2010

An Ising-based approach for tracking illegal P2P content distributors

Lu Dai
Iowa State University

Recommended Citation

Dai, Lu, "An Ising-based approach for tracking illegal P2P content distributors" (2010). *Graduate Theses and Dissertations*. 11810.
https://lib.dr.iastate.edu/etd/11810

An Ising-based approach for tracking illegal P2P content distributors

by

Lu Dai

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Lei Ying, Major Professor
Yong Guan
Aditya Ramamoorthy

Iowa State University

Ames, Iowa

2010

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This thesis focuses on the problem of tracking illegal P2P content distributors. By viewing the collection of files of a peer as a relatively precise reflection of its owner, we use the Ising model which originates from statistical physics to mathematically model the behavior of P2P networks and identify the relationships of peers. Based on it, we develop an effective approach to track the behavioral-based structures of P2P networks and use it as a guidance to narrow down the search scope for illegal P2P content distributors. The sum-product algorithm and mean field algorithm which are based on the Ising model are then used to efficiently compute the marginal distribution of peers that are holding or held a particular file of known contraband. Experimental results have shown that this behavioral-based approach significantly outperforms several tracking algorithms that ignore the relationships of peers in P2P networks.

CHAPTER 1. INTRODUCTION

Peer-to-peer (P2P) network technology has enabled many interesting and popular content sharing and streaming multimedia applications and services, such as KaZaA, BitTorrent, Limewire, and Skype.com. However, in recent years, it has also become the standard instrumentality for the sharing and distribution of child pornography [Koontz, L. D. (2003)] and other digital contraband. For example, the increasing availability of child pornography and the ease of access to them have put millions of juvenile users of peer-to-peer networks at significant risk of inadvertent exposure to pornography, including child pornography. In addition, copyright-protected materials (such as pre-release movies, music, software products, etc.) are being distributed on P2P illegally. Therefore, it is critical to build the effective deterrence and forensics capability to successfully track, identify and prosecute illegal P2P content distributors within the requirements of the law.

We define an illegal P2P content distributor to be a P2P node that is holding or held a particular file of known contraband. There are two main P2P network architectures for sharing content on the Internet: Gnutella and BitTorrent. In Gnutella, all peers operate as client and server, both originating search queries for files and distributing contents. When a peer receives a query for contents that match its local deposits, a reply is returned to the peer that originated it. The originator of the query then downloads the desired contents from one of the received responses. Otherwise, the peer which received the query forwards the query to its neighboring peers to propagate further until a specified maximum number of hops from the originator is reached. Most Gnutella peers share files (less likely a portion of the file) among them. In Bittorrent (BT), files are split into small segments. Usually, BT uses tracker (like a directory service) to keep track of which peers are sharing which portions (subset of segments) of files. After a peer downloads a segment, it becomes a server for it. These segments can be downloaded in parallel from multiple peers. If a peer has a full-copy of the file and allows others to download it, it becomes a seeder. A peer initially downloads a file from the first seeder, and can then become seeders themselves. So all peers that hold the given contraband are distributors of the illegal content in P2P networks. Due to the highly-dynamic and stateless nature of P2P systems, especially in Gnutella which lacks a central tracker, it is very challenging to track and identify illegal P2P content distributors who can take ways to evade the detection and eliminate evidence relevant to them [Ieong, R. (2009)] [Liberatore, M. (2010)]. For example, the illegal content distributor can intentionally stop the uploading service of the file, once it learns that a sufficient number of peers have become the seeders of that file.

In this thesis, we focus on the problem of tracking the illegal P2P content distributors. Though it is hard to solve, we found that most users and activities in P2P systems are still susceptible to tracking in different ways. We generally believe/view a peer and the collection of files on it a relatively precise reflection of its owner. Most likely, we can learn his/her interests and background and how they evolve/develop over time from his/her local stores. Also, the interactions between peers may indicate certain level or set of common interests and preferences of their owners. In the case that two owners of the peers share some common interests in classical music, if one has a particular piece of music, it is very (more) likely that the other has the same piece in its local store. We take advantage of these observations and knowledge about the peers' preference and shared common interests and use the Ising model [Ising, E. (1925)] (from statistical physics) to mathematically model the behavior of P2P networks and identify the relationships of peers. Based on it, we develop an effective approach to track the behavioral-based structures of P2P networks and use it as a guidance to narrow down the search scope for illegal P2P content distributors. The sum-product and mean field algorithm based on the Ising model are then used to efficiently compute the marginal distribution that a peer is holding or held a particular file of known contraband. Thereby, we can use it to track illegal contents in P2P networks. Experimental results have shown that this Ising-based approach can significantly outperform several simple tracking algorithms that ignore the relationships of peers.

This thesis contains the following contributions. First, we introduce the Ising model to mathematically model the relationships of peers in a P2P file-sharing networks. Each peer under the Ising model has two states $\{-1, +1\}$, where $+1$ indicates the peer has the illegal content and -1 indicates otherwise. We derive the probability distribution of peer states conditioned on a subset of observed peers, and the corresponding sum-product and mean field algorithms that can efficiently compute the marginal distributions of peer states in large P2P networks. Second, we analyze a real P2P file-sharing network based on the data collected in [Fast, A. (2005)]. We show that even a highly-dynamic P2P file-sharing network exhibits scale-free and small-world properties, which are properties that have been observed in many social networks [Barabasi, A. (1999)] [Jackson, M. (2008)] [Easley, D. (2010)]. These observations confirm that interactions between peers indicate certain level or set of common interests and preferences of peers. This relationships of peers can be exploited to facilitate tracking illegal distributors. Third, we experiment the sum-product and mean field tracking algorithms on the P2P data collected in [Fast, A. (2005)], and compare the tracking accuracy (to be defined in Chapter 4) with some algorithms that ignore the relationships in P2P networks. The experimental results show that the Ising-based approach achieves much higher tracking accuracy.

This thesis is organized as follows: Chapter 2 gives the basic model for P2P file-sharing networks; Chapter 3 introduces the data set used in experiment; Chapter 4 first introduces the belief propagation algorithm as well as evaluation metrics, then gives the result of experiment; Chapter 5 gives the conclusion and discussion.

CHAPTER 2. BASIC MODEL

2.1 Ising Model in Statistical Physics

Ising model was proposed as a mathematical model of ferromagnetism in statistical physics and named by Ernst Ising [Ising, E. (1925)]. This model is used to explain the phenomenon of "spontaneous magnetization", the appearance of an ordered spin state (magnetization) at zero applied magnetic field in a ferromagnetic or ferrimagnetic material when the temperature is below a critical point called the "critical temperature". The model consists of spins that can be in one of two states, that is either up or down which are denoted by $\{+1, -1\}$. The spins are arranged in a lattice or graph, and each spin interacts only with its nearest neighbors. In the model of ferromagnetism which Ising model originates from, the lattice sites are occupied by atoms of a magnetic material. Each atom has a magnetic moment which is allowed to point either "up" or "down". Variable σ_i is used to denote which state the ith lattice site is in. σ_i is 1 if the ith lattice site is in the state of "up", and -1 if it is in the state of "down". For each configuration $\sigma = (\sigma_1, \ldots, \sigma_N)$, the energy of the lattice is

$$H = H(\sigma) = -\sum_{\langle i,j\rangle} E\sigma_i\sigma_j - \sum_i J\sigma_i,$$

where $< i, j >$ denotes a pair of neighboring spins, constant E >0 represents the magnetic interaction that keeps neighboring spins aligned along the same direction, and constant J $>$ 0 represents the effect of external magnetic field of intensity. It is easy to see that when all the spins are in the same direction as the external field, the energy is minimized. Gibbs measure can be constructed for the Ising model such that for a given configuration $\sigma = (\sigma_1, \ldots, \sigma_N)$, that is

$$p(\sigma) = \frac{e^{-\beta H(\sigma)}}{Z}$$

where

$$H\left(\sigma\right) = -\sum_{\langle i,j\rangle} E\sigma_i\sigma_j - \sum_i J\sigma_i,$$

$$\beta = \frac{1}{KT},$$

and K is the Boltzmann's constant and T is temperature. The normalizing constant Z, defined by $Z = \sum_\sigma e^{-\beta H(\sigma)}$ is called the partition function.

There are two reasons for Gibbs measure's being used for the Ising model. First, assume the expected energy is $E\left(H\right) = a$, then Gibbs measure defined above is the probability measure that maximizes entropy among all probability measures which result in the same expected energy a. Second, Gibbs measure has the Markov random field (MRF) property, that is the state of a spin is independent of other spins given the states of its neighbors. The Ising model successfully explains the phase transition in 2-dimensional case [Ising, E. (1925)].

