adds in neighboring segments, ordered by their network distances (mid points) to $\mathtt{u}$'s position. The process halts when $\mathtt{u}$'s privacy requirements are met. For example, as shown in Figure 44(b). the four segments with the minimum network distance to $\mathtt{u}$'s position are added incrementally to form $\mathtt{u}$'s anonymous location.

Discussion

It is observed that under the same requirements (δ_k, δ_l, σ_s), the RANDOM SAMPLING scheme results in a set of segments evenly distributed over the spatial region as defined by σ_s, which is essentially equivalent to issuing a set of queries at different locations, thus leading to high query processing cost. However, the strength of this scheme lies in its high resilience against inference attacks since the set of segments are selected at random.

In contrast, the NETWORK EXPANSION scheme results in a set of segments lying in a tightly compact structure. As will be theoretically proved in Section 6.4, such structure leads to the minimal query processing cost (the number of boundary nodes grows sub-linearly with the number of segments), which is further reduced by the fact that the expanded network is a partial result in the query processing [108]. However, the NETWORK EXPANSION scheme suffers low attack resilience as the expansion process follows a best-first search strategy which can be potentially exploited by the adversary to perform a reverse-engineering attack.

6.2.5 XSTAR: A Star Graph Based Approach

Motivated by the strengths and weaknesses of the two schemes above, we develop XSTAR, a star-graph based location anonymization model, aiming at achieving an optimal balance between low query processing cost and high attack resilience. Specifically, XSTAR achieves this balance in two main phases: first, it groups neighboring queries into basic structures, called *cloaking star*. The goal of this phase is to carefully choose the set of stars that minimize the computation and communication costs; then it adjusts the cloaking stars resulted from the previous step, and merges neighboring stars into *super-star* structures if necessary, to fulfill the privacy requirement of each individual user.

Intuitively, in XSTAR, the low query processing cost (see Section 6.4) is guaranteed

by two main factors: (1) the cost-aware cloaking-star selection scheme, and (2) the compact structure of the anonymous location as resulted from the basic star-graph structure and the operation of merging neighboring stars. Meanwhile, the high attack resilience (see Section 6.5) is contributed by two factors: (1) employing the cloaking star as the basic unit of location anonymization, and (2) injecting randomness into the cloaking-star merging operation.

Furthermore, in implementing XStar (see Section 6.6), we propose a suite of multi-folded optimizations to improve its efficiency, and introduce sharing processing of multiple queries for the service provider, which leads to considerable performance enhancement.

6.3 Anatomy

The location anonymization operation of XStar is composed of two main phases, *cloaking-star construction* and *super-star construction*. Concretely, in the first phase, a set of neighboring queries are grouped into a cloaking-star structure resulted from a cost-aware selection procedure; in the second phase, the privacy requirements of individual users are imposed by merging a set of neighboring stars into a super-star structure. The details of the two phases are presented in Section 6.3.1 and Section 6.3.2, respectively.

6.3.1 Cloaking-Star Construction

We first introduce the concept of *cloaking star*, which serves as the fundamental structure for location anonymization in XStar.

Definition 43 (Cloaking Star)**.** *For an intersection node n in the road network G, the cloaking star ϕ_n is the subgraph of G consisting of n and all the segments adjacent on n.*

By this definition, every node n with $\mathsf{d}_G(n) \geq 3$ is associated with a unique star

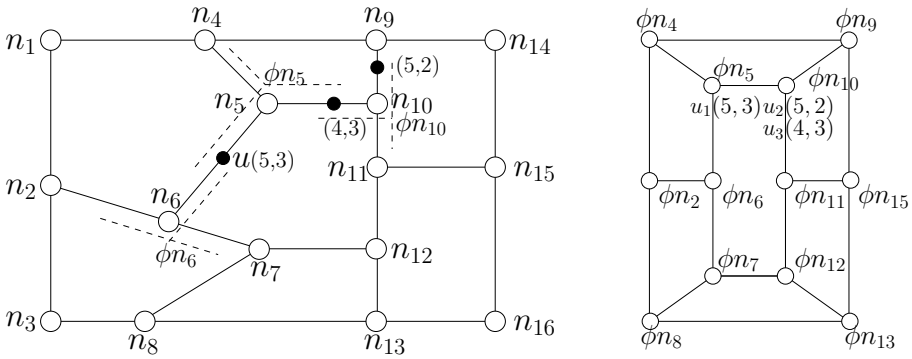


Figure 45: Illustration of XSTAR model.

ϕ_n. For example, in the left plot of Figure 45, the star ϕ_{n_5} comprises the node n_5 and the segments $\{\overline{n_5n_4}, \overline{n_5n_6}, \overline{n_5n_{10}}\}$.

The star structure possesses several desirable properties for our purpose: (i) it preserves the locality of neighboring segments, therefore employing it as the unit of anonymization is expected to lead to a compact structure of the anonymous location; (ii) it is amenable to indexing, since only the node identifier is needed to represent a star without information loss, hence using it to represent the anonymous location can reduce the communication cost, and ease the implementation.

Given a road network $G = (\mathcal{V}_G, \mathcal{E}_G)$, one can construct a corresponding *star network* $G_\phi = (\mathcal{V}_{G_\phi}, \mathcal{E}_{G_\phi})$, where each node of G_ϕ represents a star in G, and two nodes of G_ϕ are adjacent if their corresponding stars in G share a segment. The right plot of Figure 45 shows the star network corresponding to the left road network. Note that in G_ϕ, all the edges are of unit length. We define the distance between two stars ϕ_i and ϕ_j in a road network G as their network distance in G_ϕ, called *hop*, denoted by $\mathsf{h}_G(\phi_i, \phi_j)$. For example, in Figure 45, $\mathsf{h}_G(\phi_{n_6}, \phi_{n_{10}}) = 2$, as their shortest path in G_ϕ is composed of ϕ_{n_6}, ϕ_{n_5} and $\phi_{n_{10}}$.

A segment is marked as *active* if it is associated with at least one active query. To make our scheme resilient against the inference attacks (see Section 6.5), and amenable to the sharing processing of multiple queries (see Section 6.6), all the queries on the same segment share the same anonymous location.

If a star ϕ is chosen as the anonymous location for the queries on certain active segment s, it is said that ϕ is "selected", and s is "assigned" to ϕ, denoted by $s \leftarrow \phi$. Consider a segment s with two ends as n_0^s and n_1^s. If both $\mathsf{d}_G(n_s^s) \geq 3$ and $\mathsf{d}_G(n_t^s) \geq 3$ hold, then s is associated with two stars $\phi_{n_s^s}$ and $\phi_{n_t^s}$, e.g., ϕ_{n_5} and ϕ_{n_6} for $\overline{n_5 n_6}$ in Figure 45. In this case, it is to be determined to which star s should be assigned, $\phi_{n_0^s}$ or $\phi_{n_1^s}$. In a sequel, over the whole network, one need to choose a set of stars Φ to cover all the active segments.

To achieve low query processing cost, it is desired to incorporate the cost model into this selection phase. Formally, let $\mathsf{Cost}(\phi)$ be the cost of executing a typical query with the anonymous location as ϕ (as will be discussed in Section 6.4), AS denote the set of current active segments in the road network G, and Φ be the set of selected stars, then the minimization of the overall cost can be formalized as

$$\begin{aligned} \min_{\Phi} \quad & \textstyle\sum_{\phi \in \Phi} \mathsf{Cost}(\phi) \\ \text{s.t.} \quad & \forall s \in AS, \exists \phi \in \Phi, s \leftarrow \phi \end{aligned}$$

This cost-aware segment-to-star assignment scheme aims at finding a set of stars Φ that cover all the active segments, with the lowest overall cost. Note that this modeling ignores the numbers of queries on active segments for simplicity. However, as indicated in the empirical evaluation, it has captured the essential elements of the overall cost of query processing at the server, especially after introducing the machinery of *sharing processing of multiple queries* (see Section 6.6).

Nevertheless, no efficient solution to this optimization problem exists, unless P = NP, as shown in the next theorem (the proof is referred to our technical report).

Theorem 18. *Reductible from the weighted vertex cover problem, this cloaking star selection problem is NP-Hard.*

Consequently, instead of attempting to find a global optimal solution, we propose an efficient randomized algorithm that can find high-quality approximate solution,

and is robust against inference attacks. The procedure of inserting a new arrival query q associated with segment s (InsertQuery) is sketched in the following four cases: (1) if certain star already covers s, the algorithm halts; (2) if both stars $\phi_{n_0^s}$ and $\phi_{n_1^s}$ are already selected, yet s is not covered, one assigns s to one of the two stars with probability in reverse proportion to their corresponding costs; (3) if only one star $\phi_{n_s^s}$ or $\phi_{n_t^s}$ has been selected, s is assigned to that star; (4) if neither $\phi_{n_s^s}$ nor $\phi_{n_t^s}$ is selected, one assigns s to the one of them with probability reversely proportional to the corresponding cost.

Function `InsertRequest(`*a new request* q`)`

```
   // insertion of a new request
   // I_φ:active star index
1  s ← the segment containing q;
2  case s is already assigned to φ_{n_t^s} (or φ_{n_s^s})
3  |  return φ_{n_t^s} (or φ_{n_s^s});
4  case {φ_{n_s^s}, φ_{n_t^s}} ∩ I_φ = {φ_{n_s^s}, φ_{n_t^s}}
5  |  assign s to φ_{n_s^s} w.p. Cost(φ_{n_t^s}) / (Cost(φ_{n_s^s})+Cost(φ_{n_t^s})) or φ_{n_t^s} o.w.;
6  case {φ_{n_s^s}, φ_{n_t^s}} ∩ I_φ = φ_{n_s^s} (or φ_{n_t^s})
7  |  assign s to φ_{n_s^s} (or φ_{n_t^s});
8  case {φ_{n_s^s}, φ_{n_t^s}} ∩ I_φ = ∅
   |  // neither star is selected yet
9  |  if d_G(n_s^s) = 1 or d_G(n_t^s) = 1 then
   |  |  // only one end corresponds to a star
10 |  |  add φ_{n_t^s} (or φ_{n_s^s}) to I_φ;
11 |  |  assign s to φ_{n_t^s} (or φ_{n_s^s});
12 |  else
   |  |  // both ends are intersection nodes
13 |  |  assign s to φ_{n_s^s} w.p. Cost(φ_{n_t^s}) / (Cost(φ_{n_s^s})+Cost(φ_{n_t^s})) or φ_{n_t^s} o.w.;
14 |  |  add φ_{n_s^s} (or φ_{n_t^s}) to I_φ;
15 return φ_{n_t^s} (or φ_{n_s^s});
```

Intuitively, this approach ensures that each active segment s is assigned to $\phi_{n_0^s}$ with probability $\mathsf{Cost}(\phi_{n_1^s})/[\mathsf{Cost}(\phi_{n_0^s}) + \mathsf{Cost}(\phi_{n_1^s})]$, or $\phi_{n_1^s}$ otherwise. This property guarantees that the quality of the star set Φ selected by our randomized algorithm does not deviate far from the optimal one, as shown in the lemma below (the proof

is referred to our technical report):

Lemma 7. *Let* Cost^{opt} *be the cost achieved by the optimal star set. The randomized star-selection algorithm achieves the cost* Cost^{rnd} *satisfying* $\mathbb{E}\left[\mathsf{Cost}^{rnd}\right] \leq 2 \cdot \mathsf{Cost}^{opt}$.

It is also important to note that the quality of the selected stars does not degrade with continuous insertion/deletion of queries, given the fact that it makes no assumption about the order of the arrival of the queries, which is a desirable feature in supporting real-time road network based LBS.

6.3.2 Super-Star Construction

In the previous phase, a set of stars are selected to cover active segments, with the criteria of processing cost measure. In this phase, we fulfill the privacy requirements of mobile users. Concretely, this objective is achieved by merging a set of nearby stars to form a "super-star" structure, which then serves as the anonymous location for the queries inside.

