PRIVACY-PRESERVING NAÏVE BAYESIAN CLASSIFIER-BASED COLLABORATIVE FILTERING

Cihan KALELİ

Master of Science Thesis

Computer Engineering Program

May, 2008

JÜRİ VE ENSTİTÜ ONAYI

Cihan Kaleli'nin "**Basit Bayes Sınıflandırıcı Tabanlı Gizliliği Koruyan İşbirlikçi Filtreleme**" başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Yüksek Lisans Tezi 07.05.2008 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Yard. Doç. Dr. HÜSEYİN POLAT**	**…..…….**
Üye	**: Yard. Doç. Dr. CÜNEYT AKINLAR**	**..……..**
Üye	**: Yard. Doç. Dr. AHMET YAZICI**	**…….**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …… tarih ve …… sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ABSTRACT

Master of Science Thesis

PRIVACY-PRESERVING NAÏVE BAYESIAN CLASSIFIER-BASED COLLABORATIVE FILTERING

Cihan KALELİ

Anadolu University
Graduate School of Sciences
Computer Engineering Program

Supervisor: Assist. Prof. Dr. Hüseyin POLAT
2008, 75 pages

Collaborative filtering (CF) has become very popular on the Internet. Although CF systems are widely used, they have various challenges in recommendation process. The first one is collection of users' private data. For better referrals, such systems need quality data; however, due to privacy concerns, users hesitate to send their private data or they might send false data. The second challenge is that CF systems provide referrals on existing databases compromised of ratings recorded from groups of people evaluating various items; sometimes, the systems' ratings might be split among different parties. The parties may wish to share their data; but they may not want to disclose their data. The third challenge is optimizing problem. Online computation time increases with augmenting number of users/items.

In this thesis, approaches are proposed to overcome challenges for naïve Bayesian classifier (NBC)-based CF algorithm. A new scheme is proposed to produce NBC-based recommendations while preserving users' privacy by utilizing randomized response techniques (RRT). To offer CF services on distributed data between two parties without violating their privacy, solutions are provided. And finally, a method is proposed for optimizing privacy-preserving NBC-based CF scheme using *k*-modes clustering. To assess the proposed schemes, experiments are conducted using real data sets. The solutions are analyzed in terms of accuracy, privacy, and additional costs. After drawing conclusions, future works are presented.

Keywords: Collaborative Filtering, Privacy, Naïve Bayesian Classifier, Performance, Randomized Response Techniques

ÖZET

Yüksek Lisans Tezi

BASİT BAYES SINIFLANDIRICI TABANLI GİZLİLİĞİ KORUYAN İŞBİRLİKÇİ FİLTRELEME

Cihan KALELİ

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yard. Doç. Dr. Hüseyin POLAT
2008, 75 sayfa

İşbirlikçi filtreleme (İF) İnternet'te kullanılan çok popüler bir teknik haline gelmiştir. İF sistemleri çok yaygın kullanılmalarına rağmen bu sistemlerin bazı problemleri vardır. Bunlardan ilki kullanıcıların gizli verisini toplamaktır. Daha iyi önerilerde bulunmak için bu sistemler kaliteli veriye ihtiyaç duyarlar; fakat gizlilik nedeni ile kullanıcılar özel verilerini göndermekte tereddüt ederler veya yanlış veri göndermeye karar verebilirler. İkinci problem ise bazen öneri için kullanılacak veriler iki farklı grup arasında paylaşılmış olabilir. Bu iki grup verilerini birleştirmek isteyebilirler ama gizlilik endişelerinden dolayı birbirlerine verilerini göstermek istemeyebilirler. Üçüncü problem ise iyileştirme problemidir. Kullanıcı ve ürün sayılarının artması ile çevrimiçi hesaplama süresi artar.

Bu tezde, basit Bayes sınıflandırıcı (BBS) tabanlı İF algoritmasının sorunlarını gidermek için yöntemler önerilmiştir. Rastgele cevap teknikleri kullanılarak BBS tabanlı önerilerin kullanıcıların gizliliğini koruyarak gerçekleştirecek yeni bir yöntem sunulmuştur. İki grup arasında bölünmüş veriden bu grupların gizliliklerini koruyarak İF servisleri üretmek için çözümler önerilmiştir. Son olarak, *k*-mod kümeleme algoritması kullanılarak gizliliği koruyan BBS tabanlı İF algoritmasını iyileştirme yöntemi sunulmuştur. Çözümlerin doğruluk, gizlilik ve ek maliyetler açısından analizleri yapılmıştır. Sonuçlar açıklandıktan sonra gelecekte yapılması planlanan işler sunulmuştur.

Anahtar Kelimeler: İşbirlikçi Filtreleme, Gizlilik, Basit Bayes Sınıflandırıcı, Performans, Rastgele Cevap Teknikleri

ACKNOWLEDGEMENTS

I would like to thank my advisor Assist. Prof. Dr. Huseyin Polat for his guidance and support during my study. It was my pleasure to work with him during this study. Also, I would like to thank my fellow workers Ibrahim Yakut and Muzaffer Dogan for their scientific support.