2.2 Ising Model from Graphical Model

Although Ising model originates from statistical physics, it can also be viewed as a graphical model. A graphical model is a probabilistic model in which nodes represent random variables and link represent the dependence of random variables. The formalism of probabilistic graphical models captures the complex dependencies among random variables, and can be used to establish large-scale multivariate statistical models. Graphical models have been used in statistical, computational and mathematical fields, including bioinformatics, communication theory, statistical physics, combinatorial optimization, signal and image processing, information retrieval and statistical machine learning [Wainwright, M. J. (2003)].

We next form the Ising model using the graphical model. A graph is denoted by $G = (V, E)$, where V is the set of vertices and E is the set of edges. Each edge connects a pair of vertices $(s, t) \in E$ and can be either undirected or directed. In the case of directed graph, the edge (s, t) and edge (t, s) are different. In the remainder of this thesis, we limit our discussion to undirected graph so there is only one link corresponding to every edge. Graphical models associate each vertice $s \in V$ with a random variable X_s, taking values in state space $\mathcal{X}_s$. The state space $\mathcal{X}_s$ may be either continuous or discrete. In this thesis, we only consider finite and discrete state space. We use lower-case letters ($x_s \in \mathcal{X}_s$) to denote elements of $\mathcal{X}_s$, so that the notation $\{X_s = x_s\}$ means that the random variable X_s takes the value $x_s \in \mathcal{X}_s$.

With these notations, an undirected graphical model, also known as Markov random field is a collection of distributions that factorize as

$$p\left(x_1, x_2, \ldots, x_m\right) = \frac{1}{Z}\prod_{C \in \mathcal{C}} \psi_C\left(x_C\right)$$

where Z is a normalizing constant to ensure that p is a valid probability distribution. Potential function $\psi_C\left(x_C\right)$ is defined on the set of clique $\mathcal{C}$, which is often taken to be the set of all maximal cliques of the graph.

For a tree structure, the distribution can be written as

$$p\left(x_1, x_2, \ldots, x_m\right) = \frac{1}{Z}\prod_{s \in V} \psi_s\left(x_s\right) \prod_{(s,t) \in E(T)} \psi_{st}\left(x_s, x_t\right)$$

The potential functions are defined over single nodes and pairwise edges.

If we limit the potential functions to the exponential family [Efron, B. (1978)] [Wainwright, M. J. (2003)], then the probability distribution can be written as

$$p\left(x_1, x_2, \ldots, x_m\right) = \frac{1}{Z}\prod_{s \in V} \exp\left\{\theta_s x_s\right\} \prod_{(s,t) \in E(T)} \exp\left\{\theta_{st} x_s x_t\right\}$$

which could be further written in the following standard form

$$p\left(x_1, x_2, \ldots, x_m\right) = \frac{1}{Z}\exp\left\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E(T)} \theta_{st} x_s x_t\right\} = \exp\left\{\sum_{s \in V} \theta_s x_s + \sum_{(s,t) \in E(T)} \theta_{st} x_s x_t - A(\theta)\right\}$$

or in a general form

$$p_\theta\left(x_1, x_2, \ldots, x_m\right) = \exp\left\{\left\langle \theta, \phi\left(x\right)\right\rangle - A\left(\theta\right)\right\}.$$

Here $\theta = \{\theta_s, s \in V\} \cup \{\theta_{st}, (s,t) \in E(T)\}$, and $\phi(x) = \{x_s, s \in V\} \cup \{x_s x_t, (s,t) \in E(T)\}$. Here we assume that each single variable x_s takes values in the space $\{-1, 1\}$ and the space for X is $\{-1, 1\}^m$. The quantity A is the log partition function or cumulant function, and is

$$A\left(\theta\right) = \log \sum_{x \in \mathcal{X}^m} \exp\left\langle \theta, \phi\left(x\right)\right\rangle$$

It is easy to verify that the definition of $A(\theta)$ ensures p_θ is properly normalized, that is

$$\sum_{x \in \mathcal{X}^m} \exp\{\langle \theta, \phi(x) \rangle - A(\theta)\} = 1$$

If x_s is allowed to take values in the discrete space of $\{1, 2, \ldots, r-1\}$ for some integer $r > 2$, it leads to the Potts model, which is the generalization of Ising model [Wainwright, M. J. (2003)]. We only consider discrete space of $\{+1, -1\}$.

2.3 Ising Model on P2P Network

In this thesis, we consider a peer-to-peer (P2P) file-sharing network represented by a graph $G = (V, L)$ where V is the set of peers, and L is the set of links. Denote by $\theta \in R_+^{|V| \times |V|}$ the link weight matrix, where $\theta_{ij} \geq 0$ is the weight of link (i, j). The value of θ_{ij} represents the strength of the connection between peer i and peer j. Larger θ_{ij} represents a stronger relationship between peer i and peer j. The values of θ are defined based on network statistics we collect.

In P2P file-sharing networks such as Gnutella and BitTorrent, there are two major interactions between peers: query and file-transfer. While query processes are in general random, a file-transfer indicates that the two peers (sender and receiver) share a common interest or preference on the file. We therefore believe the number of file transfers between two peers has a strong (positive) correlation with the level of common interests and preferences of the owners of the two peers. For the data set to be studied in Chapter 3, we will define the link weight θ_{ij} to be (linearly) proportional to the number of file transfers between peer i and peer j. The experimental results confirm that exploiting the relationships of peers (defined based on the number of file transfers between peers, for example) significantly improves the tracking accuracy.

The focus of this paper is to track (illegal) content distributors in P2P networks. Specifically, assuming that an illegal content has been found at some peer (say peer j), we are interested in finding out which other peers are also likely to have that file. To mathematically model this problem, we define a binary random variable X_i for each peer i, which represents whether peer i has the file we are interested (e.g., the illegal content) in. X_i takes values in $\{-1,+1\}$, $X_i = 1$ indicates the peer has the file, and otherwise, $X_i = -1$. The state of P2P network is therefore represented by vector $X = \left(X_1, X_2, \ldots, X_{|V|}\right)$. Each realization of X is called a configuration of the P2P network. Based on the definitions above, mathematically, the tracking problem is to find the subset of peers that have large values of

$$\Pr\left(X_i = 1 | X_j = 1\right), \tag{2.1}$$

which corresponds to the subset of peers that most likely are holding or held the illegal content.

In this thesis, we adopt a general Ising model to mathematically model P2P file-sharing networks. We allow heterogeneous link weights in the general Ising model to reflect heterogeneous connection strengths among peers. We further assume $J = 0$, i.e., no external influence is present in the network (see Remark 1), and define the probability that the P2P network is in configuration X is

$$\Pr\left(X\right) = \frac{\exp\left(\sum_{(i,j)} \beta\theta_{ij} X_i X_j\right)}{Z}, \tag{2.2}$$

where β is a scaling factor and

$$Z = \sum_{X} \exp\left(\sum_{(i,j)} \beta\theta_{ij} X_i X_j\right).$$

Remark 1: We set $J = 0$ by assuming a P2P network is a closed system, i.e., the state of a peer only depends on the states of other peers in the network. While in practical systems, the behavior of the owner of a peer is likely to be affected by many external factors, we found it is difficult to quantitatively measure and model these factors in the model. Therefore we ignore the external influence in our model. Our experimental results in Chapter 4 however show that the algorithm based on this closed-network assumption is still effective in facilitating tracking.

Remark 2: The Ising model defines a Markov random field, so the network state is determined by local interactions. We believe it is proper for modeling P2P networks because the (global) distribution of a file among peers in P2P networks is determined by (local) file transfers between peers. More importantly, this Markov random field model allows us to adopt belief-propagation methods, such as the sum-product algorithm, to efficiently compute the marginal distribution (2.1) for large P2P networks. Due to the large scale of P2P networks, the computation complexity is a critical consideration in the design of tracking algorithms for P2P networks.

Under the Ising model (2.2), we now can compute (2.1). Then we can rank the peers based on (2.1), and track the peers with large values of (2.1). If the Ising model is accurate, i.e., the peers that have large values of (2.1) under the Ising model are indeed the peers that are likely to hold the content in reality, then the ranking can help us efficiently find illegal content distributors.

In the next section, we discuss the sum-product and mean field algorithm, algorithms that can efficiently approximate the marginal distributions under the Ising model (2.2). While they may not result in the exact marginal distributions, they have been known to give very accurate approximations in many applications [Wainwright, M. (2008)]. We will evaluate the accuracies of both the Ising model and the sum-product tracking algorithm as well as mean field tracking algorithm using the real P2P network data collected in Fast, A. (2005) in Chapter 4.