Definition 44 (Super Star)**.** *A set of stars* $\{\phi_i\}_{i=1}^{|\psi|}$ *are said to form a super-star* ψ *if the subgraph consisting of* $\{\phi_i\}_{i=1}^{|\psi|}$ *is connected, where* $|\psi|$ *denotes the number of stars in* ψ *(the cardinality of* ψ*).*

As an example, in Figure 45, the user $\mathtt{u}_1$ is assigned to the star ϕ_{n_5}, while $\mathtt{u}_2$ and $\mathtt{u}_3$ assigned to $\phi_{n_{10}}$. However, the numbers of segments and active users in ϕ_{n_5} or $\phi_{n_{10}}$ alone do not satisfy users' privacy requirement (δ_l^u, δ_k^u) as $(5,3)$, $(5,2)$ and $(4,3)$, respectively. By merging ϕ_{n_5} and $\phi_{n_{10}}$, one obtains a superstar ψ that meets the requirements of all users involved.

The privacy profile of each user $\mathtt{u}$ considered in XStar is a tuple $(\delta_k^{\mathtt{u}}, \delta_l^{\mathtt{u}})$, which specifies the k-anonymity and l-diversity, respectively. Besides, to guarantee the service quality, e.g., to limit the query result size, each user $\mathtt{u}$ is also encouraged to specify the spatial and temporal tolerance $(\sigma_s^{\mathtt{u}}, \sigma_t^{\mathtt{u}})$, which indicate the bound on

the magnitude of the anonymous location, and the tolerable time delay due to the anonymization operation, respectively. Specifically, given a super-star ψ, σ_s^u is defined in terms of the hop count between the star covering u and the furthest star in ψ, and while a query can not be honored within σ_t^u, it is typically discarded.

Now we describe how to construct the super-star that satisfies the privacy profiles of all users involved in it. The procedure of merging stars to form super-stars (MergeStar) is sketched as follows: starting from an initial star ϕ, one applies a bottom-up aggregation, incrementally adding in neighboring stars until the privacy requirements of all users inside the super-star are satisfied, or the spatial tolerance of certain user is reached. Concretely, one first checks if the star ϕ already satisfies the privacy requirement of users in it; if not, one iteratively adds in neighboring active stars if possible. At each iteration, one first identifies all the neighboring stars whose merging with current superstar ψ do not violate the spatial tolerance of any user inside; if such star exists, one randomly picks one to merge with ψ to form a new super-star. This expansion process repeats until meeting the privacy requirements of all the users inside ψ, or reports failure (all the involved queries will be pushed back to the stack waiting for anonymization triggered by new arrival queries).

6.4 Model of Processing Cost

The cost of processing a location-based query consists of both the query execution cost at the LBS provider, and the communication cost of transferring the query results back to the mobile clients on the move. In this section, we establish an analytical model to compare the three alternative location anonymization models considered in this chapter from the perspective of query processing cost.

6.4.1 Measurement of Cost

Query Execution Cost

We use $\mathcal{C}_s$ and $\mathcal{C}_n$ to denote the computational cost (in terms of both CPU and IO) of a segment-based operation and a node-based one, respectively. Note that both $\mathcal{C}_s$ and $\mathcal{C}_n$ may vary from segment to segment and from node to node, depending on the condition of the road network (e.g., the density of objects), the predicate of each query (e.g., the parameter k of a k-NN query) and the system implementation (e.g., the performance of the look-up table). In the XStar prototype system, we set $\mathcal{C}_s$ and $\mathcal{C}_n$ statically for a typical setting. One direction of our ongoing research is to develop finer granularity and dynamic cost model.

Hence, for a typical query with a set of segments S as its anonymous location, the execution cost, $\mathsf{Cost}_{exec}(S)$, can be approximately estimated as follows: $\mathsf{Cost}_{exec}(S) = \mathcal{C}_n \cdot |\mathcal{BV}_S| + \mathcal{C}_s \cdot |S|$, where $|\cdot|$ denotes the cardinality of the set.

Communication Cost

Now we analyze the additional communication cost incurred due to the location anonymization operation. We measure the communication cost in terms of the length of the sent and received messages. Recall the XStar framework in Section 6.2, for given privacy requirement, i.e., stable number of segments in the anonymous location, the cost of sending/receiving the candidate answer becomes the dominating communication cost between the location anonymization engine and the LBS provider. Given a k-NN query $\mathtt{q}$ and a set of segments S as the anonymous location, $|\mathcal{R}(\mathtt{q}, S)|$ can be estimated as: $|\mathcal{R}(\mathtt{q}, S)| \approx k \cdot |\mathcal{BV}_S| + \sum_{s \in S} |\mathcal{O}(\mathtt{q}, s)|$, where the first component corresponds to the result size of issuing a k-NN query over each boundary node of S, and the second represents all the objects on the segments of S.

Let ρ_o denote the average number of objects on a segment, and $\mathcal{C}_o$ be the cost

of sending/receiving an object o. The communication cost for an anonymous k-NN query, with $\mathcal{S}$ as the anonymous location, is estimated as: $\mathsf{Cost}_{comm}(S) = \mathcal{C}_o \cdot [k \cdot |\mathcal{BV}_S| + \rho_o \cdot |S|]$.

6.4.2 Cost Analysis of Anonymization Models

Now we analyze the impact of the thee location anonymization models over the query processing overhead. For ease of exposition, we consider a uniform grid as the underlying road network, and assume that all the static objects of interest and active mobile users are distributed over the network with average density of ρ_o and ρ_u per segment. Consider a typical query $\mathtt{q}$ with (δ_k, δ_l) specified as k-anonymity, l-diversity, respectively. Thus its anonymous location comprises a set of segments S, with $|S| = \max(\delta_k/\rho_u, \delta_l)$. It is noticed that for fixed $|S|$, the size of the bound node set $|\mathcal{BV}_S|$ becomes the dominating factor in the two cost models above. Thus in the following, we focus on analyzing $|\mathcal{BV}_S|$ for each scheme.

Random Sampling

With the assumptions above, the RANDOM SAMPLING scheme results in a set of segments S with a boundary node set of size $2 \cdot |S|$ in the worst case, where no two selected segments are adjacent. Clearly, both the query execution and the communication costs grow linearly with the number of segments $|S|$.

Network Expansion

In contrast, the NETWORK EXPANSION scheme results in a set of segments S with $|\mathcal{BV}_S| = \sqrt{|S| + 2}$ in the worst case. Given the fact that the cost of node-based operation usually dominates the computation, i.e., $\mathcal{C}_n \gg \mathcal{C}_s$, here the execution cost grows sub-linearly, square-root-wise, more precisely, with the cardinality of S.

XStar

We have shown in Section 6.3.1 that the cost-aware star selection scheme guarantees the low overall cost of the set of cloaking stars. Here we concentrate our analysis on the second phase of XStar, super-star construction.

To be attack resilient, the MergeStar operation picks neighboring active stars at random, without considering any cost metrics. However the resulted anonymous location, usually exhibits fairly desirable properties in terms of query processing costs for two reasons: (1) the star structure preserves the locality of neighboring segments, i.e., in a star, the number of boundary nodes is no more than that of the segments involved; (2) the star selected at each iteration must satisfy the user-specified maximum spatial tolerance requirement, therefore leading to a highly compact super-star structure.

It can be proved that the XStar scheme produces a super-star ψ (with segments as $|S|$) with $|\mathcal{BV}_S|$ no more than $(2|S|+4)/3$, i.e., approximately one third of that under the Random Sampling model. Also note that this worst case occurs only at the extreme case where the stars forming ψ lay in a chain structure, with probability lower than $(1/4)^{|S|/3}$. In real applications, as verified by our experiments, XStar usually leads to anonymous location with quality comparable to that achieved by the Network Expansion model.

6.5 Model of Inference Attack

The obfuscation of the original location information is only a part of the story, one needs to consider the resilience of the anonymization against the adversary's attack: based on her prior knowledge or understanding regarding the working anonymization model, the adversary attempts to reveal users' original location through the blurred information. For the ease of presentation, in the following discussion, we assume a homogeneous road network, e.g., a grid.

Given the anonymous location as a set of segments S, the ideal protection is achieved if each segment is indistinguishable to the adversary, i.e., from her perspective, the mobile user is associated with each segment in S with equal probability $1/|S|$. However, with effective attacks, the adversary can identify that the association between u and a specific segment $s \in S$ has much higher probability than $1/|S|$, thus revealing u's private location with high confidence. We capture such vulnerability using the notion of *Linkability*:

Definition 45 (Linkability)**.** *For a user $\mathfrak{u}$ with original location as segment s^* and anonymous location as a set of segments S. The linkability $\mathsf{Link}[\mathfrak{u} \leftarrow s^*|S, K_{ad}]$ is defined as the probability that an adversary can infer that $\mathfrak{u}$ is associated with s^* based on S and her background knowledge K_{ad}.*

In particular, the background knowledge K_{ad} considered in this work includes (1) the location anonymization model, (2) the underlying road network structure, and (3) the estimation of query processing cost for every cloaking star (for XStar only). Following, we present a general *Replay Attack* model, which serves to measure attack resilience of the three location anonymization model.

6.5.1 Replay Attack

In a replay attack, for each segment $s \in S$, by re-running the anonymization algorithm with s assumed to be the original location, the adversary estimates the likelihood of s to generates the anonymous location S, $\mathsf{Like}[S|\mathfrak{u} \leftarrow s, K_{ad}]$. Under this model, the linkability is calculated as:

$$\mathsf{Link}[u \leftarrow s^*|S, K_{ad}] = \frac{\mathsf{Like}[S|\mathfrak{u} \leftarrow s^*, K_{ad}]}{\sum_{s \in S} \mathsf{Like}[S|\mathfrak{u} \leftarrow s, K_{ad}]}$$

Specifically, we assume that the adversary has full knowledge regarding the location anonymization algorithm, $\mathcal{A}(\cdot)$, which takes a segment s as input, and generates a set of segments as the anonymous location for s. Therefore, she is able to *replay* the

anonymization process: for each $s \in S$, (1) run $\mathcal{A}(s)$, and generate a set of segments S', with $|S'| = |S|$; (2) compute the likelihood $\mathsf{Like}[S, |\mathsf{u} \leftarrow s, K_{ad}] = |S' \cap S|/|S|$; (3) pick the segment s^+ that leads to the largest likelihood value as the original location, $s^+ = \arg\max_s \mathsf{Like}[S|\mathsf{u} \leftarrow s, K_{ad}]$.

6.5.2 Analysis of Attack Resilience

In this section, we evaluate the attack resilience of the three anonymization models with respect to the replay attack.

Random Sampling

In this scheme, the extra $(|S|-1)$ segments are selected at random. Therefore following the replay attack model, the adversary will find that each segment $s \in S$ can generate S with identical probability, i.e., $\max_s \mathsf{Link}[\mathsf{u} \leftarrow s|S, K_{ad}] = 1/|S|$, which implies the possible strongest protection.

Network Expansion

In this scheme, the $(|S| - 1)$ extra segments are expanded from the original one s^*, based on their network distance to s^*. Under the replay attack model, the adversary runs the network expansion algorithm for each $s \in S$. Clearly, s^* will generate $S' = S$, i.e., the highest likelihood, while other segments tend to result in likelihood no larger than s^*, therefore highlighting s^* as the expansion source. This is empirically verified in our experiments.

XStar

Recall that given the original segment s^*, the first phase of XStar generates a star ϕ covering s^*, and the second phase expands from ϕ and produces a super-star ψ.