Cihan Kaleli

May, 2008

CONTENTS

LIST OF TABLES

ABBREVATIONS

a : Active User

k : Number of Clusters

CA : Classification Accuracy

CF **:** Collaborative Filtering

F1 : *F*-Measure

HPD : Horizontally Partitioned Data

M : Number of Groups

n : Number of Train Users

NBC : Naïve Bayesian Classifier

PPCF : Privacy-Preserving Collaborative Filtering

PPDM : Privacy-Preserving Data Mining

TN : Top-*N* Recommendation

q : Target Item

RRT : Randomized Response Techniques

PL : Privacy Level

VPD : Vertically Partitioned Data

1. INTRODUCTION

Many e-commerce sites employ recommender systems to increase their sales while suggesting products to customers. Also, many search engine developers and vendors use recommender systems for increasing users' satisfaction by predicting user preference based on the user behavior. Recommender systems are implemented in commercial and non-profit web sites to predict the user preferences. For commercial web sites, accurate predictions may result in higher selling rates. The main functions of such systems include analyzing user data and extracting useful information for further predictions. These systems are designed to allow users to locate the preferable items quickly and to avoid the possible information overloads. Recommender systems apply data mining techniques to determine the similarity among thousands or even millions of data. There are three major processes in these systems: data collections and representations, similarity decisions, and recommendation computations. Recommender systems employ different techniques. Collaborative filtering (CF) is one of such techniques and it is widely used by many online vendors [10].

1.1 Collaborative Filtering

CF aims at finding the relationships between an active user (*a*) and the existing data, which contains lots of users' data to further determine the similarity and provide recommendations. It is an assumption that similar users have similar preferences in CF [20]. In other words, by finding users that are similar to *a* and by examining their preferences, the recommender system can predict *a*'s preferences for items and provide a ranked list of items, which *a* will most probably like. CF generally ignores the form and the content of the items and can therefore also be applied to non-textual items [20]. It can detect relationships between items that have no content similarities but are linked implicitly through the groups of users accessing them.

CF compares users according to their preferences [20]. Therefore, a database of users' preferences must be available. The preferences can be collected either

explicitly or implicitly. In the first case, the user's participation is required. The user explicitly submits her rating of the given item. Such rating can, for example, be given as a score on a rating scale from 1 to 5. The implicit ratings, on the other hand, are derived from monitoring the user's behavior. In the context of the Web, access logs can be examined to determine such implicit preferences. For example, if the user accesses the document, she implicitly rates it 1. Otherwise, the document is assumed to be rated 0 by the user.

There are two main approaches of CF algorithms. These approaches are memory- and model-based CF. In addition, there are hybrid CF approaches.

Memory-based algorithms utilize the entire user-item database to generate a prediction [61]. These systems employ statistical techniques to find a set of users, known as neighbors that have a history of agreeing with *a*. Once a neighborhood of users is formed, these systems use different algorithms to combine the preferences of neighbors to produce a prediction or top-*N* recommendation (TN) for *a*. The techniques, also known as nearest-neighbor or user-based CF, are more popular and widely used in practice.

Model-based CF algorithms provide recommendations by developing a model of user ratings. The model building process is performed by different machine learning algorithms such as Bayesian network, clustering, and rule-based approaches. The Bayesian network model [5] formulates a probabilistic model for CF problem. The clustering model treats CF as a classification problem [1, 39, 65]; and works by clustering similar users in the same cluster, estimating the probability that a particular user is in a particular cluster *C*, and from there computes the conditional probability of ratings. The rule-based approach applies association rule discovery algorithms to find association between co-purchased items and then generates item recommendation based on the strength of the association between items [59].

GroupLens [35, 55] introduce an automated collaborative filtering (ACF) using a neighborhood-based algorithm. The Ringo Music Recommender [62] and the Bellcore Video Recommender [23] describe a technique for making personalized recommendations from any type of database. Resnick and Varian [54] assume that a

good way to find interesting content is to find other people who have similar interests and then recommend titles that those similar users like.

Breese et al. [5] describe several algorithms for CF, including techniques based on correlation coefficients, vector-based similarity calculation, and statistical Bayesian methods. They compare the predictive accuracy of the various methods. In [4], Billsus and Pazzani present a learning algorithm that addresses the limitations of CF approaches. Their proposed method is based on dimensionality reduction through the singular value decomposition (SVD) of an initial matrix of user ratings. SVD is used for dimensionality reduction to improve the performance of CF algorithm [60]. Sarwar et al. [58] define and implement a model for integrating content-based ratings into a CF system. Data clustering and partitioning algorithms are applied ratings data in CF [41, 65]. Gupta et al. [21] adopt off-line principal component analysis (PCA) and clustering in an effort to develop a more efficient recommendation algorithm, which is a model-based algorithm. Fisher et al. [18] present Java-based framework for building and studying CF systems.