We note that the Ising model (2.2) we use only captures the relationships of peers. It however does not model individuals' preferences or interests. In other words, the interest of a peer on the target file is neutral if other peers' preferences are unknown. Mathematically, as shown in Proposition 1, we have $\Pr(X_i = +1) = \Pr(X_i = -1) = 0.5$.

Proposition 1. *Under the Ising model (2.2),*

$$\Pr(X_i = +1) = \Pr(X_i = -1) = 0.5.$$

Proof. For each configuration $\mathbf{X}$, we denote by $\tilde{\mathbf{X}}$ the configuration such that $\tilde{X}_i = -X_i$ for all i. Due to the symmetry of (2.2), we first have

$$\Pr(\mathbf{X}) = \Pr(\tilde{\mathbf{X}}).$$

Therefore,

$$\begin{aligned}
\Pr(X_i = +1) &= \sum_{\mathbf{X}:X_i=+1} \Pr(\mathbf{X}) \\
&= \sum_{\mathbf{X}:X_i=+1} \Pr(\tilde{\mathbf{X}}) \\
&= \sum_{\mathbf{X}:X_i=-1} \Pr(\mathbf{X}) \\
&= \Pr(X_i = -1).
\end{aligned}$$

□

For tracking, we usually start from a small set of evidence that a known contraband is found in some peers' local stores. This is a reasonable assumption because we track illegal content distributors only after we find the illegal content in the network (i.e., find some peer holds or held the illegal content). Therefore in this paper, we focus on the case where a set of peers have been observed. That is, a given contraband has been found and identified in some peers' local stores.

Without loss of generality, we assume the states of peers 1 to peer k have been observed. The observed states are represented by a vector $\mathbf{Y} = (Y_1, Y_2, \ldots, Y_k)^T$.

Proposition 2. *Given a set of observed peers* $\mathbf{Y}$, *the probability distribution of* $\mathbf{X}$ *conditioned on* $\mathbf{Y}$ *is*

$$\Pr(\mathbf{X}|\mathbf{Y}) = \frac{\exp\left(\sum_{(i,i')} \beta\theta_{i,i'} X_i X_{i'} + \sum_{1\le j\le k} \theta_j X_j\right)}{Z}, \tag{2.3}$$

where $\theta_i = +\infty$ *if* $Y_i = 1$ *and* $\theta_i = -\infty$ *otherwise, and*

$$Z = \sum_{\mathbf{X}} \exp\left(\sum_{(i,i')} \beta\theta_{i,i'} X_i X_{i'} + \sum_{j\le k} \theta_j X_j\right).$$

Proof. For those observed peers, $\Pr(Y_i = y_i | X_i = x_i)$ is the probability of observing $Y_i = y_i$ given the true state of peer i is x_i. Since $x_i \in \{+1, -1\}$, we have

$$\begin{aligned}
& \Pr(Y_i = y_i | X_i = x_i) \\
= \; & \Pr(Y_i = y_i | X_i = 1)^{\frac{x_i}{2}+\frac{1}{2}} \Pr(Y_i = y_i | X_i = -1)^{-\frac{x_i}{2}+\frac{1}{2}}.
\end{aligned}$$

The posterior probability of $\mathbf{X}$ conditioned on $\mathbf{Y}$ can be written as

$$\Pr\left(\mathbf{X}|\mathbf{Y}\right) \propto \Pr\left(\mathbf{X}\right)\Pr\left(\mathbf{Y}|\mathbf{X}\right).$$

Since the probability distribution of Y_j is solely determined by X_j, we have

$$\begin{aligned}
&\Pr\left(\mathbf{Y}|\mathbf{X}\right) \\
=\ & \textstyle\prod_{j=1}^{k} \Pr\left(Y_j = y_j | Y_1 = y_1, \ldots, Y_{j-1} = y_{j-1}, \mathbf{X} = \mathbf{x}\right) \\
=\ & \textstyle\prod_{j=1}^{k} \Pr\left(Y_j = y_j | X_j = x_j\right),
\end{aligned}$$

which implies that

$$\begin{aligned}
& \Pr\left(\mathbf{X} = \mathbf{x}|\mathbf{Y}\right) \\
\propto\ & \frac{\exp\left\{\beta \sum_{(i,j)} \theta_{ij} x_i x_j\right\}}{Z} \prod_{j=1}^{k} \Pr\left(Y_j = y_j | X_j = x_j\right) \\
\propto\ & \exp\left\{\beta \sum_{(i,j)} \theta_{ij} x_i x_j\right\} \times \\
& \prod_{j=1}^{k} \Pr\left(Y_j = y_j | X_j = 1\right)^{\frac{x_j}{2}+\frac{1}{2}} \times \\
& \qquad \Pr\left(Y_j = y_j | X_j = -1\right)^{-\frac{x_j}{2}+\frac{1}{2}} \\
\propto\ & \exp\left\{\beta \sum_{(i,j)} \theta_{ij} x_i x_j + \right. \\
& \left. \sum_{j=1}^{k} \frac{1}{2} \ln \frac{\Pr\left(Y_j = y_j | X_j = 1\right)}{\Pr\left(Y_j = y_j | X_j = -1\right)} x_j \right\}.
\end{aligned}$$

To compute $\Pr\left(\mathbf{X}|\mathbf{Y}\right)$, we need to first compute

$$\frac{1}{2} \ln \frac{\Pr\left(Y_j = y_j | X_j = 1\right)}{\Pr\left(Y_j = y_j | X_j = -1\right)} x_j$$

for every observed peer j. We note that

$$\frac{1}{2} \ln \frac{\Pr\left(Y_j = y_j | X_j = 1\right)}{\Pr\left(Y_j = y_j | X_j = -1\right)}$$

could be viewed as the node weight θ_j for observed peer j.

Assume that there is no observation noise, i.e., $\Pr(Y_j = X_j) = 1$. Then if $y_j = 1$, we have

$$\frac{1}{2} \ln \frac{\Pr(Y_j = 1 | X_j = 1)}{\Pr(Y_j = 1 | X_j = -1)} = +\infty;$$

and otherwise,

$$\frac{1}{2} \ln \frac{\Pr(Y_j = -1 | X_j = 1)}{\Pr(Y_j = -1 | X_j = -1)} = -\infty.$$

In other words, $\theta_j = +\infty$ if $y_j = 1$ and $\theta_j = -\infty$ otherwise, which concludes the proposition.

□

We comment that an observed peer (say peer j) introduces a node influence θ_j in the network, which forces the state of peer j to be the same as the observed state. We further note that we cannot use ∞ in computation, so we replace ∞ with a large positive constant M in computation. Now given the expression of the conditional probability (2.3), to compute the marginal distributions of peer states, we can use the Markov chain Monte Carlo (MCMC) method. However, when the network size is large, the MCMC can take an extremely long time to converge. We next introduce the sum-product algorithm and mean field algorithm [Aji, S. M. (2000)], [Kschischang, F. (2001)], [Wainwright, M. (2008)] , the well-known efficient and accurate algorithms for computing marginal distributions in Markov random fields.

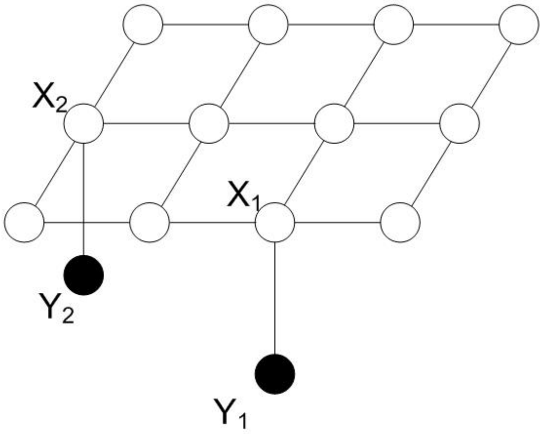

Figure 2.1 HMM model

Here is one example, assume there are 5 users and the first user V_1 is observed to have one particular file, the strength for correlation between these users are labeled on the Figure 2.2.

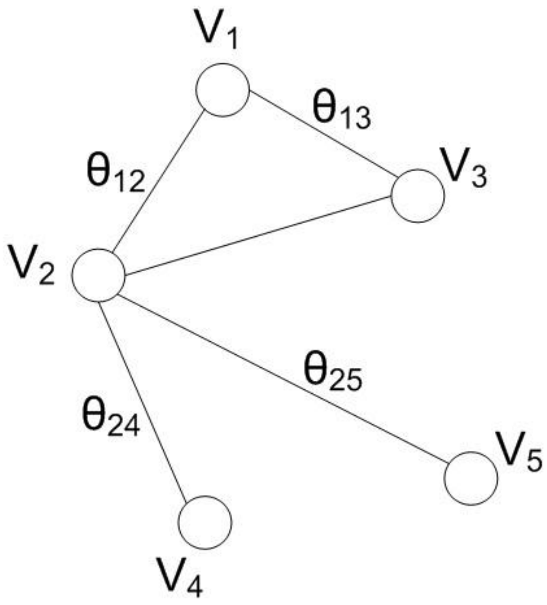

Figure 2.2 Example for the model

then the probability for the states of all the nodes is:

$$p\left(x_1, x_2, \ldots, x_5 | Y_1 = 1\right) = \exp\left\{\theta_1 x_1 + \sum_{\langle i,j\rangle} \beta\theta_{ij} x_i x_j - A\left(\theta\right)\right\}$$

where $\theta_1 = +\infty$.