Assume that ϕ consists of the segments $\{s_i\}_{i=1}^n$, with the corresponding neighboring stars as $\{\phi_i\}_{i=1}^n$. According to the design principle of our star selection scheme, the

likelihood that $\mathtt{u}$ is associated with s_i, given ϕ as $\mathtt{u}$'s anonymous location, $\mathsf{Like}[\phi, |\mathtt{u} \leftarrow s_i, K_{ad}]$, can be calculated as:

$$\mathsf{Like}[\phi|\mathtt{u} \leftarrow s_i, K_{ad}] = \frac{\mathsf{Cost}(\phi_i)}{\mathsf{Cost}(\phi) + \mathsf{Cost}(\phi_i)}.$$

Furthermore, the posterior probability $\mathsf{Prob}[\mathtt{u} \leftarrow s_i|\phi, K_{ad}]$ is given by:

$$\mathsf{Prob}[\mathtt{u} \leftarrow s_i|\phi, K_{ad}] = \frac{\mathsf{Like}[\phi|\mathtt{u} \leftarrow s_i, K_{ad}]}{\sum_{j=1}^{n} \mathsf{Like}[\phi|\mathtt{u} \leftarrow s_j, K_{ad}]}$$

It is observed that the adversary can identify the association between $\mathtt{u}$ and a specific segment s_i with high probability only if the cost of ϕ_i and other stars are extremely biased, i.e., $\mathsf{Cost}(\phi_i) \gg \mathsf{Cost}(\phi_j)$ for all $j \neq i$, which however is unusual in real scenarios.

Now consider the second phase, provided the facts that (1) ψ is generated by randomly expanding from some initial star, and (2) all the stars in ψ contain active users, and each can initiate and lead to the construction of ψ. Without further knowledge, from the perspective of the adversary, $\mathtt{u}$ is associated with each star with identical probability. Formally, assume that ψ consists of the stars $\{\phi_i\}_{i=1}^{m}$. The probability that the adversary can infer that $\mathtt{u}$ belongs to ϕ_i given ψ follows: $\mathsf{Prob}[\psi|\mathtt{u} \leftarrow \phi_i, K_{ad}] = 1/m$.

Combining the results above, we can now estimate the linkability under the XStar model: $\mathsf{Link}[\mathtt{u} \leftarrow s^*|\psi, K_{ad}] = \sum_{\phi \in \psi} \mathsf{Prob}[\mathtt{u} \leftarrow s^*|\phi, K_{ad}] \cdot \mathsf{Prob}[\mathtt{u} \leftarrow \phi|\psi, K_{ad}]$, where $\mathsf{Prob}[\mathtt{u} \leftarrow s^*|\phi, K_{ad}] = 0$ if $s^* \notin \phi$.

The above analysis is empirically verified in Section 6.7. It is also shown that in real scenarios, the XStar model provides almost the same amount of resilience as the Random Sampling model against the replay attack.

6.6 Implementation

In this section, we deal with the issues in implementing XStar in the location anonymization engine (LAE), and propose multiple optimizations to improve its performance, bringing in considerable enhancement over the basic version.

6.6.1 Location Anonymization Engine

Algorithm 10: Location Anonymization Engine

```
   // Q_q:arrival-query queue, H_q:expiration heap
   // I_φ:active-star index, Q_φ:ready-star queue
1  while true do
2      Q_φ ← ∅;
       // purge of expired requests
3      while true do
4          q ← top entry of H;
5          if q not expired then break;
6          φ ← DeleteQuery(q);
           // φ still active
7          if φ ∈ I_φ then add φ to Q_φ;
       // insertion of new requests
8      if Q_q ≠ ∅ then
9          q ← first entry of Q_q;
10         φ ← InsertQuery(q);
11         add φ to Q_φ;
       // location anonymization
12     while Q_φ ≠ ∅ do
13         φ ← first entry of Q_φ;
14         MergeStar(φ);
```

Algorithm 10 sketches the main procedure of the engine, where the InsertQuery, DeleteQuery and MergeStar compose a complete framework of location anonymization. At each iteration, it first purges all expired requests, and pushes the affected active stars to the ready-star queue Q_ϕ to be processed (line 3-7); it then pops up one new request q from the query queue Q_q, and pushes the selected star ϕ to Q_ϕ (line 8-11); finally, it attempts to perform anonymization for each star in Q_ϕ (line 12-14).

6.6.2 Optimizations

Though simple to implement, the basic version of LAE introduced above suffers several drawbacks: (1) each request deletion operation results in a trial of anonymizing the affected star, without checking the success condition; (2) it attempts to anonymize

the affected star immediately after a new query is inserted. It is expected that a significant number of attempts would fail due to insufficient numbers of active users and segments to satisfy users' privacy requirement; (3) for each request, the anonymization process starts from the scratch in a bottom-up manner, thus incurring the scalability problem.

In this section, we present multi-folded optimizations to improve the success rate and the scalability of the location anonymization engine. Corresponding to the drawbacks above, we propose three optimization policies as below.

Lazy Update for Deletion

We devise this policy based on the following observation: for a star ϕ that is not anonymized successfully in the previous iteration, if no updates happen to other stars, then ϕ can possibly be anonymized only if updates occur to its profile parameters $(\delta_d^\phi, \delta_k^\phi, \sigma_s^\phi)$, which are defined as the maximum k-anonymity, maximum l-diversity, and minimum spatial tolerance values associated with the queries in ϕ, respectively. Therefore, we introduce the policy of *lazy update* for the deletion operation: for a query being deleted, one attempts to anonymize the affected star ϕ only if its profile $(\delta_d^\phi, \delta_k^\phi, \sigma_s^\phi)$ is updated.

Batch Insertion of Queries

To improve the success rate of the anonymization operation, at each iteration, one can insert a batch of new queries, i.e., waiting for a period of time, before beginning the anonymization process. Concretely, a parameter T_w is specified as the waiting time before performing the anonymization, and it can be adjusted to trade the average processing burden over the anonymization engine for the success rate of the operation. Also, T_w should be set according to users' maximum tolerable service delay. The optimal setting of T_w is empirically discussed in Section 6.7.

Early Failure Detection

The star-merging operation is costly in the sense that it may fail due to no enough segments or active users appearing in the neighborhood of the initial star ϕ. Thus it is desirable to maintain such statistical information, and stop the merging process early if detecting no enough number of active users or segments available.

Specifically, for each intersection node n, we maintain the number of active users $\mathsf{Num}_u(n, R)$ and segments $\mathsf{Num}_s(n, R)$ in the sub-network of radius R hops to n. The statistical information for multiple R's $\{R_i\}_{i=0}^{h}$ (with R_0 corresponding to the star centering n) can be maintained in order to achieve more effective detection, though at higher maintenance cost.

The cached information can be used in two ways: (1) on anonymizing an initial star ϕ, with center node n, and profile $(\delta_l^\phi, \delta_k^\phi, \sigma_s^\phi)$, check if $\exists i \in [0, h]$ such that (i) $R_i \geq \sigma_s^\phi$, and (ii) $\mathsf{Num}_u(n, R_i) < \delta_k^\phi$ or $\mathsf{Num}_s(n, L_i) < \delta_l^\phi$. If such i exists, then the merging process stops as failure. (2) On adding a new star ϕ' centering n' to the current super-star ψ, check if $\exists i \in [0, h]$, such that (i) $R_i \geq \sigma_s^{\phi'}$, and (ii) $\mathsf{Num}_u(n', R_i) < \max\{\delta_k^{\phi'}, \delta_k^\psi\}$ or $\mathsf{Num}_s(n', R_i) < \max\{\delta_l^{\phi'}, \delta_l^\psi\}$. If such i exists, then one can safely exclude ϕ' from the candidate list for expansion.

6.6.3 Multiple Queries Sharing

From the perspective of the query processing at the server, XStar enjoys two major advantages over conventional methods: (1) independence of the underlying implementation of the processing techniques. XStar can be optimized for specific implementation, as introduced in Section 6.4, by configuring the cost function according to the adopted models, e.g., solution indexing [78], network expansion [108], etc. (2) capability of sharing processing of multiple requests. It considers the possibility of sharing processing in the location anonymizing operation, by grouping queries with nearby locations together and perturbing their location as an entirety.

Concretely, given a set of k-NN style queries $\{\mathtt{q}_i\}_{i=1}^{t}$ sharing the same anonymous location ψ, having corresponding k's as $\{k_i\}_{i=1}^{t}$ and $k_i \leq k_j$ for $i < j$, one can first evaluate $\mathtt{q}_t$ and retrieve the top k_i objects of $\mathcal{O}(\mathtt{q}_t, n)$ as $\mathcal{O}(\mathtt{q}_i, n)$ for each $n \in \mathcal{BV}_\psi$ and $i \in [1, t-1]$. This principle developed for k-NN queries can easily be extended to other types of requests, e.g., range queries, which is omitted here due to the space limit.

6.7 Evaluation

In this section, we perform an empirical analysis of the location anonymization models proposed in the chapter. The experiments are designed to compare these models based on the following three metrics: 1) attack resilience. The performed protection should be robust against malicious inference attack, i.e., hard for the adversary to penetrate the protection to identify users' exact position with high certainty; 2) cost awareness. The privacy protection mechanism should not incur excessive system burden for either the service provider or the mobile clients, in terms of query processing and communication cost; 3) operation efficiency. The anonymization operation should be computationally efficient and scalable, and a location anonymization engine equipped with modest computational resources should be able to handle a large number of mobile users on continuous move, and support real time query processing.

6.7.1 Experimental Setting

All our experiments are performed over real road maps from areas of the United States: the first road map corresponds to the highways in the entire State of California [89], which contains 21,048 nodes and 21,693 edges; moreover, it is associated with a real dataset of 104,771 *points of interest* (POIs), as categorized into 62 classes, e.g., church, hospital, airport, etc., which we used as queried objects in our simulation; the second road map corresponds to the roads in the City of Oldenburg, which contains 6,105 nodes and 7,035 edges. Choosing these road maps, we intend to evaluate the

performance of location anonymization models for road networks at varying scales.

On these maps we simulated different traffic conditions using the Network-based Generator of Moving Objects by T. Brinkhoff [2], a state-of-the-art traffic simulator. We assign a same number (10,000) of moving objects to each map. Since the two maps are of significantly different scales, we intend to simulate high user density (rush hour) and low user density (non-rush hour) conditions, respectively. In each simulation, we defined two classes of moving objects, with speeds corresponding to slow (e.g., trucks) and fast (e.g., passenger cars) vehicles, respectively. With a randomly assigned probability, each vehicle generates a set of k-NN (or none) queries during the simulation, with the parameters specified as follows: 1) the requested number of nearest points of interest as k; 2) the category of the points of interest as c, e.g., church, hospital, etc.; 3) privacy requirements as k-anonymity (δ_k) and l-diversity (δ_l); and 4) service quality requirements as the spatial (σ_s) and temporal tolerance (σ_t). The values of each query are drawn independently following certain distributions, with parameters listed in Table 9. After issuing a query, the vehicle waits for some normally distributed inter-wait time γ, waiting for the request to be either answered or dropped, before issuing another service request.

Table 9: Default parameter setting for query generation. Note: all the parameters except c follow normal distributions; c follows a uniform distribution over the interval [0, 62]; the values of σ_t and γ are in the unit of second.

parameters	k	c	δ_k	δ_l	σ_s	σ_t	γ
mean	5	N/A	5	5	4	10	20
deviation	1	N/A	1.5	1.5	1	2	2

For the location anonymization engine, we implemented four different methods: Random Sampling (R), Network Expansion (E), basic XStar (X), and optimized XStar (OX). For the privacy-aware query processing server, two versions are developed, one with the machinery of multiple queries sharing processing (MQS), and

[2] http://www.fh-oow.de/institute/iapg/personen/brinkhoff

the other without MQS. Most of the algorithms are implemented in Java. The experiments are performed on a Linux box running 1.5Ghz CPU with 512 MB memory.

6.7.2 Experimental Results

Cost Awareness

Now, we proceed to evaluating the impact of the location anonymization operation over the service performance, i.e., the query execution cost and the communication cost, under varying setting of traffic condition, privacy (δ_k, δ_l) and service quality requirements (k, σ_s). Specifically, in terms of the query execution cost, we measure the average execution time of processing a query at the server side; while in terms of the communication cost, we measure the average number of objects returned in the candidate result for each query. Specifically, we used the road map of California and its associated dataset of points of interest in query processing.