Miyahara and Pazzani [40] propose an approach for CF based on naïve Bayesian classifier (NBC). Chen and George [13] propose a Bayesian approach for the problem of predicting missing ratings from the observed ratings. A unified probabilistic framework is proposed by Popescul et al. [52] for merging collaborative and content-based recommendations. Herlocker et al. [22] present an algorithmic framework that breaks the prediction process into components; and they provide empirical results regarding variants of each component. Sarwar et al. [59] compare the performance of several different recommender algorithms and show the results. Chandrashekhar and Bhasker [9] introduce a new memory-based approach to ratings based CF. Unlike existing memory-based CF approaches, this approach exploits the predictable portions of even some complex relationships between users while selecting the mentors for an *a* through the use of the novel notion of selective predictability, which can be measured using the entropy measure.

Goldberg et al. [19] describe Eigentaste, a new algorithm that applies Pearson correlation coefficient to a dense subset of the ratings matrix. Lemire [37] modifies a

wide range of the filtering systems to make them scale- and translation invariant. Kleinberg and Sandler [34] identify certain parameters of mixture models and show that for any system in which these parameters are bounded, it is possible to give recommendations whose quality converges to optimal as the amount of data grows. Chen and Jin [11] propose a new CF algorithm based on influence sets. They define a new prediction computation method. Chen and Cheng [12] propose a novel CF methodology for product recommendation when the preference of each user is expressed by multiple ranked lists of items.

Pennock and Horvitz [46] propose a hybrid CF method, which is called personality diagnosis. Given a user preferences for some items, they compute the probability that she is of the same personality type as other users, and, in turn, the probability that she will like new items. Su et al. [64] propose hybrid CF algorithms, sequential mixture CF and joint mixture CF, each combining advice from multiple experts for effective recommendation. These proposed hybrid CF models work particularly well in the common situation when data are very sparse. Lekakos and Giaglis [36] propose recommendation approaches that follow the CF reasoning and utilize the notion of lifestyle as an effective user characteristic that can group consumers in terms of their behavior as indicated in consumer behavior and marketing theory.

1.2 Challenges of Collaborative Filtering

Although CF systems are very popular and widely used; they have some challenges. Today's filtering systems have a number of disadvantages [71]. The most important one is that they are a threat to individuals' privacy. The individuals share their data with data vendors, so there are several risks for individuals' privacy [14]. One of them is unsolicited marketing. Another risk is that users' profiles might be used in criminal case. Most online vendors collect customer buying information and preferences. Such data is valuable asset, and it has been sold when some e-companies suffered bankruptcy. Some people might divulge their information if they can get benefits. These benefits can be purchase discount, useful recommendations, and

information filtering. According to a survey conducted in 1999 [15], the privacy fundamentalists are concern about any use of their data and they generally unwilling to provide their data to web sites. The pragmatic people are also concerned about data use, but less than the fundamentalists. They often have specific concerns and they can be addressed using particular tactics. The marginally concerned users are generally willing to provide data to web sites under almost any condition, although they often express a mild general concern about privacy.

Two different data owners might want to merge their data for producing more accurate predictions while protecting their individual privacy. This is another privacy issue in CF systems. Prediction qualities of a filtering system might increase if these data owners are able to share their data for filtering services. Combining data may help CF systems to overcome difficulties caused by sparseness of data and to improve recommendations' accuracy.

CF systems can produce accurate referrals when numbers of users/items that they have increase. Although increasing the numbers of users/items improves accuracy level, they increase run time of the system, too. This is an important challenge for CF systems. For efficiency of CF systems, this challenge must be overcome.

1.3 Privacy-Preserving Collaborative Filtering

Canny proposes two schemes for privacy-preserving collaborative filtering (PPCF) [7, 8]. In the first one, he describes a new method for CF, which protects the privacy of individual data. His method is based on a probabilistic factor analysis model. Privacy protection is provided by a peer-to-peer protocol. The factor analysis approach handles missing data without requiring default values for them. In the second schema, he proposes an alternative model in which users control all of their log data. He describes an algorithm whereby a community of users can compute a public "aggregate" of their data that does not expose individual users' data. The aggregate allows personalized recommendations to be computed by members of the

community, or by outsiders. Canny uses homomorphic encryption to allow sums of encrypted vectors to be computed and decrypted without exposing individual data.

Berkovsky et al. [2] propose a novel approach to overcome the inherent limitations of CF (sparsity of data and cold start) by exploiting multiple distributed information repositories. These may belong to a single domain or to different domains. To facilitate their approach, they use LoudVoice, a multi-agent communication infrastructure that can connect similar information repositories into a single virtual structure called implicit organization. Repositories are partitioned between such organizations according to geographical or topical criteria. They employ CF to generate user-personalized recommendations over different data distribution policies. This approach eliminates the usage of server. Individuals provide their recommendations.

Hurt et al. [25] present a tool called “iOwl”, which addresses privacy concerns. They use an agent–based approach in a distributed environment to provide recommendations. They address the problem that, on one hand side, a vast amount of valuable data is created, while people surf the web and, on the other hand, these data are lost for further searches. iOwl is based on mining techniques to generate profile data out of the click stream. The system helps its users to share information. An agent collects meta data about the surfed web sites, process the data, and exchanges the results with other agents. This helps the user of the agent system to gain additional knowledge about her current interest.