We claim that the Ising model is very useful for search in P2P networks. And we will verify that in the following chapters.

CHAPTER 3. DATA SET

In this chapter, we verify the accuracy of the Ising model for P2P networks. In the previous chapter, we introduced the Ising model for P2P file-sharing networks. The Ising model provides a mathematical model to predict the state of a P2P network, e.g., which peers have the specific file. We next use the P2P data set from University of Massachusetts Amherst [Fast, A. (2005)] to validate the Ising model for P2P file-sharing networks. The data were collected from a campus network for peer-to-peer file sharing based on OpenNap server. The data set contains all the records of mp3 files which were shared by and transferred between users from February 28, 2003 to May 21, 2003. Users are all uniquely identified by an anonymous MD5 hash. After consolidation was performed, the network contains 291,295 MP3 files, and 6,464 users.

The raw data set contains a large number of records which can be categorized into *Object*, *Link* and *Attribute*. *Object* can be further categorized into *User*, *File*, *Transfer* and *Query*. The most frequent event involves two users, one file, one transfer and one query, i.e., user A makes a query Q to user B about file F, and user B makes a transfer T of file F to user A. The links represent relationships between these objects. A link from a user to a file represents the user possesses the file, the link from a user to a query represents the user has made that query. The Attributes record all detailed information of objects and links. The relation is shown in Figure 3.1.

The data set recorded 221,152 file transfers occurred among these users. Of course there are some files participated in multiple transfers. The query is a file request made by a peer to another peer, which is associated with one transfer and one file. The query information is not used in our experiment. Each transfer has two links connecting to users, the user who sent the file and the user who received the file.

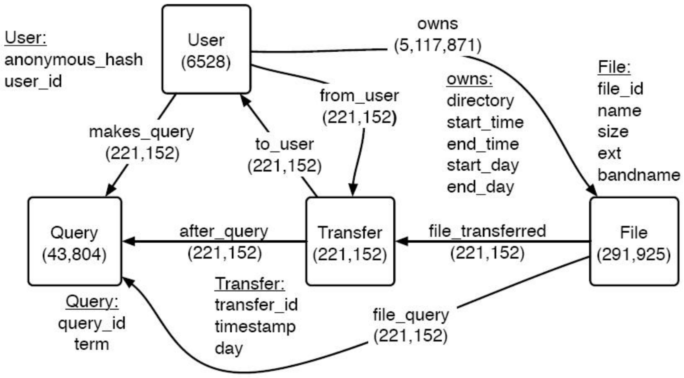

Figure 3.1 Data set

We constructed a graph $G = (V, E)$ based on the file transfer records as follows. A peer is represented by a vertex in the graph. If a file transfer happened between two peers, a link was added between the two vertices representing the two users. In our experiment, we set the link weight to be the number of file transfers between two peers multiplied by a scaling factor. Following this procedure, we obtained a graph with 6,464 nodes. The graph turns out to contain two disconnected components, one with 4,447 nodes, and the other with 2,017 nodes. We select the one with 4,447 nodes for our experiment, which includes 88,998 links totally.

We analyzed structure properties of this 4,447-node network, and found the network exhibits some interesting properties that have been observed in many social networks.

We first studied the degree distribution of the network. The log-log plot of the degree distribution is shown in Figure 3.2. We can see that the log-log plot is close to a linear line, which indicates that the network is similar to a scale-free network [Jackson, M. (2008)] [Easley, D. (2010)]. The maximum degree of the network is 1,731, and the average degree is around 40.

Table 3.1 Neighborhood size

	$i = 2$	$i = 3$	$i = 4$
Average size of i-hop neighborhoods	2,345	4,230	4,442

Further we investigated the average distance (in terms of number of hops) between two

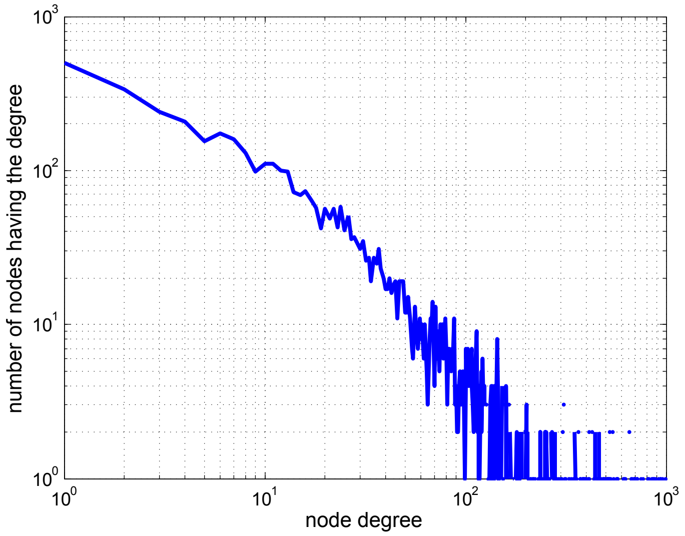

Figure 3.2 The degree distribution of the 4, 447-node network

peers in the network. The statistics is illustrated in Table 3.1. We can see that major pairs (95.12%) are within three hops or less. The network therefore exhibits the small-world property [Jackson, M. (2008)], [Easley, D. (2010)].

We comment that scale-free and small-world properties are key properties of many social networks where connections are driven by common interests. The existence of these properties in the P2P network indicates that the interaction of peers may be also driven by common interests. The correlation of the peers can then be exploited to facilitate the tracking process.

We continued to investigate the number of transfers which every file is involved. There are totally 221,152 transfers occurred among all the peers. Among them there are 40,951 files being transferred for only once, 8,887 files being transferred for twice, 4,567 files being transferred for three times, 2,911 files being transferred for four times and 1,994 files being transferred for more than five times. So we can see most of the files in the network have not been transferred during the 81 day period recorded in the data set, and may be transferred and obtained by users before this period. The distribution of transfer times of these files is shown in Table 3.2 below.

In the raw data, the set of files associated with a peer does not contain all the files that are sent or received by the peer. We therefore modified the set of files associated with a peer by *(i)*

Table 3.2 Transfer time of file

transfer time of file	**1**	**2**	**3**	**4**	**5**
number of files	40951	8887	4567	2911	1994

removing those files that are never transferred and *(ii)* adding those files that were transferred to/received from the peer. Further, as we find that a large fraction of files were transferred only once (40,951 files transferred for only once). Since we are interested in the files that are distributed to multiple peers in the network, we removed these files by viewing these files as observation noises.

Figure 3.3 illustrates the number of files each peer has after the modification. The average number of files a peer has is about 42, and the maximum number of files a peer has is $1,662$.

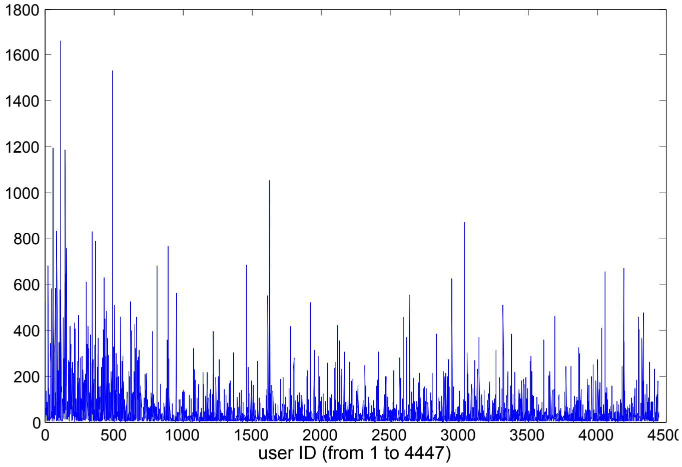

Figure 3.3 Number of files each peer holds

We define the link weight $\theta_{i,j}$ to be the number of file transfers, normalized by the maximal number of file transfers over all pairs of peers. We define the link weight in this way because the number of file transfers indicates the intensity of the interaction between two peers.