We measured the query processing cost corresponding to the three anonymization models (R, E and X represent the Random Sampling, Network Expansion and XStar methods, respectively) in the case without Multiple Query Sharing Processing (MQS), and the XStar method with MQS policy ($\mathsf{X}^{\mathsf{MQS}}$), as the mean values of the parameters k, δ_l, δ_k and σ_s change within the intervals of [2, 10], [3, 15], [3, 15] and [1, 5], respectively. In each set of experiments, we fix three parameters and vary the last one. Figure 46 plots the average execution time of processing each anonymous query by the server, while Figure 47 row plots the average size of the candidate result returned by the server.

With respect to anonymous query execution time, it is observed that the R-scheme incurs the highest system overhead at the server among the three schemes and the X-scheme outperforms both R and E in most cases, which validates our theoretical analysis regarding its superiority in terms of query execution cost. Note that although working ideally in the homogeneous grid world (Section 6.4), the E-scheme ignores the heterogeneity of real road maps when selecting extra segments, leading to its

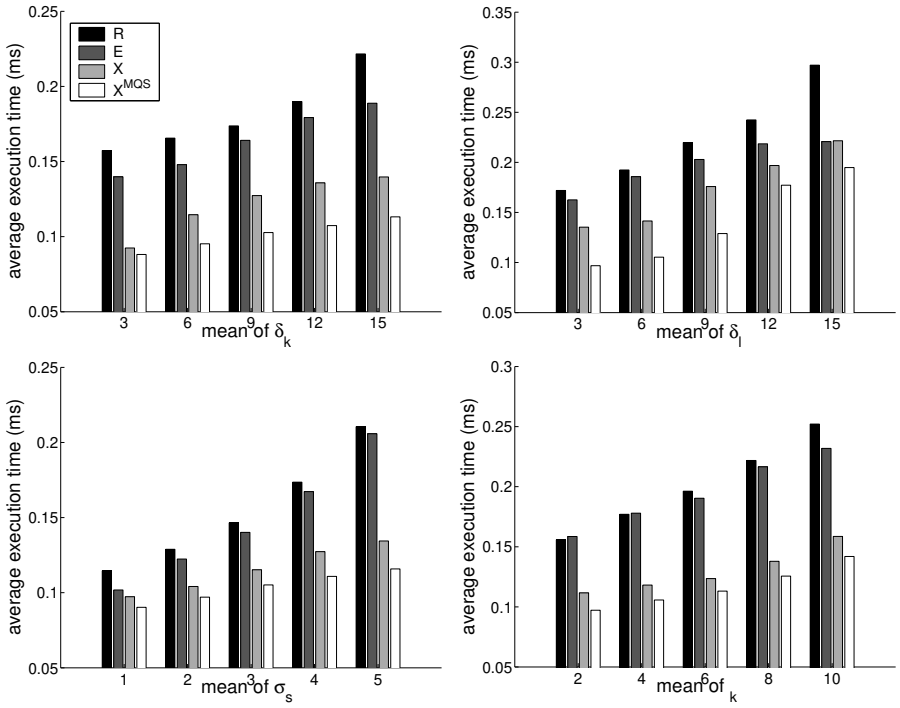


Figure 46: Average execution time per query with respect to varying settings of parameters δ_k, δ_l, σ_s, and k.

unsatisfying performance in real applications. Also notice that the MQS policy further improves query processing efficiency, and the improvement tends to become significant as the parameters δ_l, δ_k, k or σ_s increase. This can be explained by the facts that 1) stronger privacy requirement leads to larger number of queries to be anonymized in batches, therefore more queries can be processed in group; 2) a larger k results in higher execution cost for each individual query, but also lends more considerable savings for grouped queries; 3) a large spatial tolerance σ_s boosts the chance for a query to be successfully anonymized (to satisfy δ_k and δ_l) by allowing more queries to be grouped together.

Moreover, by examining the impacts of these four parameters over the query execution cost, one can notice that the parameters δ_k and δ_l have stronger influence than σ_s and k for all the anonymization models; this is contributed by the fact that stricter privacy requirements result in anonymous location of much coarser granularity (larger area), with its impact over the query execution easily exceeding that

exerted by the requested number of POIs or spatial tolerance. Meanwhile, it is also interesting to note that the performance of the X-scheme is fairly insensitive to the parameter setting: in all four cases, its increase is the least significant among all the anonymization models under consideration.

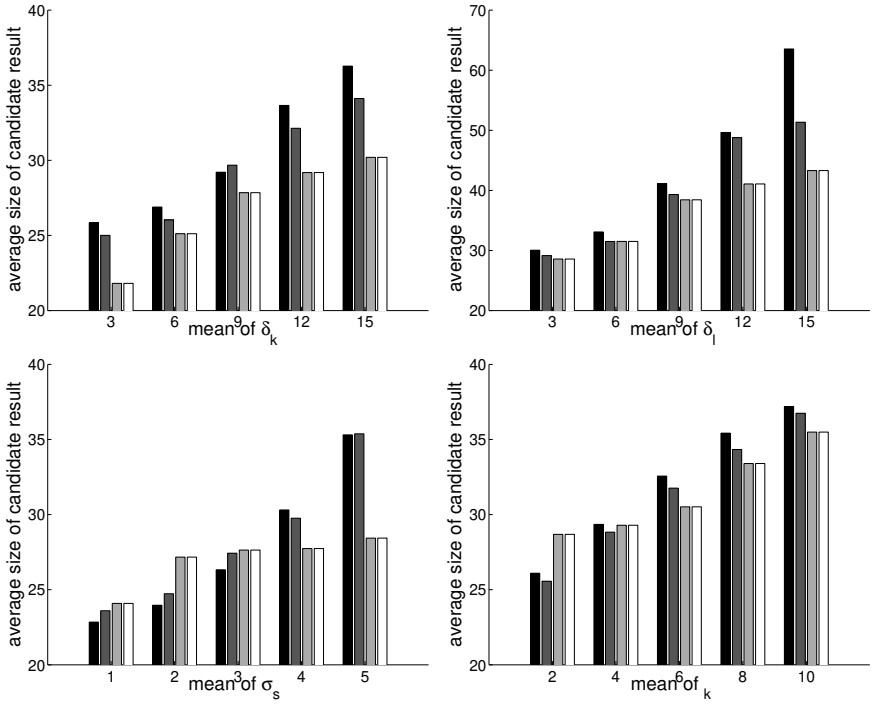


Figure 47: Average size of candidate result per query with respect to varying settings of parameters δ_k, δ_l, σ_s, and k.

With respect to communication cost, as expected, the R-scheme generates significantly larger size of candidate result than the other schemes (note that the multiple queries sharing processing does not affect the average size of candidate result; therefore, X and $\mathsf{X}^{\mathsf{MQS}}$ have the same size). Meanwhile, the X-scheme outperforms the other two for the cases of varying δ_k and δ_l, though the lead is not considerable as that in query execution time, especially for small δ_k and δ_l; in the cases of varying σ_s and k, it is interesting to observe that here both R and E schemes perform slightly better than the X-scheme for small σ_s and k. All these phenomenon is explained by the following facts: both R and E perform segment-based perturbation, which stop just after obtaining sufficient number of segments; meanwhile, X performs star-based

perturbation, and the generated anonymous location may include slightly more than enough number of segments (or nodes). This difference exhibits significantly when these parameters are small; however, as the privacy or service quality requirements grow, the inherent superiority of star-based perturbation dominates the performance.

Attack Resilience

In the first set of experiments, we take a close examination of the resilience of the anonymous locations generated by location anonymization models against malicious inference attacks. Specifically, we consider the replay attack as described in Section 6.5: given a set of segments S as the user $\mathtt{u}$'s anonymous location, the adversary attempts to estimate for each segment $s \in \mathcal{S}$ its probability to be associated with $\mathtt{u}$. We measure the strength of the privacy protection using the information entropy of the distribution of such probabilistic estimations: a larger entropy value indicates higher uncertainty for the adversary, i.e., better protection. We used both the California and Oldenburg road networks, aiming at capturing the influence of factors such as area scale, user density, etc.

The first set of results is illustrated in Figure 48, where we measured the information entropy of anonymous locations, with respect to varying δ_l and σ_s, two parameters relevant to the spatial expansion of anonymous locations. The left two plots correspond to the road network of Oldenburg, and the right two that of California. First notice that all the protection strengths of all the models increase as segment-diversity (δ_l) or spatial tolerance (σ_s) grows; intuitively, an anonymous location containing more segments tends to provide better protection. Also, as expected, under the replay attacks, the protection provided by the E-scheme is easy to penetrate, while the R-scheme demonstrates the best protection strength in most cases. The performance of the X-scheme is fairly stable, and its difference with the R-scheme tends to decrease as the number of segments increases. For the case of Oldenburg

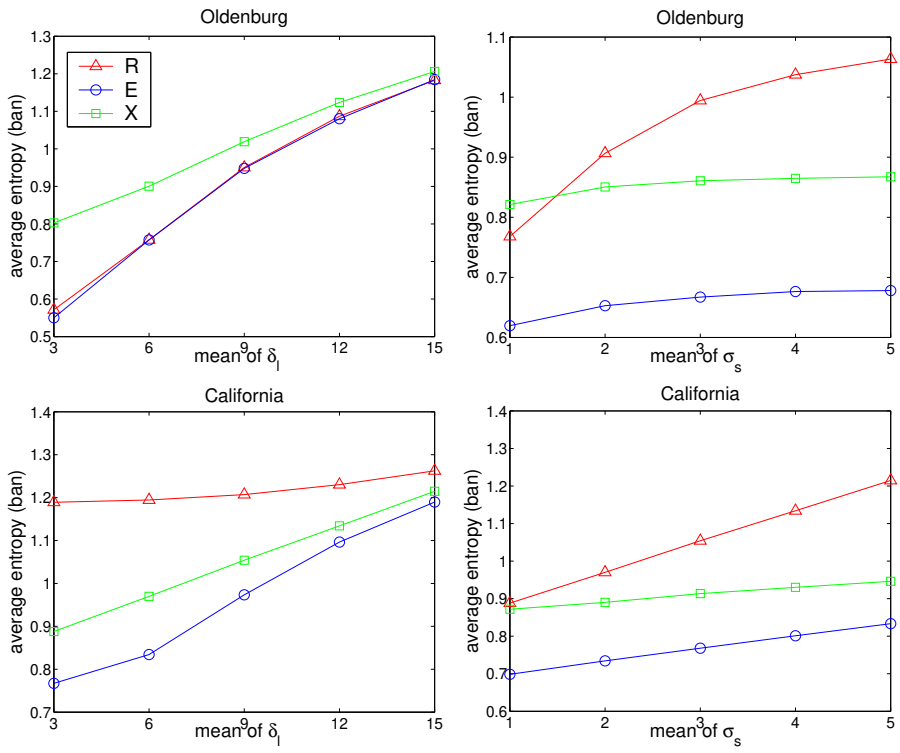


Figure 48: Average information entropy of anonymous locations generated by location anonymization models with respect to δ_l and σ_s, for maps of Oldenburg and California. Note: the entropy is in unit of ban (Hart).

road network (the leftmost plot), the entropy corresponding to the X-scheme is even higher than that provided by the R-scheme under varying setting of δ_l. This can be explained by that for a road network with sufficient user density, and for given privacy and service quality requirements, the anonymous location generated by a star-based perturbation scheme tends to feature higher segment-diversity than that produced by segment-based perturbation schemes, yet without compromising query processing efficiency (as shown above).

Also, note that the attack resilience discussed here focuses on the case of one-shot query. We anticipate that by combining the anonymous location information of multiple continuous queries of mobile users, the adversary can potentially infer more positioning information, which we consider as valuable research direction of our future work.