Polat and Du [48] employ randomized perturbation techniques (RPT) to achieve PPCF. In their schemes, users perturb their data by adding randomly created numbers to their numerical ratings. Since the users perturb their data, the data owners can not learn the original ratings. Although users mask their ratings, CF systems can still produce accurate and private recommendations using their schemes. In [47], the authors discuss achieving private referrals on item-item similarities. They use randomized response techniques (RRT) to perturb users’ data. Partitioned data-based PPCF is discussed in [50]. They propose schemes to produce private recommendations on integrated data without affecting data owners’ privacy.

Moreover, they discuss privacy-preserving protocols for providing predictions on vertically or horizontally partitioned data. In [51], Polat and Du propose a PPCF with inconsistently masked data.

Parameswaran and Blough [45] propose a framework for obfuscating sensitive information in such a way that it protects individual privacy and also preserves the information content required for CF. The proposed framework also makes it possible for multiple e-commerce sites to share data in a privacy-preserving manner. They apply different obfuscation techniques to CF and study their affects to the prediction accuracy.

1.4 Privacy-Preserving Data Mining on Partitioned Data

Privacy-preserving data mining (PPDM) on partitioned data is an important subject in e-commerce. Ionnidis et al. [26] present an extremely efficient and sufficiently secure protocol for computing the dot-product of two vectors using linear algebraic techniques. They demonstrate superior performance in terms of computational overhead, numerical stability, and security. Vaidya and Clifton [67, 68, 69] present privacy-preserving methods for different data mining tasks on vertically partitioned data (VPD). In [67], they address the problem of association rule mining, where transactions are distributed across sources. In [68], they present a method for *k*-means clustering when different sites contain different attributes for a common set of entities. Each site learns the cluster of each entity, but learns nothing about the attributes at other sites. In [69], the authors propose a solution for privacy-preserving method for NBC-based CF on VPD.

Several existing cryptographic techniques are used to create a privacy preserving NBC for horizontally partitioned data (HPD). One of the studies for this purpose is proposed by Kantarcioglu et al. [32]. They show that using secure summation and logarithm, distributed NBC can be succeeded securely. Merugu and Ghosh [39] present a framework for clustering distributed data in unsupervised and semi-supervised scenarios, taking into account privacy requirements and communication costs. Kantarcioglu and Clifton [30, 31] present methods for

association rule mining over HPD and for computing *k-nn* classification from distributed sources without revealing any information about the sources or their data. In [43], Oliveria and Zaiane address the problem of protecting the underlying attribute values when sharing data for clustering. To achieve their goal, they propose a novel spatial data transformation method called Rotation-Based Transformation (RBT). This new method is independent of any clustering algorithm. It has a sound mathematical foundation, efficient, and accurate.

Vaidya and Clifton [66] introduce a generalized privacy-preserving variant of the ID3 algorithm for VPD distributed over two or more parties. Yu et al. [72] propose an efficient and secure privacy-preserving algorithm for support vector machine (SVM) classification over VPD. Ouyang and Huang [44] focus on the privacy-preserving association rules mining in the following situation: two parties, each having a private data set, wish to collaboratively discover association rules on the union of the two private data sets.

Kaya et al. [33] propose a privacy-preserving distributed clustering protocol for HPD based on a very efficient homomorphic additive secret sharing scheme. Bunn et al. [6] describe a two-party *k*-means clustering protocol that guarantees privacy. Their method is based on the existence of any semantically secure homomorphic encryption scheme.

1.5 Definitions

Filtering is a technique to find the most interesting and valuable information from a large amount of data. With information overload problem, filtering is becoming increasingly important.

Active User *(a)* is a customer or user who is looking for referrals for products that she has not purchased previously.

Train User (*n*)'s data is collected by recommender systems providers to offer referrals.

Rating (Vote) represents the preference of a user about an item or product. The users express their preferences about items by rating them. Ratings can be numerical or binary. In binary voting, users rate items as like (1) or dislike (0).

Recommendation (Prediction) is goal of CF systems. Such a predicted preference is called *recommendation*. Recommendations can be predictions for single items or TN, which is an ordered list of items that should be liked by *a*.

Target Item (*q*) is the item for which *a* is looking for referrals.

Server is the entity that gathers ratings of items from many users for filtering purposes, and provides CF services to active users based on the collected data.

1.6 Contributions and Summary of Experiment Results

There are three contributions in this thesis. First one is producing recommendations using NBC while preserving individuals' privacy with RRT. The second contribution is producing recommendations by using two different data owners' data while preserving their privacy. The last contribution is producing efficient and private recommendations for individuals with NBC. Privacy is achieved by using RRT and efficiency is succeeded by using *k*-modes clustering and the idea of fuzzy clustering.