CHAPTER 4. BELIEF PROPAGATION ALGORITHM FOR PREDICTING THE STATE OF P2P NETWORKS

We have introduced the Ising model in Chapter 2 and the data set in Chapter 3, and in this chapter we will discuss the methods to verify the model using the real data set. We can compute the marginal probability distribution $p\left(X_i\right)$ for each peer X_i, and as noted in Chapter 2, when there is no observation in the network, the marginal probability distribution is [0.5 0.5]. So if there is no observed peer in the network, the topology of the P2P file-sharing network alone does not provide us any information of whether one specific peer has some file or not. Now assume we already know one peer (say peer s) has one specific file, the conditional probability given this observed user introduces $\theta_s x_s$ with $\theta_s = +\infty$ in the Gibbs measure. Though we can use some very large value M to represent this $+\infty$, to get an accurate marginal probability $p\left(X_i\right)$ from the joint probability is not an easy task. Of course the accurate value of marginal probability can be obtained using Markov chain Monte Carlo (MCMC). But we note that when the network size is large, the MCMC methods may take a very long time to converge. Variational method provides an approach to approximate the problem in an efficient way [Wainwright, M. J. (2003)]. In this chapter, we will first introduce the sum-product algorithm and mean field algorithm, both of which are representation of this variational method. After that, we will introduce the performance metrics and some simple tracking algorithms as bench marks. The result is given at the end of this chapter.

4.1 Sum-Product

For tree structure graphs, sum-product message passing algorithm is an approach to compute the marginal distribution of each node [Wainwright, M. J. (2003)]. Even for graphs with cycles, the algorithm has proved to be a good approximation [Wainwright, M. J. (2003)]. [Yedidia, J. S. (2005)] has related it to the Bethe Energy in statistical physics which deepens the understanding of this algorithm. The essence of message passing algorithms is iteratively computing the messages (beliefs). At each iteration, node s passes a message $M_{st}(x_t)$ to each of its neighbors $t \in N(s)$. The message $M_{st}(x_t)$ is a vector of length $|\mathcal{X}_t|$, with each element being associated with a possible state for $x_t \in \mathcal{X}_t$. So for every link (s,t), there are two messages M_{st} and M_{ts} passing in opposite directions $s \to t$ and $t \to s$, respectively. Given the factorization of a tree-structured graphical model

$$p(x_1, x_2, \ldots, x_m) = \frac{1}{Z} \prod_{s \in V} \psi_s(x_s) \prod_{(s,t) \in E(T)} \psi_{st}(x_s, x_t)$$

the full collection of messages is updated according to the following recursion

$$M_{ts}(x_s) \leftarrow \kappa \sum_{x_t'} \left\{ \psi_{st}(x_s, x_t') \, \psi_t(x_t') \prod_{u \in N(t)/s} M_{ut}(x_t') \right\}$$

where $\kappa > 0$ is a normalization constant and ψ is the potential function. Figure 4.1 shows the procedure.

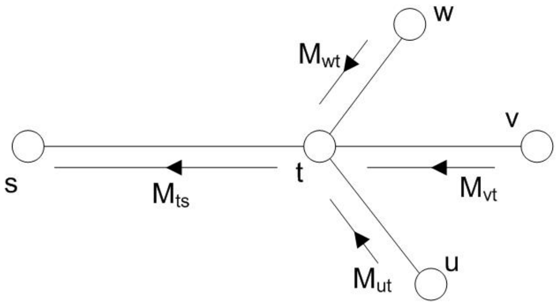

Figure 4.1 Message passing

It has been proved that for tree structure graphical models, the iteration will converge to a unique fixed point $M^* = \{M_{st}^*, M_{ts}^*, (s,t) \in E\}$ after a finite number of iteration [Kschischang, F. (2001)] [Pearl, J. (1988)].

We next define indication function for event $\{X_s = j\}$ and $\{X_s = j, X_t = k\}$ before we give the sum-product message passing form in Ising model

$$\theta_s\left(x_s\right)=\sum_j \theta_{s;j}\mathbb{I}_{s;j}\left(x_s\right),$$

and

$$\theta_{st}\left(x_s,x_t\right)=\sum_{(j,k)} \theta_{st;jk}\mathbb{I}_{st;jk}\left(x_s,x_t\right)$$

where

$$\begin{aligned}
\theta_s\left(x_s\right) &= \theta_{s;1}\mathbb{I}_{x_s=+1}+\theta_{s;-1}\mathbb{I}_{x_s=-1} \\
\theta_{st}\left(x_s,x_t\right) &= \theta_{st;+1+1}\mathbb{I}_{\substack{x_s=+1\\x_t=+1}}+\theta_{st;+1-1}\mathbb{I}_{\substack{x_s=+1\\x_t=-1}} \\
& \theta_{st;-1+1}\mathbb{I}_{\substack{x_s=-1\\x_t=+1}}+\theta_{st;-1-1}\mathbb{I}_{\substack{x_s=-1\\x_t=-1}},
\end{aligned}$$

Indication function $\mathbb{I}_{s;j}\left(x_s\right)$ takes 1 only if $x_s=j$, and takes 0 otherwise. The indication function $\mathbb{I}_{st;jk}\left(x_s,x_t\right)$ takes 1 only if $x_s=j$ and $x_t=k$ and takes 0 otherwise. Here the θs are any positive values satisfying the following two equalities

$$\begin{gathered}
\tfrac{\theta_{st;+1+1}+\theta_{st;-1-1}-\theta_{st;+1-1}-\theta_{st;-1+1}}{4}=\theta_{st} \\
\tfrac{\theta_{s;+1}-\theta_{s;-1}}{2}+ \\
\textstyle\sum_{t:(ts)\in\mathcal{E}}\tfrac{\theta_{st;+1+1}-\theta_{st;-1-1}+\theta_{st;+1-1}-\theta_{st;-1+1}}{4}=\theta_s.
\end{gathered}$$

We note that $\theta_{s;\pm1}$ and $\theta_{st;\pm1\pm1}$ are not unique, so we randomly select a set of values in our experiments.

Now given $\theta_{s;\pm1}$ and $\theta_{st;\pm1\pm1}$, the sum-product algorithm is presented in Algorithm 1 and the message passing procedure is illustrated in Figure 4.1.

It has been proved that for any discrete Markov random field in exponential family form with at most a single cycle, sum-product has a unique fixed point, and always converges to it from any initialization of the messages.

4.2 Mean Field

Mean field algorithm, which is another kind of variational approach to approximate the exact marginal probability given the joint distribution, can also be viewed as a kind of message

Algorithm 1 Sum-Product Algorithm

```
$\tau = 1$
while $M_{ts}^{\tau} \neq M_{ts}^{\tau-1}$ for some $(t, s)$ do
    Peer $t$ receives messages $M_{wt}^{\tau}$ from its neighbors $w \in N(t)$.
    Node $t$ computes message $M_{ts}^{\tau+1}$ as follows
        $M_{ts}^{\tau+1}(x_s) \leftarrow \kappa \sum_{x_t'} \left\{ \exp\left\{\beta\theta_{st}(x_s, x_t') + \beta\theta_t(x_t')\right\} \prod_{u \in N(t)/s} M_{ut}^{\tau}(x_t') \right\},$    (4.1)
    and then sends $M_{ts}^{\tau+1}$ to peer $s$.
    $\tau + +$
end while
Compute the marginal distribution
    $\Pr(X_s = x_s) = \kappa \exp\{\beta\theta_s(x_s)\} \prod_{u \in N(s)} M_{us}^{\tau}(x_s),$
where $x_s \in \{+1, -1\}$.
```

passing algorithm. Instead of vector messages, a mean value of probability distribution is passed as message between nodes. Since every variable represented by a node has a binary distribution, so we can compute the binary distribution from the mean value easily.

Now given θ_s and θ_{st} (not in indication function form), the mean field algorithm is presented in Algorithm 2. The message passing procedure is similar to that of sum-product algorithm.

Algorithm 2 Mean Field Algorithm

```
$\tau = 1$
while $\mu_t^{\tau} \neq \mu_t^{\tau-1}$ for some $t$ do
    Peer $t$ receives mean values $\mu_w^{\tau}$ from its neighbors $w \in N(t)$.
    Node $t$ computes mean value $\mu_t^{\tau+1}$ as follows
        $\mu_t^{\tau+1} \leftarrow \left\{ 1 + \exp\left[ -\left( \theta_t + \sum_{w \in N(t)} \theta_{wt}\mu_w^{\tau} \right) \right] \right\}^{-1}$
     and then update the old mean value with the new one.
    $\tau + +$
end while
```

The mean field problem is nonconvex in general, and the result can depend strongly on the initial value of mean. Despite these issues, this algorithm is known to become asymptotically exact for certain types of models as the number of nodes m grows to infinity.