Operation Efficiency

The last set of experiments are designed to evaluate the operation efficiency of various location anonymization models. In particular, we are interested in two critical measurements: the success rate of the location anonymization operation and the average execution time of anonymization for each query. We incorporate the two measurements into a single metric, *successful throughput* (SF):

$$\text{SF} = \text{query arrival rate} \times \text{anonymization success rate}$$

That is, for given number of LBS requests, a higher successful throughput indicates better performance of the location anonymization engine.

In our experiments, specifically, in addition to the R, E, X anonymization schemes discussed so far, we implemented an optimized version of X-scheme, X* which incorporates the optimizations introduced in Section 6.6. We measured the successful throughput of these four location anonymization models as the functions of the parameters δ_k, δ_l, σ_s, and σ_t (the parameter k is not relevant to query anonymization).

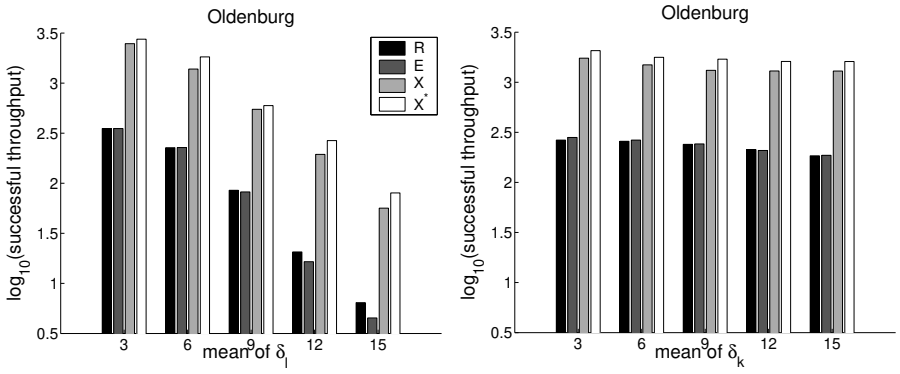


Figure 49: Successful throughput with respect to δ_l and δ_k.

The result is illustrated in Figure 49 and 50. Figure 49 shows the influence of the privacy parameters δ_k and δ_l over the SF of anonymization. Overall, it is noted that the performance of all four schemes tend to decrease as the privacy requirement becomes stronger. It is expected because 1) larger δ_l and δ_k are harder to satisfy, leading to higher rate of failure, and 2) moreover, even a successful attempt takes longer

execution time. The successful throughput of the X (and X*) is significantly higher than that of R and E, and the gap tends to grow as the privacy requirements become stricter. For example, even the basic X-scheme maintains throughput at about 56 for $\delta_k = 15$. This can be attributed to the strategy of star-based perturbation: compared with segment-based perturbation, which involves costly distance computation for node (or segment) pairs, the randomized star selection and merging operations are much less costly. It is also noted that the setting of δ_k has less impact than δ_l over SF. However, keep in mind that the Oldenburg road network features high user density; therefore, δ_k is easier to satisfy than δ_l.

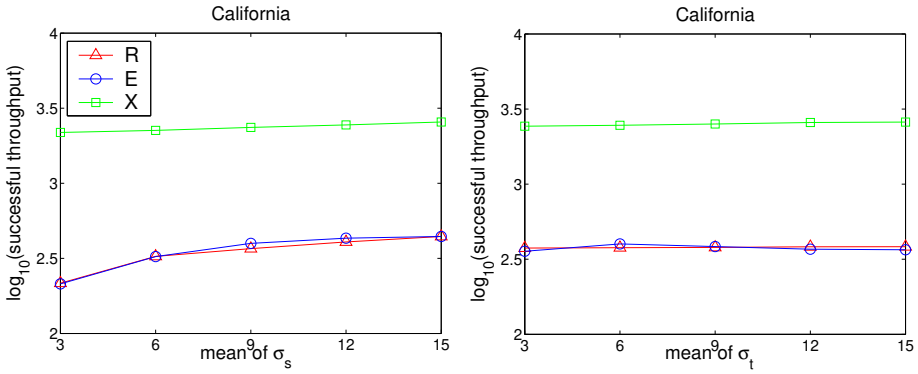


Figure 50: Successful throughput with respect to σ_s and σ_t.

Figure 50 demonstrates how the setting of spatial and temporal setting affect anonymization SF. Clearly, the SF of all anonymization models exhibit increasing trends as the spatial tolerances grows. This is expected, since larger tolerance increases the chance for a query to be successfully anonymized. Meanwhile, interestingly, the SFs of all the models stay fairly stable as the temporal tolerance changes. This may be explained as follows: longer lives of queries increase their chance to be successfully anonymized, i.e., a higher success rate of anonymization, but also increases computational overhead for the anonymization engine, i.e., a large stack of stale queries, which exist as two factors with countering influence over the successful throughput.

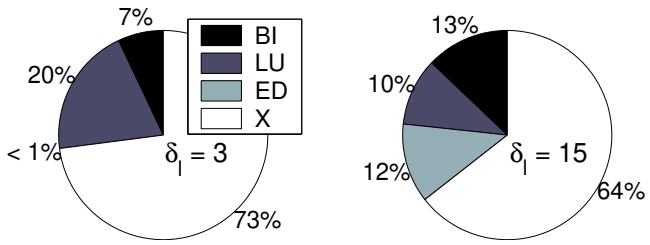


Figure 51: Fractions of improvement contributed by the multiple optimization strategies.

It is also interesting to analyze the contributions by the optimization strategies to the SF of X^*-scheme. The space limitation preclude the possibility of detailed analysis. Here, we take two snapshots of the fractions of improvement contributed by each strategy for $\delta_k = 3$ and 15, respectively. For brevity, we use the following short notations: *Lazy Update for Deletion* (LU), *Batch Insertion of Queries* (BI), and *Early Detection of Failure* (ED). Figure 51 shows the result, from which we can obtain the following observations. (1) The contribution from LU generally decreases as δ_l grows. This is explained by that as the success rate of anonymization decreases, more queries tend to be deleted, resulting in frequent changes to the privacy profiles of the corresponding stars; therefore, LU takes less effect in preventing false update. (2) The portion of (BI) stays stable as the parameters change. This is because that the insertion of more queries necessarily increases the average number of queries per star, thus improving the success rate of anonymization operation. (3) The fraction contributed by ED tends to increase with δ_l grows. This is due to that stronger privacy requirement brings higher failure probability for an anonymization operation, and FD can effectively counter its impact over the performance, by avoiding unnecessary attempts.

6.8 Related Work

Location privacy research is gaining increasing interests from both mobile networking community [17, 54, 44, 70, 69] and data management community [14, 56, 50, 51,

102]. The former has been focused on anonymous routing [54, 80], MIXes in mobile communication systems [44], source location privacy [70, 69] and Mix Zones [17, 44], Most existing solutions assume random waypoint mobility model [66] and utilize location hiding techniques to disable adversaries to associate location-based service requests (be it routing or query) with a particular mobile client.

Alternatively, location privacy research in data management community has been focused on extending data perturbation techniques developed for data privacy protection such as k-anonymity to address location privacy concerns, with examples including spatial cloaking [14, 56, 50, 51, 102], false dummies [76], and landmark objects [60]. Arguably, the most popular machinery of location privacy protection to date has been the spatial-temporal cloaking techniques, where the exact position of a mobile user is transformed to a spatial cloaking region. The transformation criterion has been solely based on location k-anonymity, and measures the amount of privacy protection in terms of the area size of the anonymous location. Unfortunately, as we have pointed out, most of the spatial cloaking algorithms tend to fail under the network-constrained mobility model, because the size of the cloaking area is no longer an effective and valid measure of the protection quality. The work [81] considers privacy-aware query processing for spatial networks; however, the adopted privacy protection model is still limited to spatial area cloaking.

Processing spatial queries over road network has been an emerging research topic recently [28, 62, 63, 78, 103, 108]. Two commonly used search paradigms have been proposed for k-NN style queries, namely incremental network expansion [108], and solution indexing [78]. The former gradually expands the search from the query point through the edges and reports the accessed objects during the expansion, while the latter precomputes and stores the solutions to the queries. Another line of research has been directed to continuous nearest neighbor (CNN) recently [63, 103], which returns both the k-NN objects and the valid scopes of the results along a path.

CHAPTER VII

CONCLUSION

This thesis centers around designing, building, analyzing and measuring massive data analytical systems wherein data come from multiple autonomous, yet correlated sources. Examples include network monitoring systems, mobile computing infrastructures and social networking platforms. Such systems hold the promise of supporting critical decision making by fusing information from different sources and providing consistent, multi-scale views of underlying phenomena. Today this promise remains largely unfulfilled, because building and operating large-scale multi-source analytical systems still face a multitude of challenges, from both system and data perspectives.

We focus on addressing the data-centric challenges facing system designers and operators, in particular including (i) *data correlation*: data may only consist of local, correlated observations of the global phenomena; and (ii) *data privacy*: data may be contributed by sources (e.g., individuals) that impose strict privacy requirements regarding their sensitive information. Although existing studies have explored to certain extent along each dimension separately, understanding the global phenomena as a function of a set of correlated observations in a privacy-preserving manner is a scope way beyond existing investigation. We propose Network-Aware Analysis and Privacy-Aware Analysis, two general design paradigms that seek to unleash the potential of multi-source analysis by making analytical systems more capable of weaving correlated observations into globally consistent pictures, and more privacy-preserving with respect to sensitive information,

Next we conclude this thesis and point to a set of future research directions.

7.1 Discussion

We begin with concluding our contributions in each concrete case study and discussing the limitations of our work.

7.1.1 Network-Aware Correlation Reasoning

Our work advances the state-of-the-art in network operational data analysis by presenting META, a general framework of learning, indexing and identifying topological and temporal correlation existing in network-event data, based on a novel class of network signatures. We present efficient learning algorithms to extract such signatures from noisy historical event data. With the help of novel indexing structures, we show how to perform efficient, online signature matching against high-volume event streams. While focusing on topological-temporal patterns only is unlikely to capture the myriad semantic structures existing in network-event data, we show that it is a powerful primitive to support a range of applications. Our experiments of deploying META with a large-scale testbed NMS to perform fault diagnosis show that META is able to perform scalable, yet robust correlation analysis. Our work also opens up several directions for further research: (i) incorporation of domain knowledge in training fault signatures; (ii) exploration of alternative models of temporal evolution, e.g., hidden Markov chains, frequent item-sets; (iii) search for data structures that can be incrementally adapted as network evolves; and (iv) incorporation of a richer set of topological relationships derived from multi-layer networks, e.g., mining information diffusion or providing risk-aware access-control in socio-information networks.

7.1.2 Network-Aware Causality Tracking

Our work advances the state-of-the-art in modeling the influence between the actions of individuals with social ties. We present a novel heat field over product network model (HFPN) that explicitly accounts for the action-sensitivity, time-sensitivity and user-sensitivity of such influence. We show that a broad range of applications, such

as resource recommendation and mobile phone call services, can be benefited by this model in terms of accuracy and freshness of customer action prediction. While our framework is rich and flexible, several key challenges need to be addressed before it can be readily adopted. (i) *Measures of social influence*. To meet our ambitious objective of informative action prediction, it may require to assign influence strength over social ties. Various measures can be used to derive influence strength; it is however challenging to quantify such measures and to identify the optimal one. (ii) *Network evolution*. While our model captures dynamic aspects including the shift of communities' and individuals' interests, as well as the dynamic nature of social influence, it assumes relatively static interconnections between users and objects. How to extend it to support social or object networks in rapid evolution? (iii) *Multi-type objects*. In this work we considered the interactions between users and one type of objects. What happens when different types of objects are present? How to transfer the knowledge for the interactions with one type of objects to another? Our work might be part of a temporary solution until more comprehensive models are available, and it might inform the design of these models.