In [28], it is proposed that private NBC-based recommendations can be produced by using RRT. Various parameters that affect privacy are explained and their effects to privacy and accuracy are shown in experiment results. According to experiment results, it can be said that accurate recommendations can be produced while preserving privacy by using the proposed approaches.

In [27], an approach is proposed for combining two different data owners' data for producing more accurate recommendations while preserving their privacy. It is shown that data owners can combine their vertically or horizontally split data. The experiment results show that more accurate predictions can be produced by using the proposed approach.

More efficient and private recommendations can be provided by using clustering techniques with NBC. Data owners cluster their data by using *k*-modes

clustering and produce recommendations based on the data in each cluster independently. Experiment results show that using *k*-modes clustering, run time of producing recommendations can be decreased and accuracy can be improved. Also, the results show that private recommendations can be produced while decreasing the run time of the producing recommendations.

2. PROVIDING PRIVATE RECOMMENDATIONS USING NAÏVE BAYESIAN CLASSIFIER

Many e-commerce sites employ CF techniques to increase their sales while suggesting products to customers. However, today's CF systems fail to protect users' privacy. Without privacy protection, it becomes a challenge to collect sufficient and high quality data for CF. With privacy protection; users feel comfortable to provide more truthful and dependable data. In this chapter, it is proposed to employ RRT to protect users' privacy while producing accurate referrals using NBC, which is one of the most successful learning algorithms. Various experiments are performed using real data sets to evaluate the privacy-preserving schemes. The effects of parameters on accuracy and privacy are analyzed.

According to the experiments results, it can be said that private recommendations can be produced using NBC with RRT. The results of the experiments are shown in Section 2.6. In the first part of the chapter, general idea about the proposed approach is discussed and in the following part, producing private recommendations is presented.

2.1 Introduction

With the advent of the Internet, e-commerce has become very popular. To increase their sales and have competitive edge over others, online vendors employ CF techniques, which are widely used for filtering and recommendation purposes. Providing accurate referrals are advantageous to online vendors because customers prefer returning to stores with better referrals and they search for more products to buy. Online shopping sites incorporate recommendation systems that suggest products to customers based on like-minded users' preferences about items they have ordered before or showed interest.

CF has many important applications in e-commerce, direct recommendations, and search engines [7]. With the help of CF, users can get recommendations about many of their daily activities. CF systems predict the preferences of *a*, based on the preferences of others. The idea in CF is that *a* will prefer those items that like-minded

users prefer, or that dissimilar users do not. Different approaches are employed for CF; and NBC is one of them and used for producing referrals.

Although CF systems have several advantages, they have a number of disadvantages [7]. The most important one of these disadvantages is threats to users' privacy. Without privacy protection, CF systems cannot produce good results. The individuals do not divulge true rating values when they do not feel comfortable about their privacy. There is a great potential for individuals to share all kinds of information; but the privacy risks are many and severe. Moreover, customer data is a valuable asset and it has been sold when some e-companies suffered bankruptcy. If privacy is protected, people feel comfortable to give private data and contribute more truthful data.

In this chapter, how to achieve private recommendations efficiently based on the NBC using RRT is investigated. The answers of the following questions are looked for: *How can users contribute their personal information for CF purposes without greatly compromising their privacy? How can the server provide referrals efficiently with decent accuracy without deeply jeopardizing users' privacy?* The goal of this chapter is to prevent the server from learning the true values of users' ratings and the items rated and/or unrated by users. Moreover, such goals should be achieved for *a*, too, because *a* also provides her private data to the server when requesting recommendations.

2.2 Naïve Bayesian Classifier

NBC is one of the most successful machine learning algorithms in many classification domains. Despite its simplicity, it is shown to be competitive with other complex approaches, especially in text categorization and content-based filtering tasks. Also, NBC is stable with respect to small changes to training data. NBC does not require large amounts of data before learning.

In [40], Miyahara and Pazzani employ NBC for CF, where they define two classes, like and dislike. Since customers vote items as like (1) or dislike (0), the sparse user ratings matrix includes Boolean values indicating whether the user rated

items as 1 or 0. *a*'s ratings for items are class labels of the training examples. In the user ratings matrix, other users correspond to features and the matrix entries correspond to feature values. The naïve assumption states that features are independent given the class label. Therefore, the probability of an item belonging to $class_j$, where *j* is like or dislike, given its *n* feature values, can be written, as follows:

$$p(class_j|f_1,f_2,...,f_n) \propto p(class_j)\prod_i^n p(f_i|class_j), \quad (2.1)$$

where both $p(class_j)$ and $p(f_i|classj)$ can be estimated from training data and f_i corresponds the feature value of *q* for user *i*. To assign a target item to a class, the probability of each class is computed, and the example is assigned to the class with the highest probability. Only known features and the data that both users commonly rated are used for predictions.

In [40], the authors propose two different types of CF algorithms. They first propose a user-based CF algorithm, which described above, by using NBC and they also propose a scheme which is item-based CF algorithm.

Although NBC is widely used, it has important challenges. First of all, it depends on whole user database so when number of users increases, performance of NBC algorithm decreases. Another challenge of NBC is not preventing users' privacy. If users do not feel comfortable about their privacy, they do not send their true data and the accuracy of the classification decreases.