4.3 Performance Metrics

We now define the performance metrics we use to evaluate the performance of tracking algorithms. For each peer i, we let $\mathcal{S}_i$ denote the set of files peer i has, and $|\mathcal{S}_i|$ therefore is the number of files peer i has. Now assume a peer (say peer j) is observed, then for each other peer i, we can compute the following quantity:

$$p_{i|j} = \frac{|\mathcal{S}_i \cap \mathcal{S}_j|}{|\mathcal{S}_j|}.$$

Now each file held by peer j can be viewed as a sample of illegal contents, and $\mathcal{S}_j$ is the sample space. Then $p_{i|j}$ is the empirical measure of the probability that peer i has the same (illegal) file as that of peer j. In our performance evaluation, we use $p_{i|j}$ as the ground truth. The goal of the tracking algorithm is to find those peers with large $p_{i|j}$.

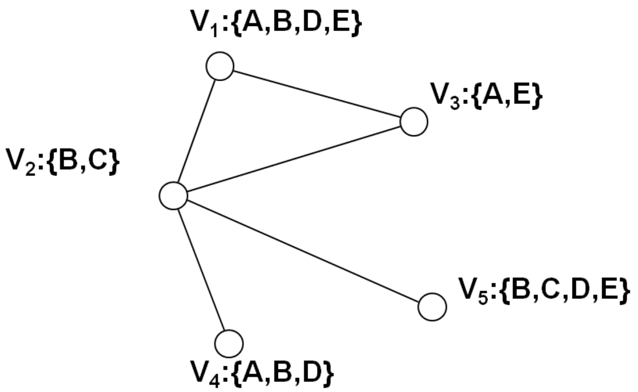

Figure 4.2 A five-node network example

Next we present a five-node network example as shown in Figure 4.2 to further explain the evaluation process. Assuming that peer V_5 is observed, $\mathcal{S}_5 \cap \mathcal{S}_i$ and $p_{i|5}$ are summarized in Table 4.1. Clearly, if peer 5 is known to have the target file, then peer 1 is mostly likely to have the target file as well.

Table 4.1 Statistics of the five-node example

Peer ID	1	2	3	4
$\mathcal{S}_5 \cap \mathcal{S}_i$	$\{B, D, E\}$	$\{B, C\}$	$\{E\}$	$\{B, D\}$
$p_{i\|5}$	0.75	0.5	0.25	0.5

We note that in practical systems, the calculation of $p_{i|j}$ requires the comparison between $\mathcal{S}_j$ and $\mathcal{S}_i$ for all peer i. Since the file set $\mathcal{S}_j$ may be of a large size, the calculation of $p_{i|j}$ can be computationally expensive.

We note that $p_{i|j}$ is the marginal distribution of peer i having a file given that peer j holds the file, so $p_{i|j}$ can also be computed using the sum-product and mean field tracking algorithm. The algorithm only requires the knowledge of link weights that can be obtained by comparing $\mathcal{S}_i$ and $\mathcal{S}_j$ for those peers who are neighbors, and can be implemented in a parallel fashion. So the sum-product and mean field tracking algorithms are very efficient methods for computing (or approximating) $p_{i|j}$.

The real interest of the tracking problem is to find the set of peers that have a high probability to hold the illegal content, so the most important output of a tracking algorithm is the ranking of the peers. We let $\mathcal{T}(i)$ denote the ranking of peer i under tracking algorithm $\mathcal{T}$, and $\mathcal{T}^{-1}(m)$ to be the identity of the peer who is ranked m under algorithm $\mathcal{T}$. We let $\mathcal{T}^*$ denote an "ideal" policy that can determine the ranking precisely.

Now assume peer i_o is observed, we consider the following two metrics: tracking rate and hitting rate.

- **Tracking rate:**

$$\mu(\mathcal{T}) = \frac{1}{N_t} \sum_{m=1}^{N_t} p_{\mathcal{T}^{-1}(m)|i_o},$$

  where N_t is the number of peers we plan to track.

- **Hitting rate:**

$$h(\mathcal{T}) = \frac{|\{i : \mathcal{T}^*(i) \leq N_t\} \cap \{i : \mathcal{T}(i) \leq N_t\}|}{N_t}.$$

We note that the tracking rate $\mu(\mathcal{T})$ is the probability of finding the target content in the top N_t peers ranked by algorithm $\mathcal{T}$. We later will compare this quantity with $\mu(\mathcal{T}^*)$, the average probability of finding the target content in the true top N_t peers. The hitting rate $h(\mathcal{T})$ is the number of peers that are *(i)* ranked as top N_t peers under $\mathcal{T}$ and *(ii)* also among the real top N_t list. We use this quantity to measure the accuracy of ranking output by $\mathcal{T}$.

4.4 Simple Tracking Algorithms

In our experiment, we also implemented following two simple tracking algorithms:

- **Random selection $\mathcal{T}^r$:** The algorithm ranks the peers randomly.
- **Size based selection $\mathcal{T}^s$:** The peers are ranked according to the number of files they have. The peer has the largest number of files is ranked as the top one peer.

The random selection does not exploit any information of the P2P data set, and the size based selection only uses the number of files a peer holds. The purpose of comparing the sum-product tracking algorithm and mean field tracking algorithm with these two algorithms is to show that the behavioral-based algorithm can significantly improve the tracking accuracy compared to simple tracking algorithms.

4.5 The Performance of Algorithms

We choose some network to evaluate our algorithms, that is sum-product and mean field tracking algorithms. First we choose a large size network containing 4,000 nodes which covers almost all the nodes. We randomly select a peer in the 4,000-node network and assume that the selected peer is the observed peer. Table 4.2 illustrates the top-twenty lists (i.e., $N_t = 20$) under various tracking algorithms assuming that peer 38 is observed. The tracking rate and hitting rate of the three tracking algorithms are shown in Figure 4.3. For the sum-product and mean field tracking algorithm, we choose the scaling factor β to be 0.3.

We observe that the sum-product tracking algorithm and mean field tracking algorithm significantly outperform the random selection and size-based algorithms. The tracking rate under the sum-product algorithm is 0.0758, and the mean field is 0.0728, which is close to the true value 0.0985; while the tracking rate of the random selection is 0.0019 and the size-based

Table 4.2 Result for one user given in some subnet

	1st	**2nd**	**3rd**	**4th**	**5th**
$\mathcal{T}^{sp}$	(1575,0.218)	(2874,0.172)	(1338,0.099)	(3099,0.026)	(2735,0.024)
$\mathcal{T}^{mf}$	(1575,0.218)	(2874,0.172)	(1338,0.099)	(3099,0.026)	(279, 0.055)
$\mathcal{T}^{r}$	(3429,0.018)	(2386,0.000)	(2319,0.000)	(3532,0.000)	(1374,0.000)
$\mathcal{T}^{s}$	(279,0.055)	(584,0.011)	(164,0.018)	(139,0.051)	(99,0.075)
$\mathcal{T}^{*}$	(1575,0.218)	(2874,0.172)	(179,0.139)	(3604,0.108)	(659,0.101)
	6th	**7th**	**8th**	**9th**	**10th**
$\mathcal{T}^{sp}$	(405,0.097)	(99,0.075)	(179,0.139)	(2878,0.095)	(3459,0.048)
$\mathcal{T}^{mf}$	(2735,0.024)	(405,0.097)	(99,0.075)	(179,0.139)	(2878,0.095)
$\mathcal{T}^{r}$	(2489,0.000)	(501,0.002)	(1840,0.002)	(1606,0.011)	(1564,0.000)
$\mathcal{T}^{s}$	(405,0.097)	(236,0.031)	(103,0.040)	(43,0.064)	(460,0.000)
$\mathcal{T}^{*}$	(1338,0.099)	(35,0.097)	(405,0.097)	(2878,0.095)	(141,0.084)
	11th	**12th**	**13th**	**14th**	**15th**
$\mathcal{T}^{sp}$	(279,0.055)	(269,0.051)	(556,0.079)	(3203,0.015)	(3299,0.031)
$\mathcal{T}^{mf}$	(3459,0.049)	(269,0.050)	(556,0.079)	(3203,0.015)	(3299,0.031)
$\mathcal{T}^{r}$	(2990,0.000)	(2497,0.004)	(3037,0.000)	(381,0.000)	(2279,0.002)
$\mathcal{T}^{s}$	(440,0.059)	(157,0.007)	(223,0.051)	(293,0.057)	(179,0.139)
$\mathcal{T}^{*}$	(3325,0.084)	(2149,0.081)	(334 ,0.079)	(556,0.079)	(99,0.075)
	16th	**17th**	**18th**	**19th**	**20th**
$\mathcal{T}^{sp}$	(1318,0.033)	(2168,0.037)	(659,0.101)	(1472,0.040)	(2149,0.081)
$\mathcal{T}^{mf}$	(1318,0.033)	(2168,0.037)	(659,0.101)	(2568,0.053)	(143,0.007)
$\mathcal{T}^{r}$	(3512,0.000)	(1918,0.000)	(781,0.000)	(2802,0.000)	(29,0.000)
$\mathcal{T}^{s}$	(281,0.004)	(12,0.026)	(214,0.018)	(1318,0.033)	(128,0.000)
$\mathcal{T}^{*}$	(300,0.075)	(49,0.073)	(1306,0.073)	(654,0.070)	(3480,0.070)

is 0.0418. The hitting rate of the sum-product tracking algorithm is roughly 50% and the mean field is roughly 45% while the hitting rate of the size-based algorithm is 15% and the random algorithm is almost 0%.