7.1.3 Privacy-Aware Data Dissemination

We represent a systematical study on the problem of protecting general proximity privacy, with findings of broad applicability. Our contributions are multi-fold: we highlight and formalize proximity breach in a data-model-neutral manner; we propose a unified privacy definition, $(\epsilon, \delta)^k$-dissimilarity, with theoretically guaranteed protection against association attack in terms of both exact and proximate QI-SA association; we derive the criteria that enable to efficiently check the satisfiability of the principle for given microdata; we develop a novel anonymization model, XColor, to fulfill this principle, which offer flexible control over multiple privacy requirements.

Our work also opens several directions for future research. First, we define proximity breach as a set of SA-values proximate to a common one in a QI group (star structure). As other topological structures are taken into consideration, e.g., clique, chain and cycle, how to measure and remedy the possible privacy breach is an important problem. Second, in developing our solution, we only allow the vertices to be exchanged between two color classes. While allowing vertices to be transferred in a cycle, e.g., moving vertex v_1 from class V_1 to V_2, v_2 from V_2 to V_3, and v_3 from V_3 to V_1, the solution is still valid, and we envision a better bound than that provided in this work. Third, in our generalization framework, we apply a suite of heuristics to optimize the statistical utility of the resulted data (a best-effort strategy). It is worth investigating how to incorporate utility as a first-class citizen in designing the anonymization solution.

7.1.4 Privacy-Aware Data Mining

We highlight the importance of imposing privacy protection over (stream) mining output, a problem complimentary to conventional input privacy protection. We articulate a general framework of sanitizing sensitive patterns (models) to achieve output-privacy protection. We present the inferencing and disclosure scenarios wherein the adversary performs attacks over the mining output. Motivated by the basis of the attack model, we propose a lighted-weighted countermeasure, BUTTERFLY*. It counters the malicious inference by amplifying the uncertainty of vulnerable patterns, at the cost of trivial decrease of output accuracy; meanwhile, for given privacy and accuracy requirement, it maximally preserves the utility-relevant semantics in mining output, and thus achieving the optimal balance between privacy guarantee and output quality. The efficacy of BUTTERFLY* is validated by extensive experiments on both synthetic and real datasets.

7.1.5 Privacy-Aware (Location) Data Management

We present a systematic study of the problem of protecting location privacy under the network-constrained mobility model. We develop XStar, a general privacy-aware mobile service model, which exhibits three distinct features compared with prior work: (i) it supports road network specific and customizable privacy requirements on a per request level; (ii) it provides a careful tradeoff between the metrics of anonymous query processing cost and the attack resilience of users' hidden location information; (iii) it scales well to support a large number of mobile users with varying personalized privacy requirements, through a star-based anonymization technique, powered by multi-fold optimizations in implementation.

Our research will continue along several dimensions. First, we plan to develop finer granularity cost models for location-based query execution and communication, taking account of dynamic information, such as traffic condition, and complex road network semantics. Second, we will continue to study other types of inference attacks beyond the replay attack model, to evaluate and enhance the attack resilience of the XStar model. Finally, we are interested in extending the current framework to support continuous location-based queries, which are subjected to more complicated inference attacks than one-shot queries.

7.2 Future Directions

The explosive popularity of social media, Web 2.0 technologies, and cloud computing infrastructures are just a few examples that give a glimpse of the ever-increasing demand for large-scale, context-rich data analytical systems. Our future research will continue to address this demand by inventing new methods across the entire data collection, processing and dissemination cycle to improve data analytical systems in terms of analysis capability, system scalability and privacy preservation, from both theoretical and system perspectives.

7.2.1 Theoretical Models of Multi-Source Analytical Systems

While some fields of engineering have reached a level of maturity where there are concrete design principles that ensure a level of correctness and effectiveness, we are still lacking such a rubric for building data analytical systems. One of our research directions is to develop general design models and formal frameworks that can help system designer build multi-source data analytical systems with provable properties and theoretical backing. In particular, we intend to address a set of critical questions under different theoretical settings.

Model of Data Source

The locality and quality of the data from each source, and the correlation between these sources interact with each other in an intricate manner for designing analytical systems. For example, when injecting uncertainty at data sources according to their correlation (one of the most important privacy-preserving techniques), it is necessary to consider the inherent uncertainty (due to data quality issues) at each source. Thus, we intend to explore models that can incorporate these elements in a unified manner, and offer intuitive guidance for designing analytical systems.

Design of Analytical System

To ensure the correctness and effectiveness of analytical systems, it is imperative to understand systems' behavior under given conditions. For example, for given specification of data quality and locality at each source and the correlation between these sources, we intend to answer: What is the maximum achievable analyses accuracy? More importantly, what is the optimal strategy approaching this accuracy? How to construct systems that scale well with the number of data sources and the data volume of each source? What is the optimal strategy for each source to protect its privacy, while collaboratively achieving the maximum possible analyses accuracy for given privacy requirements?

Operation of Analytical System

The autonomous and dynamic nature of data sources make it necessary for analytical systems to adapt to evolving condition of input data. we intend to study the optimal adaptive strategies of analytical systems under various abruptiones. For example, in a monitoring system that relies on sampling (to save bandwidth) from multiple sources, in facing of the failure of specific sources, how to adaptively adjust the sampling rates at correlated sources to ensure the analyses accuracy?

7.2.2 Domain-specific Multi-source Analytical Systems

Another direction of our future research will focus on addressing challenges in domain-specific multi-source analytical systems in emerging contexts, including, for instance, social computing, mobile Internet, cloud and crowd computing. The design of analytical systems on these new platforms feature unique characteristics in terms of analyses tasks, performance requirements, and users' privacy needs. In the following we give three concrete examples in the domain of communication networks.

Utility-driven Network Trace Analyses

In the concrete application of network trace analyses, we face a range of tasks, which require different levels of detailed information, e.g., packet-level analyses to estimate port distributions, flow-level analyses to detect stepping-stone attack, network-level analysis to find anomaly phenomena. For such task variety, we desire the privacy-preserving solutions to provide corresponding level of sanitization, while being optimized for the intended task. we attempt to propose privacy solutions that achieve multi-level anonymization matching with the required granularity of intended tasks, and collusion resistance such that colluding data recipients cannot achieve any advantage.

Crowd-based Network Failure Detection

Today's network infrastructures still have difficulty to detect a large class of "silent" failures due to configuration errors, routing anomalies, and router bugs. One possible solution is to "outsource" the task of failure detection to the network ends where the services are received. In this model, the network ends function as event data sources as well as analytical systems. we intend to address the issues of accuracy/delay of failure detection, communication bandwidth, and privacy protection for each involved network entity.

REFERENCES

[1] Adam, N. R. and Worthmann, J. C., "Security-control methods for statistical databases: a comparative study," *ACM Comput. Surv.*, vol. 21, no. 4, pp. 515–556, 1989.

[2] Aggarwal, C. C., "On k-anonymity and the curse of dimensionality," in *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pp. 901–909, VLDB Endowment, 2005.

[3] Agrawal, D. and Aggarwal, C. C., "On the design and quantification of privacy preserving data mining algorithms," in *PODS '01: Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (New York, NY, USA), pp. 247–255, ACM, 2001.

[4] Agrawal, R., Borgida, A., and Jagadish, H. V., "Efficient management of transitive relationships in large data and knowledge bases," in *Proceedings of the 1989 ACM SIGMOD international conference on Management of data*, SIGMOD '89, (New York, NY, USA), pp. 253–262, ACM, 1989.

[5] Agrawal, R. and Srikant, R., "Fast algorithms for mining association rules in large databases," in *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, (San Francisco, CA, USA), pp. 487–499, Morgan Kaufmann Publishers Inc., 1994.

[6] Agrawal, R. and Srikant, R., "Privacy-preserving data mining," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pp. 439–450, 2000.

[7] Ahmad, Y. and Nath, S., "COLR-tree: communication-efficient spatio-temporal indexing for a sensor data web portal.," in *ICDE'08: Proceedings of the 24th IEEE International Conference on Data Engineering*, (Washington, DC, USA), IEEE Computer Society, 2008.

[8] Akaike, H., "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[9] Anagnostopoulos, A., Kumar, R., and Mahdian, M., "Influence and correlation in social networks," in *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, (New York, NY, USA), pp. 7–15, ACM, 2008.

[10] Asur, S. and Parthasarathy, S., "A viewpoint-based approach for interaction graph analysis," in *Proceedings of the 15th ACM SIGKDD international*

conference on Knowledge discovery and data mining, KDD '09, (New York, NY, USA), pp. 79–88, ACM, 2009.

[11] Atzori, M., Bonchi, F., Giannotti, F., and Pedreschi, D., "Anonymity preserving pattern discovery," *The VLDB Journal*, vol. 17, no. 4, pp. 703–727, 2008.

[12] Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J., "Models and issues in data stream systems," in *PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (New York, NY, USA), pp. 1–16, ACM, 2002.

[13] Backstrom, L., Huttenlocher, D., Kleinberg, J., and Lan, X., "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, (New York, NY, USA), pp. 44–54, ACM, 2006.

[14] Bamba, B., Liu, L., Pesti, P., and Wang, T., "Supporting anonymous location queries in mobile environments with privacygrid," in *Proceeding of the 17th international conference on World Wide Web*, WWW '08, (New York, NY, USA), pp. 237–246, ACM, 2008.

[15] Bayardo, R. J. and Agrawal, R., "Data privacy through optimal k-anonymization," in *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, (Washington, DC, USA), pp. 217–228, IEEE Computer Society, 2005.

[16] Belkin, M. and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, pp. 1373–1396, June 2003.

[17] Beresford, A. R. and Stajano, F., "Mix zones: User privacy in location-aware services," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW '04, (Washington, DC, USA), pp. 127–, IEEE Computer Society, 2004.

[18] Bogers, T. and van den Bosch, A., "Recommending scientific articles using citeulike," in *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, (New York, NY, USA), pp. 287–290, ACM, 2008.

[19] Bossomaier, T., "Linked: The new science of networks by albert-lászló barabási," *Artif. Life*, vol. 11, pp. 401–402, September 2005.

[20] Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C., and Jetcheva, J., "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '98, (New York, NY, USA), pp. 85–97, ACM, 1998.

[21] Bu, S., Lakshmanan, L. V. S., Ng, R. T., and Ramesh, G., "Preservation of patterns and input-output privacy," in *ICDE*, pp. 696–705, 2007.

[22] Calders, T., "Computational complexity of itemset frequency satisfiability," in *PODS '04: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, (New York, NY, USA), pp. 143–154, ACM, 2004.

[23] Calders, T. and Goethals, B., "Mining all non-derivable frequent itemsets," in *PKDD '02: Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, (London, UK), pp. 74–85, Springer-Verlag, 2002.

[24] Chen, B.-C., LeFevre, K., and Ramakrishnan, R., "Privacy skyline: privacy with multidimensional adversarial knowledge," in *Proceedings of the 33rd international conference on Very large data bases*, VLDB '07, pp. 770–781, VLDB Endowment, 2007.

[25] Chen, K. and Liu, L., "Privacy preserving data classification with rotation perturbation," in *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, (Washington, DC, USA), pp. 589–592, IEEE Computer Society, 2005.

[26] Chi, Y., Wang, H., Yu, P. S., and Muntz, R. R., "Catch the moment: maintaining closed frequent itemsets over a data stream sliding window," *Knowl. Inf. Syst.*, vol. 10, no. 3, pp. 265–294, 2006.

[27] Chin, F. Y. and Ozsoyoglu, G., "Statistical database design," *ACM Trans. Database Syst.*, vol. 6, no. 1, pp. 113–139, 1981.

[28] Cho, H.-J. and Chung, C.-W., "An efficient and scalable approach to cnn queries in a road network," in *Proceedings of the 31st international conference on Very large data bases*, VLDB '05, pp. 865–876, VLDB Endowment, 2005.

[29] Cohen, E., Halperin, E., Kaplan, H., and Zwick, U., "Reachability and distance queries via 2-hop labels," in *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '02, (Philadelphia, PA, USA), pp. 937–946, Society for Industrial and Applied Mathematics, 2002.