2.3 Randomized Response Techniques

Warner [70] first introduces RRT as a technique to estimate the percentage of people in a population that has attribute *A*. The interviewer asks each respondent two related questions, the answers to which are opposite to each other. Using a randomizing device, respondents choose the first question with probability θ and the second question with probability 1- θ, to answer. The interviewer learns responses but does not know which question is answered.

Let Q_a be the sensitive question and $Q^c{}_a$ be its complement. For example,

Q_a = ``Have you ever used a sick day leave when you weren't really sick?" YES N0

Q_s^c= ``Have you never used a sick day leave when you weren't really sick?" YES N0 With Q_a probability, the answer will be true and the answer will be false with the probability Q_s^c (1- Q_a).

2.4 Providing Private Recommendations Using NBC

To achieve PPCF, a scheme is proposed in which before sending their ratings to the server, users disguise their data in such a way that the server will not be able to learn the true ratings and the truthful information about users' preferences. However, the disguising scheme should still be able to allow the server to produce accurate referrals. It is proposed to use the RRT to disguise private data. Although information from each individual user is scrambled, if the number of users and/or items is significantly large, aggregate data can be estimated with decent accuracy. Since NBC-based CF is based on aggregate values of a data set, it is hypothesized that by combining the RRT with the NBC-based CF algorithms, a decent degree of accuracy for PPCF can be achieved. To verify this, RRT is implemented for an NBC-based algorithm [40]. Experiments are performed to evaluate the proposed schemes and to show the effects of varying parameters. The new schemes are analyzed in terms of accuracy and privacy.

2.4.1 RRT-based Data Disguising

A typical ratings vector includes the votes and empty cells for unrated items. An example of a ratings vector for user u is V_u = (11 | 00 | 101), where | means not rated. To disguise V_u, u generates a random number (r_u) using uniform distribution over the range [0, 1]. If $r_u \leq \theta$, then u sends the true data, V_u. Otherwise, she sends the false data (exact opposite of the ratings vector), which is V_u' = (00 | 11 | 010), where V_u' is the vector that reverses the 1s in V_u to 0s and 0s to 1s; V_u' is called the opposite of V_u. With probability θ, true data is sent while false data is sent with probability 1-θ. Although the server has the ratings vectors, it does not know whether they are true or false data, because random numbers are only known by the users.

2.4.2 Achieving Private Referrals Using RRT

With privacy as a concern, the server should not be able to learn the users' true ratings values and rated items, including active users. Users might send false data for perfect privacy, but producing accurate predictions from this data is impossible. If they send actual data, finding high quality referrals is possible, but privacy is not preserved. Since CF systems should provide referrals within a small time, the new scheme should provide predictions efficiently. Achieving private referrals efficiently with decent accuracy is aimed. Since accuracy, privacy, and efficiency conflict, a good balance between them is wanted to be archived. Thus, both one-group and multi-group schemes are used. Since CF systems perform two tasks (prediction for a single item showing whether the item will be liked or disliked by a and TN of a sorted list of N items that should be liked by a), proposed privacy-preserving schemes should be achieved such tasks using the NBC.

In the one-group scheme [16], all ratings are put into the same group and all of them are either reversed together or left unaltered. Since the random numbers are only known by the users, the server cannot know whether users tell the truth or lie. The conditional probabilities estimated from masked data are the same as the ones computed from original data because all ratings are either reversed together or left the same. Thus, in this scheme, the same accuracy on masked data can be achieved as with the original scheme. Although decent accuracy is achieved, the privacy level is very low. If the server somehow learns the true rating for only one item, it can obtain true votes for all items.

Users can partition m items into m groups (m-group scheme); with each group containing only one item. For each group, users randomly decide whether to disclose its true or false rating. The users repeat this process for all groups; the random decisions are independent for each group. The m-group scheme is very secure because each rating is independently masked. But, accuracy might become very low. A compromise between the one-group scheme and the m-group scheme is to partition the items into M groups, where the RRT is used to perturb each group independently

and $1 < M < m$. The decision is the same for all items in the same group, but the decisions for different groups are independent.

Users group the items in the same way and disguise their ratings in each group independently. A user can send the true data for one group, while she can send the false data for the other groups. Due to independent data masking, even if the server knows information about one group, it will not be able to derive information about other groups. Although privacy improves compared to one-group scheme, accuracy decreases due to increasing randomness. The server uses collected masked data to provide CF services. Based on *a*'s query and her data, the server estimates class probabilities and provides referrals. Since the server can calculate $p(class_j)$ values from *a*'s data, the problem is how to $p(f_i|class_j)$ values from masked data.