We repeated our experiment $4,000$ times by choosing different peers as observed peers. Figure 4.4 shows the average tracking $\overline{\mu}$ and average hitting rates $\overline{h}$. While the sum-product tracking algorithm only achieves 50% of the true tracking rate and 30% hitting rate; these Ising model-based algorithms still outperform the other two algorithms significantly. We notice that the performance of mean field is slightly worse than that of sum-product.

We also choose different middle-size network which contains about 1000 nodes. We first choose the user which has the smallest ID $Node_{start}$, then we pick all the nodes from $Node_{start}$

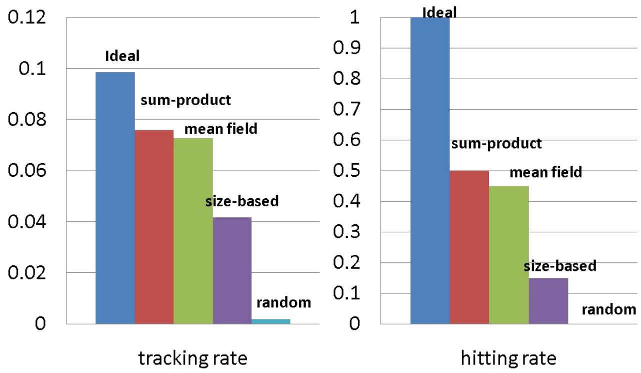

Figure 4.3 Tracking rate and hitting rate given peer 38 is observed

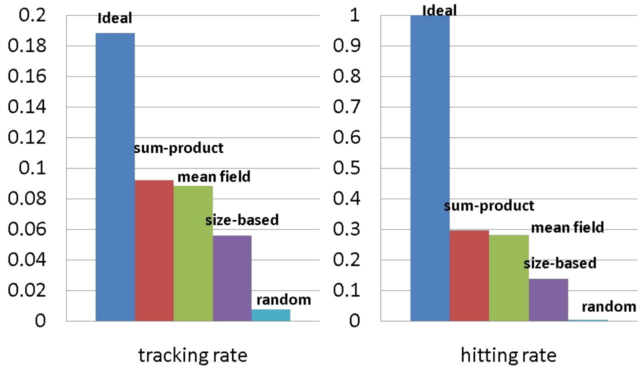

Figure 4.4 Average tracking rate and hitting rate over $4,000$ different observed peers

to $Node_{start+999}$. After removing all the isolated nodes in these 1,000 nodes group, we finally obtain the subnet to do experiment. We use *startID_size_*β to denote the experiment parameters that one specific subnet of tentative size *size* starting from *startID* under scaling factor β. We give the results of tracking rate and hitting rate in Table 4.3 and 4.4.

4.6 The Impact of Scaling Factor β

In the experiments above, we fixed the scaling factor $\beta = 0.3$ for the large network and $\beta = 0.5$ for middle networks. We observed that β plays an important role in the sum-product algorithm and mean field algorithm. We therefore varied the values of β from 10^{-299} to 1.3 for the large network. Figure 4.5 and 4.6 show the tracking rate and hitting rate under different

Table 4.3 Average tracking rate for different subnets when $\beta = 0.5$

avg tracking rate	**200_1k_0.5**	**300_1k_0.5**	**400_1k_0.5**	**500_1k_0.5**
$\overline{\mu}(\mathcal{T}^{sp})$	0.08128	0.08447	0.08490	0.08689
$\overline{\mu}(\mathcal{T}^{mf})$	0.08072	0.08432	0.08450	0.07682
$\overline{\mu}(\mathcal{T}^{r})$	0.00108	0.00121	0.00097	0.00111
$\overline{\mu}(\mathcal{T}^{s})$	0.04333	0.04187	0.04048	0.04513
$\overline{\mu}(\mathcal{T}^{*})$	0.09746	0.09954	0.09900	0.09972

Table 4.4 Average hitting rate for different subnets when $\beta = 0.5$

avg hitting rate	**200_1k_0.5**	**300_1k_0.5**	**400_1k_0.5**	**500_1k_0.5**
$\overline{h}(\mathcal{T}^{sp})$	0.4071	0.4199	0.4564	0.4089
$\overline{h}(\mathcal{T}^{mf})$	0.4058	0.4196	0.4560	0.3532
$\overline{h}(\mathcal{T}^{r})$	0.0098	0.0067	0.0076	0.0065
$\overline{h}(\mathcal{T}^{s})$	0.1729	0.1799	0.1923	0.2034
$\overline{h}(\mathcal{T}^{*})$	1	1	1	1

β's, respectively. For each β, we repeated the experiments for $4,000$ times (i.e., a different observed peer is selected at each time), and then computed the average tracking rate $\overline{\mu}$ and average hitting rate $\overline{h}$.

We can see that the sum-product algorithm has the best performance when β is between 0.05 and 0.4.

We can see if the scaling factor β is not properly chosen, the performance of both the sum-product algorithm and mean field algorithm would be deteriorated. We comment that when β is close to zero, the network consists of isolated nodes. The sum-product algorithm performs similarly to the random selection algorithm. On the other hand, when β is too large (i.e., the link strength is too strong),

$$\Pr(X_i = Y_j | X_j = Y_j) \rightarrow 1$$

for all peer i. In this case, the sum-product algorithm again ranks the peers randomly, and the performance of the sum-product algorithm is close to that of the random selection as well.

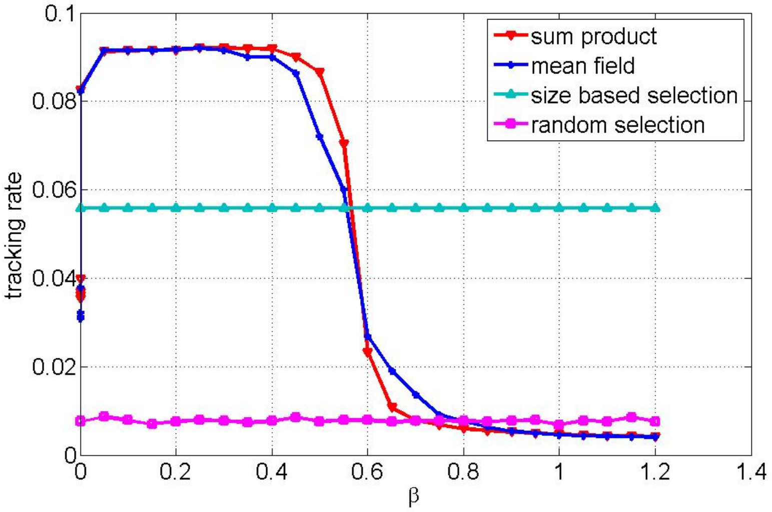

Figure 4.5 Average tracking rate under different β

4.7 Different File Sets

To further evaluate the performance of the sum-product and mean field tracking algorithm, we varied the file set. In the previous experiment, files that have been transferred for more than once during the 81-day period are all selected. Next we first restrict the file set to be those files that were transferred for at least three times. We again constructed the corresponding graph based on the restricted data set. We repeated the experiments $4,000$ times (a different observed peer was selected at each time) for the 4,000-node network, and computed the average tracking and hitting rates. The tracking rate and hitting rate under the sum-product, mean field algorithms with $\beta = 0.3$, size-based, and random algorithms are shown in Figure 4.7. Again, significant performance gain of the sum-product tracking algorithm can be observed.

We then repeated the experiment for the cases: *(i)* files that are transferred for at least four times; and *(ii)* files that are transferred for at least five times. Similar observations can be seen in Figure 4.8 and 4.9.