[30] Cohen, I., Zhang, S., Goldszmidt, M., Symons, J., Kelly, T., and Fox, A., "Capturing, indexing, clustering, and retrieving system history," in *Proceedings of the twentieth ACM symposium on Operating systems principles*, SOSP '05, (New York, NY, USA), pp. 105–118, ACM, 2005.

[31] Cohn, H., Kleinberg, R., Szegedy, B., and Umans, C., "Group-theoretic algorithms for matrix multiplication," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, (Washington, DC, USA), pp. 379–388, IEEE Computer Society, 2005.

[32] Cowen, L. J., Goddard, W., and Jesurum, C. E., "Coloring with defect," in *Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*, SODA '97, (Philadelphia, PA, USA), pp. 548–557, Society for Industrial and Applied Mathematics, 1997.

[33] Cox, L., "Suppression methodology and statistical disclosure control," *Journal of the American Statistical Association*, vol. 75, no. 370, pp. 377–385, 1980.

[34] Dalenius, T. and Reiss, S. P., "Data-swapping: A technique for disclosure control," *J. Statist. Plann. Inference*, vol. 6, pp. 73–85, 1980.

[35] de Gemmis, M., Lops, P., Semeraro, G., and Basile, P., "Integrating tags in a semantic content-based recommender," in *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, (New York, NY, USA), pp. 163–170, ACM, 2008.

[36] Dempster, A., Laird, N., and Rubin, D., "Maximum likelihood from incomplete data via the *em* algorithm," *Journal of The Royal Statistics Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[37] Denning, D. E., "Secure statistical databases with random sample queries," *ACM Trans. Database Syst.*, vol. 5, no. 3, pp. 291–315, 1980.

[38] Dobkin, D., Jones, A. K., and Lipton, R. J., "Secure databases: protection against user influence," *ACM Trans. Database Syst.*, vol. 4, no. 1, pp. 97–106, 1979.

[39] Dogear, "Lotus connections - dogear," *http://www.ibm.com/dogear*.

[40] Dwork, C. and Lei, J., "Differential privacy and robust statistics," in *Proceedings of the 41st annual ACM symposium on Theory of computing*, STOC '09, (New York, NY, USA), pp. 371–380, ACM, 2009.

[41] Eagle, N., Pentland, A. S., and Lazer, D., "Inferring friendship network structure by using mobile phone data," *Proceedings of the National Academy of Sciences*, vol. 106, no. 36, pp. 15274–15278, 2009.

[42] Evfimievski, A., Srikant, R., Agrawal, R., and Gehrke, J., "Privacy preserving mining of association rules," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 217–228, ACM, 2002.

[43] Fan, L., Cao, P., Almeida, J., and Broder, A. Z., "Summary cache: a scalable wide-area web cache sharing protocol," in *Proceedings of the ACM SIGCOMM '98 conference on Applications, technologies, architectures, and protocols for computer communication*, SIGCOMM '98, (New York, NY, USA), pp. 254–265, ACM, 1998.

[44] Federrath, H., Jerichow, A., and Pfitzmann, A., "Mixes in mobile communication systems: Location management with privacy," in *Proceedings of the First International Workshop on Information Hiding*, (London, UK), pp. 121–135, Springer-Verlag, 1996.

[45] Feldmann, A., Maennel, O., Mao, Z. M., Berger, A., and Maggs, B., "Locating internet routing instabilities," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, (New York, NY, USA), pp. 205–218, ACM, 2004.

[46] Fellegi, I. P., "On the question of statistical confidentiality," *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 7–18, 1972.

[47] Foxs-News, "Man accused of stalking ex-grilfriend with gps," *http://www.foxnews.com/story/0293313148700*, 2004.

[48] Fung, B. C. M., Wang, K., and Yu, P. S., "Top-down specialization for information and privacy preservation," in *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, (Washington, DC, USA), pp. 205–216, IEEE Computer Society, 2005.

[49] Garey, M. R. and Johnson, D. S., *Computers and Intractability; A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co., 1990.

[50] Gedik, B. and Liu, L., "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 1–18, January 2008.

[51] Ghinita, G., Kalnis, P., and Skiadopoulos, S., "Prive: anonymous location-based queries in distributed mobile systems," in *Proceedings of the 16th international conference on World Wide Web*, WWW '07, (New York, NY, USA), pp. 371–380, ACM, 2007.

[52] Giesbrecht, M., "Nearly optimal algorithms for canonical matrix forms," *SIAM J. Comput.*, vol. 24, pp. 948–969, October 1995.

[53] Goetz, M., Leskovec, J., McGlohon, M., and Faloutsos, C., "Modeling blog dynamics," in *International Conference on Weblogs and Social Media*, 2009.

[54] Goldschlag, D., Reed, M., and Syverson, P., "Onion routing," *Commun. ACM*, vol. 42, pp. 39–41, February 1999.

[55] Gomez Rodriguez, M., Leskovec, J., and Krause, A., "Inferring networks of diffusion and influence," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, (New York, NY, USA), pp. 1019–1028, ACM, 2010.

[56] Gruteser, M. and Grunwald, D., "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, MobiSys '03, (New York, NY, USA), pp. 31–42, ACM, 2003.

[57] Guan, Z., Wang, C., Bu, J., Chen, C., Yang, K., Cai, D., and He, X., "Document recommendation in social tagging services," in *Proceedings of the 19th international conference on World wide web*, WWW '10, (New York, NY, USA), pp. 391–400, ACM, 2010.

[58] Guttman, A., "R-trees: a dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, SIGMOD '84, (New York, NY, USA), pp. 47–57, ACM, 1984.

[59] Hidalgo, C. A. and Rodriguez-Sickert, C., "The dynamics of a mobile phone network," *Physica A: Statistical Mechanics and its Applications*, vol. 387, no. 12, pp. 3017–3024, 2008.

[60] Hong, J. I. and Landay, J. A., "An architecture for privacy-sensitive ubiquitous computing," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, (New York, NY, USA), pp. 177–189, ACM, 2004.

[61] Hore, B., Mehrotra, S., and Tsudik, G., "A privacy-preserving index for range queries," in *VLDB '04: Proceedings of the Thirtieth international conference on Very large data bases*, (Toronto, Canada), pp. 720–731, VLDB Endowment, 2004.

[62] Hu, H., Lee, D. L., and Lee, V. C. S., "Distance indexing on road networks," in *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pp. 894–905, VLDB Endowment, 2006.

[63] Hu, H., Xu, J., and Lee, D. L., "A generic framework for monitoring continuous spatial queries over moving objects," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, (New York, NY, USA), pp. 479–490, ACM, 2005.

[64] Huang, Y., Feamster, N., Lakhina, A., and Xu, J. J., "Diagnosing network disruptions with network-wide analysis," in *Proceedings of the 2007 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '07, (New York, NY, USA), pp. 61–72, ACM, 2007.

[65] Huang, Z., Du, W., and Chen, B., "Deriving private information from randomized data," in *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, (New York, NY, USA), pp. 37–48, ACM, 2005.

[66] Hyytiä, E. and Virtamo, J., "Random waypoint mobility model in cellular networks," *Wirel. Netw.*, vol. 13, pp. 177–188, April 2007.

[67] Imrich, W. and Klavžar, S., *Product Graphs: Structure and Recognition*. Wiley-Interscience, 2000.

[68] Internet Engineering Task Force, "OSPF version 2," *http://www.ietf.org/rfc*.

[69] Kamat, P., Xu, W., Trappe, W., and Zhang, Y., "Temporal privacy in wireless sensor networks," in *Proceedings of the 27th International Conference on Distributed Computing Systems*, ICDCS '07, (Washington, DC, USA), pp. 23–, IEEE Computer Society, 2007.

[70] Kamat, P., Zhang, Y., Trappe, W., and Ozturk, C., "Enhancing source-location privacy in sensor network routing," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ICDCS '05, (Washington, DC, USA), pp. 599–608, IEEE Computer Society, 2005.

[71] Kantarcioğlu, M., Jin, J., and Clifton, C., "When do data mining results violate privacy?," in *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 599–604, ACM, 2004.

[72] Karger, P. and Frankel, Y., "Security and privacy threats to its," in *World Congress on Intelligent Transport Systems*, 1995.

[73] Kargupta, H., Datta, S., Wang, Q., and Sivakumar, K., "On the privacy preserving properties of random data perturbation techniques," in *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, (Washington, DC, USA), p. 99, IEEE Computer Society, 2003.

[74] Katzela, I. and Schwartz, M., "Schemes for fault identification in communication networks," *IEEE/ACM Trans. Netw.*, vol. 3, pp. 753–764, December 1995.

[75] Kempe, D., Kleinberg, J., and Tardos, E., "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, (New York, NY, USA), pp. 137–146, ACM, 2003.

[76] Kido, H., Yanagisawa, Y., and Satoh, T., "An anonymous communication technique using dummies for location-based services," *International Conference on Pervasive Services*, pp. 88–97, 2005.

[77] Kifer, D. and Gehrke, J., "Injecting utility into anonymized datasets," in *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, SIGMOD '06, (New York, NY, USA), pp. 217–228, ACM, 2006.

[78] Kolahdouzan, M. and Shahabi, C., "Voronoi-based k nearest neighbor search for spatial network databases," in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, VLDB '04, pp. 840–851, VLDB Endowment, 2004.

[79] Kondor, R. I. and Lafferty, J. D., "Diffusion kernels on graphs and other discrete input spaces," in *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, (San Francisco, CA, USA), pp. 315–322, Morgan Kaufmann Publishers Inc., 2002.

[80] Kong, J. and Hong, X., "Anodr: anonymous on demand routing with untraceable routes for mobile ad-hoc networks," in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, MobiHoc '03, (New York, NY, USA), pp. 291–302, ACM, 2003.

[81] Ku, W.-S., Zimmermann, R., Peng, W.-C., and Shroff, S., "Privacy protected query processing on spatial networks," in *Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, (Washington, DC, USA), pp. 215–220, IEEE Computer Society, 2007.

[82] Lakhina, A., Crovella, M., and Diot, C., "Mining anomalies using traffic feature distributions," in *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '05, (New York, NY, USA), pp. 217–228, ACM, 2005.

[83] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R., "Incognito: efficient full-domain k-anonymity," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, SIGMOD '05, (New York, NY, USA), pp. 49–60, ACM, 2005.

[84] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R., "Mondrian multidimensional k-anonymity," in *Proceedings of the 22nd International Conference on Data Engineering*, ICDE '06, (Washington, DC, USA), pp. 25–, IEEE Computer Society, 2006.

[85] LeFevre, K., DeWitt, D. J., and Ramakrishnan, R., "Workload-aware anonymization," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pp. 277–286, 2006.

[86] Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., and Ghahramani, Z., "Kronecker graphs: An approach to modeling networks," *J. Mach. Learn. Res.*, vol. 11, pp. 985–1042, March 2010.

[87] Leskovec, J., Huttenlocher, D., and Kleinberg, J., "Signed networks in social media," in *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, (New York, NY, USA), pp. 1361–1370, ACM, 2010.

[88] Lewis, L. M., "A case-based reasoning approach to the resolution of faults in communication networks," in *Proceedings of the IFIP TC6/WG6.6 Third International Symposium on Integrated Network Management*, (Amsterdam, The Netherlands), pp. 671–682, North-Holland Publishing Co., 1993.

[89] Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., and Teng, S., "On trip planning queries in spatial databases," in *SSTD*, 2005.

[90] Li, F., Sun, J., Papadimitriou, S., Mihaila, G. A., and Stanoi, I., "Hiding in the crowd: Privacy preservation on evolving streams through correlation tracking," in *ICDE'07: Proceedings of the 23th IEEE International Conference on Data Mining*, (Washington, DC, USA), pp. 686–695, IEEE Computer Society, 2007.

[91] Li, J., Tao, Y., and Xiao, X., "Preservation of proximity privacy in publishing numerical sensitive data," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pp. 473–486, 2008.