It is still possible for the server to estimate the conditional probabilities because it is able to estimate the probabilities of having true or false data, given perturbed data. The server knows that the users send true or false data with probabilities θ and 1- θ, respectively. Moreover, it can employ the distribution of 1s and 0s in perturbed data to compute the probability of having true or false data. If the perturbed data is called Y_k and the true data X_k, then $X_k^{'}$ represents the exact opposite of X_k (or false data), where $k = 1, 2,\ldots, M$, and k shows the group name, the server needs to find $p(X_k|Y_k=X_k)$ and $p(X_k^{'}|Y_k=X_k)$ for each group, where $p(X_k|Y_k=X_k)+ p(X_k^{'}|Y_k=X_k)=1$. $p(X_k|Y_k=X_k)$ can be calculated using the Bayes' rule, as follows:

$$p(X_k|Y_k=X_k)=[\, p(Y_k|Y_k=X_k)p(X_k)]/p(Y_k=X_k), \tag{2.2}$$

where $p(Y_k|Y_k=X_k)$ is θ. The value of $p(Y_k=X_k)$ can be calculated from disguised data, while the value of $p(X_k)$ can be computed, as follows, using the facts that $p(Y_k|Y_k=X_k)= \theta$ and $p(Y_k= X_k^{'} \mid X_k)=1- \theta$:

$$p(Y_k=X_k)= \theta\, p(X_k) + (1- \theta)p(X_k^{'}) \tag{2.3}$$

Eq. (2.3) can be solved for $p(X_k)$, as follows, using the fact that $p(X_k)+ p(X^{'}_k)=1$

$$p(X_k)=[\, p(Y_k=X_k) + \theta - 1]/(2\theta -1) \tag{2.4}$$

The following is get after replacing $p(X_k)$ with its equivalent in Eq. (2.1):

$$p(X_k|Y_k=X_k) = [\theta^2 + \theta p(Y_k=X_k) - \theta]/ [2\theta p(Y_k=X_k)] - p(Y_k=X_{k2}) \qquad (2.5)$$

Since X_k and Y_k are ratings vectors, to find $p(Y_k=X_k)$, the server finds posterior probabilities for all items in each group k, selects the best one, and uses it as $p(Y_k=X_k)$. After finding $p(X_k|Y_k=X_k)$ values for each group, the server can now use them for providing predictions. The server needs to consider all possibilities to find the conditional probabilities because it does not know whether the received data is true or false. Since the disguised data can be true or false in each group, the ratings vector that the server received from a user can be one of the 2^M possible vectors of that user. Therefore, the server can estimate the conditional probabilities, as follows, where $CP = p(f_i|class_j)$:

$$CP = CP_{(Y1=T \wedge \ldots \wedge YM=T)} P^M + CP_{(Y1=T \wedge \ldots \wedge YM-1=T \wedge YM=F)} P^{M-1}(1-P) + \ldots$$

$$+ CP_{(Y1=T \wedge \ldots \wedge Y2=F \wedge \ldots \wedge YM=F)} P(1-P)^{M-1} + CP_{(Y1=F \wedge \ldots \wedge YM=\mathrm{F})}(1-P)^M, \qquad (2.6)$$

where $Y_k = T$ and $Y_k = F$ mean the server considers the data in group k is true and false, respectively. The results are only described up to five-group because undesirable performance for schemes beyond five-group makes them not very useful. The proposed multi-group schemes can provide accurate referrals because aggregate data is interesting rather than individual data items and since when users tell lie, they also reverse the rating of q, like in one-group scheme, the conditional probabilities calculated within the group that includes q stay same whether the data is true or false. Moreover, since q is assigned to the class with the highest probability, it is needed to compare class probabilities for a, rather than finding the exact class probability values.

The proposed scheme can be easily extended to provide TN. The server computes class probabilities for all a's unrated items, select those items will be liked by a, sorts them decreasingly according to class probabilities, and provides the first N items. Since online computation cost is critical, instead of finding referrals for all

unrated items, *a* asks recommendations for N_a items, where $N < N_a < m-d$, and *d* is the number of items rated by *a*.

2.4.3 Providing Full Privacy

It might be more damaging for a user to have it revealed that she voted an item (for example, a pornographic site or magazine) than to know what the specific rating is. To prevent the server from learning rated items, users randomly select some unrated items' cells to be filled with fake ratings. The number of cells to be filled depends on the user's privacy level. Before they disguise their data, first, each user *u* finds the number of unrated items (m_{ut}) and uniformly randomly creates an integer, (m_{ur}) over the range (1, γ). They then choose *f* number of cells, and fill them, where $f = m_{ur} * m_{ut}/100$. The server will not be able to learn the number of chosen cells. After filling them, users perturb their private data together with the filled cells. Each user *u* fills $[(m_{ur} * m_{ut})/(100 * 2)]$ randomly selected items' cells with 1 and the remaining cells with 0. The server will not be able to learn the ratio of true ratings. Since users fill empty cells with equal numbers of 1s and 0s, when there are enough users, the contributions of appended ratings to probability computations will be close to zero. The range over which m_{ur} is selected can be adjusted to achieve required levels of accuracy and privacy.