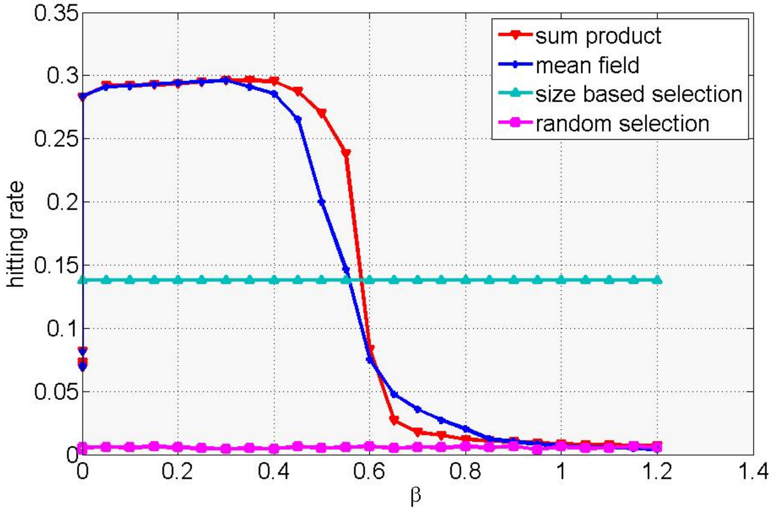

Figure 4.6 Average hitting rate under different β

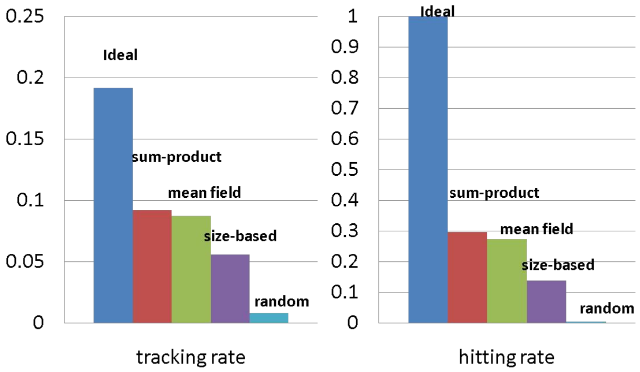

Figure 4.7 Average tracking rate and hitting rate for the case where only those files that were transferred for at least *three* times are included

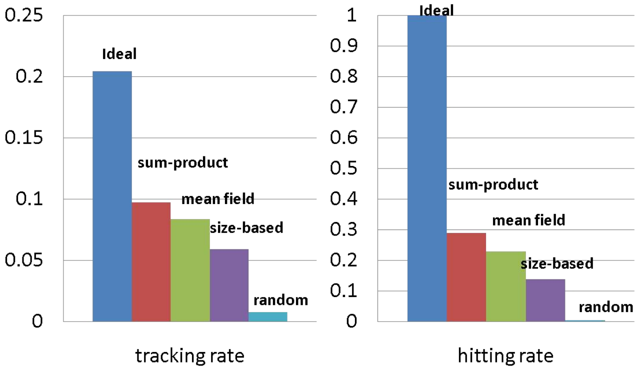

Figure 4.8 Average tracking rate and hitting rate for the case where only those files that were transferred for at least *four* times are included

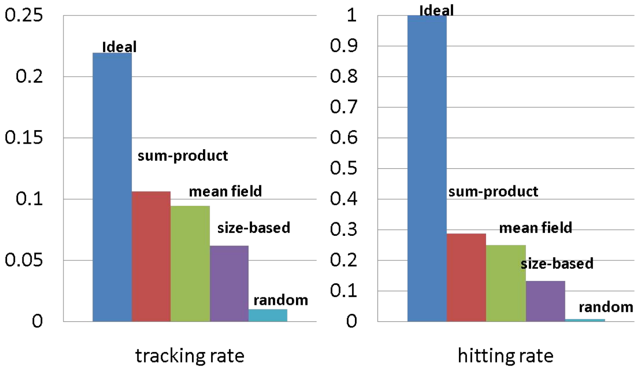

Figure 4.9 Average tracking rate and hitting rate for the case where only those files that were transferred for at least *five* times are included

CHAPTER 5. DISCUSSIONS AND CONCLUSIONS

In this thesis, we proposed using the Ising model to help search in P2P networks. We used the sum-product tracking algorithm and mean field tracking algorithm to get the conditional probability of one peer having a specific file given other observed peer. To quantitatively compare the performance, we used two evaluation metrics in experiment, tracking rate and hitting rate. Besides that we also use two simple algorithms as bench marks to do the experiment. We found that both sum-product algorithm and mean field algorithm perform significantly better than two other simple algorithms.

It can be seen that scaling factor β plays an important role in the performance of our Ising model based approach. When the β is not properly set, both the tracking rate and hitting rate of sum-product and mean field are deteriorated. In the original Ising model in statistical physics, this parameter represents the temperature. At the critical temperature the "spontaneous magnetization" would happen, this phenomenon is successfully explained by Ising model. The problem of how to find the optimal value for this parameter is left for future work.

In the experiments to evaluate the performance of the model and tracking algorithms, we first construct network from real data set, then we set one user as observed and use the set of all the files that user has as the sample space. The result for every unobserved node after sampling provides a trustful result of every node's probability of having the same file with that observed node. We use random selection and size based algorithms as two benchmark algorithms. And tracking rate and hitting rate are used to give the evaluation of performance.

It is natural to consider the case of multiple nodes being observed in experiment. Although it is not difficult to compute the probability of each node using both sum-product and mean field algorithms from our model, the reliable empirical probability which is used for verification is not easy to obtain. The reason is that the intersection of observed nodes' file set usually contains very few files, which makes the empirical probability after sampling unreliable.

Both mean field and sum-product algorithm can be classified as variational method which aim to approximate marginal probability. Mean field algorithm passes message of random variable's mean value, while sum-product algorithm passes a vector of information with each element corresponding to belief in one possible state of variable. From the result of tracking rate and hitting rate in our experiment, we can see sum-product algorithm outperform mean field algorithm slightly.

Besides mean field and sum-product algorithms, other methods like tree reweighted sum-product (TRW) could also give an approximation of marginal probability. The drawback of TRW is that it needs global information of the whole network (the topology) to determine the weight of spanning tree. Also different choice of spanning trees' weight lead to different approximation result.

BIBLIOGRAPHY

Wainwright, M. J. (2005). A new class of upper bounds on the log partition function. *IEEE Transactions on Information Theory*, *51*(7), 2313-2335.

Ising, E. (1925). Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, *31*(1), 253-258.

Koontz, L. D. (2003). File-sharing programs: Child pornography is readily accessible over peer-to-peer networks. *Technical Report GAO-03-537T.*

Ieong, R. , Lai, P. , Chow, K. , Kwan, M. , Law, F. , Tse, H. , and Tse, K. (2009).Forensic Investigation of Peer-to-Peer Networks. *Handbook of Research on Computational Forensics, Digital Crime and Investigation: Methods and Solution.*

Liberatore, M. , Erdely, R. , Kerle, T. , Levine, B. N. and Shields, C. (2010).Forensic Investigation of Peer-to-Peer File Sharing Networks. *Proc. DFRWS Annual Digital Forensics Research Conference.*

Fast, A. , Jensen, D. and Levine, B. (2005).Creating social networks to improve peer-to-peer networking. *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 568–573.

Barabasi, A. and Albert, R. (1999).Emergence of scaling in random networks. *Science 286*(5439), 509.

Jackson, M. (2008). Social and economic networks. *Princeton University Press.*

Networks, Crowds, and Markets: Reasoning About a Highly Connected World. *Cambridge University Press.*

Efron, B. (1978). The geometry of exponential families, annals of statistics. *Annals of statistics*, 362-376.

Wainwright. M. J. (2003). Graphical models, exponential families, and variational inference. *Technical Report, UC Berkeley, Department of statistics.*

Wainwright, M. and Jordan, M. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, *1*(1-2), 1-305.

Aji, S. M., and McEliece, R. J. (2000). The generalized distributive law. *Transaction on Information Theory*, *46*(2), 325–343.

Kschischang, F. , Frey, B. and Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *Transaction on Information Theory*, *47*(2), 498–519.

Yedidia. J. S. (2005). Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, *51*(7), 2282-2312.

Pearl, J. (1988). *Probabilistic reasoning in Intelligent Systems.* San Mateo: Morgan Kaufman.

ACKNOWLEDGEMENTS

This degree would not be possible without the help and guidance of my major professor, Dr. Lei Ying. I would like to take this opportunity to express my thanks to him for helping me in various aspects of conducting research and the writing of this thesis. I would also like to thank my committee members: Dr. Yong Guan and Dr. Aditya Ramamoorthy. I thank you for your invaluable help, instruction, patience and time during the development of my work. I would also like to thank my parents who always support me. Finally my special thanks go to all those who have helped me.