[92] Li, N., Li, T., and Venkatasubramanian, S., "t-closeness: privacy beyond k-anonymity and l-diversity," in *Proceedings of the 23nd International Conference on Data Engineering*, ICDE '07, (Washington, DC, USA), IEEE Computer Society, 2007.

[93] Lin, C.-Y., Cao, N., Liu, S. X., Papadimitriou, S., Sun, J., and Yan, X., "Smallblue: Social network analysis for expertise search and collective intelligence," in *Proceedings of the 2009 IEEE International Conference on Data Engineering*, (Washington, DC, USA), pp. 1483–1486, IEEE Computer Society, 2009.

[94] Lindell, Y. and Pinkas, B., "Privacy preserving data mining," in *CRYPTO '00: Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, (London, UK), pp. 36–54, Springer-Verlag, 2000.

[95] Liu, L., Tang, J., Han, J., Jiang, M., and Yang, S., "Mining topic-level influence in heterogeneous networks," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, (New York, NY, USA), pp. 199–208, ACM, 2010.

[96] Lovász, L., "On decompositions of graphs," *Studia Sci. Math. Hungar.*, vol. 1, pp. 237–238, 1966.

[97] Lukasiewicz, T., "Probabilistic logic programming with conditional constraints," *ACM Trans. Comput. Logic*, vol. 2, no. 3, pp. 289–339, 2001.

[98] Machanavajjhala, A., Kifer, D., Gehrke, J., and Venkitasubramaniam, M., "L-diversity: Privacy beyond k-anonymity," *ACM Trans. Knowl. Discov. Data*, vol. 1, March 2007.

[99] Martin, D., Kifer, D., Machanavajjhala, A., Gehrke, J., and Halpern., J., "Worst-case background knowledge in privacy," in *Proceedings of the 23nd International Conference on Data Engineering*, ICDE '07, (Washington, DC, USA), IEEE Computer Society, 2007.

[100] Meng, X., Jiang, G., Zhang, H., Chen, H., and Yoshihira, K., "Automatic profiling of network event sequences: Algorithm and applications," in *2008 IEEE INFOCOM - The 27th Conference on Computer Communications*, pp. 266–270, IEEE, 2008.

[101] Meyerson, A. and Williams, R., "On the complexity of optimal k-anonymity," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '04, (New York, NY, USA), pp. 223–228, ACM, 2004.

[102] Mokbel, M. F., Chow, C.-Y., and Aref, W. G., "The new casper: query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pp. 763–774, VLDB Endowment, 2006.

[103] Mouratidis, K., Yiu, M. L., Papadias, D., and Mamoulis, N., "Continuous nearest neighbor monitoring in road networks," in *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pp. 43–54, VLDB Endowment, 2006.

[104] Nergiz, M. E., Atzori, M., and Clifton, C., "Hiding the presence of individuals from shared databases," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, (New York, NY, USA), pp. 665–676, ACM, 2007.

[105] Nygate, Y. A., "Event correlation using rule and object based techniques," in *Proceedings of the fourth international symposium on Integrated network management IV*, (London, UK, UK), pp. 278–289, Chapman & Hall, Ltd., 1995.

[106] O'Connor, L., "The inclusion-exclusion principle and its applications to cryptography," *Cryptologia*, vol. 17, no. 1, pp. 63–79, 1993.

[107] Palla, G., Barabasi, A.-L., and Vicsek, T., "Quantifying social group evolution," *Nature*, vol. 446, no. 7136, pp. 664–667, 2007.

[108] Papadias, D., Zhang, J., Mamoulis, N., and Tao, Y., "Query processing in spatial network databases," in *Proceedings of the 29th international conference on Very large data bases - Volume 29*, VLDB '2003, pp. 802–813, VLDB Endowment, 2003.

[109] Park, H. and Shim, K., "Approximate algorithms for k-anonymity," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, (New York, NY, USA), pp. 67–78, ACM, 2007.

[110] Pearl, J., *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988.

[111] Salfner, F., "Event-based failure prediction: an extended hidden markov model approach," *Department of Computer Science, Humboldt-Universität zu Berlin, Germany*, 2008.

[112] Shepitsen, A., Gemmell, J., Mobasher, B., and Burke, R., "Personalized recommendation in social tagging systems using hierarchical clustering," in *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, (New York, NY, USA), pp. 259–266, ACM, 2008.

[113] Shoshani, A., "Statistical databases: Characteristics, problems, and some solutions," in *VLDB '82: Proceedings of the 8th International Conference on Very Large Data Bases*, (San Francisco, CA, USA), pp. 208–222, Morgan Kaufmann Publishers Inc., 1982.

[114] Singla, P. and Richardson, M., "Yes, there is a correlation: - from social networks to personal behavior on the web," in *Proceeding of the 17th international conference on World Wide Web*, WWW '08, (New York, NY, USA), pp. 655–664, ACM, 2008.

[115] Song, C., Qu, Z., Blumm, N., and Barabasi, A.-L., "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[116] Steinder, M. and Sethi, A. S., "A survey of fault localization techniques in computer networks.," *Sci. Comput. Program.*, vol. 53, no. 2, pp. 165–194, 2004.

[117] Sun, J., Tao, D., and Faloutsos, C., "Beyond streams and graphs: dynamic tensor analysis," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, (New York, NY, USA), pp. 374–383, ACM, 2006.

[118] Sun, Y., Yu, Y., and Han, J., "Ranking-based clustering of heterogeneous information networks with star network schema," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, (New York, NY, USA), pp. 797–806, ACM, 2009.

[119] Sweeney, L., "k-anonymity: a model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, October 2002.

[120] Tang, J., Sun, J., Wang, C., and Yang, Z., "Social influence analysis in large-scale networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, (New York, NY, USA), pp. 807–816, ACM, 2009.

[121] Traub, J. F., Yemini, Y., and Woźniakowski, H., "The statistical security of a statistical database," *ACM Trans. Database Syst.*, vol. 9, no. 4, pp. 672–679, 1984.

[122] USAToday, "Authorities: Gps systems used to stalk woman," *http://www.usatoday.com/tech/news/2002-12-30-gps-stalker_x.htm*, 2002.

[123] Vaidya, J. and Clifton, C., "Privacy preserving association rule mining in vertically partitioned data," in *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, (New York, NY, USA), pp. 639–644, ACM, 2002.

[124] Vavasis, S. A., "Quadratic programming is in np," *Inf. Process. Lett.*, vol. 36, no. 2, pp. 73–77, 1990.

[125] Wan, G. and Lin, E., "A dynamic paging scheme for wireless communication systems," in *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, MobiCom '97, (New York, NY, USA), pp. 195–203, ACM, 1997.

[126] Wang, K., Fung, B. C. M., and Yu, P. S., "Handicapping attacker's confidence: an alternative to k-anonymization," *Knowl. Inf. Syst.*, vol. 11, no. 3, pp. 345–368, 2007.

[127] Wang, K., Yu, P. S., and Chakraborty, S., "Bottom-up generalization: A data mining solution to privacy protection," in *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, (Washington, DC, USA), pp. 249–256, IEEE Computer Society, 2004.

[128] Wang, T. and Liu, L., "Butterfly: Protecting output privacy in stream mining," in *ICDE '08: Proceedings of the 2008 IEEE International Conference on Data Engineering*, (Washington, DC, USA), IEEE Computer Society, 2008.

[129] Wang, T. and Liu, L., "Privacy-aware mobile services over road networks," *Proc. VLDB Endow.*, vol. 2, pp. 1042–1053, August 2009.

[130] Wang, T. and Liu, L., "Xcolor: Protecting general proximity privacy," in *ICDE '10: Proceedings of the 2010 IEEE International Conference on Data Engineering*, (Washington, DC, USA), IEEE Computer Society, 2010.

[131] Wang, T. and Liu, L., "Output privacy in data mining," *ACM Trans. Database Syst.*, vol. 36, pp. 1:1–1:34, March 2011.

[132] Wang, T., Meng, S., Bamba, B., Liu, L., and Pu, C., "A general proximity privacy principle," in *ICDE '09: Proceedings of the 2009 IEEE International Conference on Data Engineering*, (Washington, DC, USA), pp. 1279–1282, IEEE Computer Society, 2009.

[133] Wang, T., Srivatsa, M., Agrawal, D., and Liu, L., "Learning, indexing, and diagnosing network faults," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, (New York, NY, USA), pp. 857–866, ACM, 2009.

[134] Wang, T., Srivatsa, M., Agrawal, D., and Liu, L., "Spatio-temporal patterns in network events," in *Proceedings of the 6th International COnference*, Co-NEXT '10, (New York, NY, USA), pp. 3:1–3:12, ACM, 2010.

[135] Wang, T., Srivatsa, M., Agrawal, D., and Liu, L., "Modeling data flow in socio-information networks: A risk estimation approach," in *Proceeding of the 16th ACM symposium on Access control models and technologies*, SACMAT '11, (New York, NY, USA), ACM, 2011.

[136] Want, R., Hopper, A., Falcão, V., and Gibbons, J., "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, pp. 91–102, January 1992.

[137] Winkleby, M., Jatulis, D., Frank, E., and Fortmann, S., "Socioeconomic status and health: how education, income, and occupation contributes to risk factors for cardiovascular disease," *Am. J. Public Health*, vol. 82, no. 6, 1992.

[138] Woeginger, G. J. and Yu, Z., "On the equal-subset-sum problem," *Inf. Process. Lett.*, vol. 42, pp. 299–302, July 1992.

[139] Wong, R. C.-W., Li, J., Fu, A. W.-C., and Wang, K., "(alpha, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '06, pp. 754–759, 2006.

[140] Wu, P., Bhatnagar, R., Epshtein, L., Bhandaru, M., and Zhongwen, S., "Alarm correlation engine," in *Proceedings of the 1998 IEEE Network Operations and Management Symposium*, vol. 3, pp. 733–742, 1998.

[141] Xiang, R., Neville, J., and Rogati, M., "Modeling relationship strength in online social networks," in *Proceedings of the 19th international conference on World wide web*, WWW '10, pp. 981–990, ACM, 2010.

[142] Xiao, X. and Tao, Y., "Anatomy: simple and effective privacy preservation," in *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pp. 139–150, VLDB Endowment, 2006.

[143] Xiao, X. and Tao, Y., "M-invariance: towards privacy preserving republication of dynamic datasets," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pp. 689–700, 2007.

[144] Yang, H., King, I., and Lyu, M. R., "Diffusionrank: a possible penicillin for web spamming," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, (New York, NY, USA), pp. 431–438, ACM, 2007.

[145] Yemini, S. A., Kliger, S., Mozes, E., Yemini, Y., and Ohsie, D., "High speed and robust event correlation," *Communications Magazine, IEEE*, vol. 34, no. 5, pp. 82–90, 1996.

[146] Yuan, C., Lao, N., Wen, J.-R., Li, J., Zhang, Z., Wang, Y.-M., and Ma, W.-Y., "Automated known problem diagnosis with event traces," *SIGOPS Oper. Syst. Rev.*, vol. 40, pp. 375–388, April 2006.

[147] Zang, H. and Bolot, J. C., "Mining call and mobility data to improve paging efficiency in cellular networks," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, MobiCom '07, (New York, NY, USA), pp. 123–134, ACM, 2007.

[148] Zhang, J., Rexford, J., and Feigenbaum, J., "Learning-based anomaly detection in bgp updates," in *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, MineNet '05, pp. 219–220, ACM, 2005.

[149] Zhang, Q., Koudas, N., Srivastava, D., and Yu, T., "Aggregate query answering on anonymized tables," *Data Engineering, International Conference on*, pp. 116–125, 2007.

[150] Zhang, T., Ramakrishnan, R., and Livny, M., "Birch: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, SIGMOD '96, (New York, NY, USA), pp. 103–114, ACM, 1996.

[151] Zheleva, E., Sharara, H., and Getoor, L., "Co-evolution of social and affiliation networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, (New York, NY, USA), pp. 1007–1016, ACM, 2009.