2.4.4 Preserving Active Users' Privacy

Three methods are proposed to protect *a*'s data. In the first one, *a* generates *Y* -1 random ratings vectors and sends them including the true ratings vector to the server, which finds referrals for the received vectors. It sends *Y* recommendations to *a*, who can distinguish the referral calculated from the true ratings vector. *a* can generate random vectors in such a way to get *Y* predictions instead of one. For business purposes, this is not desirable. In the second method, the 1-out-of-*n* Oblivious Transfer protocol [17] is used, which refers to a protocol where at the beginning of the protocol one party, Bob has *n* inputs $X_1,\ldots, X_n$ and at the end of the protocol the other party, Alice, learns one of the inputs X_i for some $1 \leq i \leq n$ of her

choice, without learning anything about the other inputs and without allowing Bob to learn anything about *i*. An efficient 1-out-of-*n* Oblivious Transfer protocol is proposed by [41]. The 1-out-of-*n* Oblivious Transfer protocol could be achieved with polylogarithmic (in *n*) communication complexity. In the last method, *a* also perturbs her private data like other users do and sends the disguised data to the server. In this case, accuracy is expected to be the lowest because more randomness is added. Among these three, the solution based on the 1-out-of-*n* Oblivious Transfer protocol is more efficient than the others, *a* sends *Y* -1 randomly generated vectors and her true ratings vector to the server. After finding referrals, the server uses the 1-out-of-*n* Oblivious Transfer protocol to send them. *a* receives only one prediction instead of *Y* recommendations.

2.5 Overhead Costs and Privacy Analysis

Privacy-preserving scheme does not introduce additional storage costs. The communication costs increase due to protecting active users' privacy. Active users send *Y* vectors rather than one vector. Besides, the 1-out-of-*n* Oblivious Transfer protocol is employed, which introduces additional communication costs. The scheme introduces extra computation costs. Although with increasing *M* values, computation costs increase exponentially, since 5-group scheme is employed, the computation costs are still acceptable. Moreover, protecting active users' privacy also increases computation costs because the server finds *Y* referrals for random vectors, rather than one for the actual one. Since privacy, accuracy, and efficiency conflict, it should be sacrificed from accuracy and efficiency. The parameters of the new proposed schemes can be adjusted to accomplish a fair balance.

The server does not know the rated items due to fake ratings. However, it can guess the randomly selected unrated items. The probability of guessing the correct m_{ur} is 1 out of 100. After guessing it, the server can learn the number of filled cells (*f*) with the help of empty cells in the perturbed vector when it is not totally filled. After guessing *f*, the probability of guessing the *f* randomly selected cells filled with 1s and 0s are 1 out of $C_{f/2}{}^{m_1'}$ and 1 out of $C_{f/2}{}^{m_0'}$, respectively. m_1' and m_0'

represent the number of 1s and 0s, respectively; and C^g_h represents the number of ways of picking h unordered outcomes from g possibilities. Thus, the probability of guessing the fake ratings is 1 out of $(100*(C_{f/2}{}^{m_1'})(C_{f/2}{}^{m_0'}))$. It can be similarly computed when the masked vector is totally filled.

Privacy can be measured with respect to the reconstruction probability (p) with which the server can obtain the true ratings vector of a user given disguised data. Thus, it can be defined the privacy level (PL) in terms of p, as follows [56]: PL = $(1 - p) * 100$, where p can be written in terms of $p(X_k|Y_k= X_k)$ and M:

$$p= [p(X_k|Y_k= X_k)]^M = [(\theta^2 + \theta Y -\theta)/(2\theta Y - Y)]^M \quad (2.7)$$

where $Y=p(Y_k=X_k)$. With increasing p, PL decreases. To decrease p, the randomness should be increased, which makes accuracy worse. With increasing M, p decreases, while PL increases. The value of p depends on θ, M, and the value of Y or X, where $X = p(X_k)$. Since the randomization process is conducted independently for different groups, PL increases with increasing M. When θ approaches 0.50, PL increases due to increasing randomness. PLs can be calculated on varying θ and M values and showed them in Table 2.1, where $X = 0.3$. As expected, PLs increase with decreasing θ from 1 to 0.51 and increasing M values.

Table 2.1 Privacy Levels with Varying M and θ Values

θ	0.51					0.60					0.70				
M	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
PL (%)	69	90	97	99	99.7	61	85	94	98	99.1	50	75	87	94	97

2.6 Experiments

Jester and MovieLens Million (MLM) data sets are used in the experiments. GroupLens at the University of Minnesota (www.cs.umn.edu/research/Grouplens) collected MLM. Jester [21] is a web-based joke recommendation system, developed

at University of California; Berkeley includes continuous ratings, while MLM consists of discrete votes. The ratings in Jester range from -10 to 10, while votes in MLM range from 1 to 5. Although Jester has 100 jokes, MLM has 3,592 movies. Jester has 17,988 users, while MLM has 7,463 users. Classification accuracy (CA) and F-measure (F1) are used for measuring accuracy. CA is the ratio of the number of correct classifications to the number of classifications. F1 is a weighted combination of precision (P) and recall (R), where $F1 = (2 * P * R) / (P + R)$.

Using the similar methodology conducted by [40], firstly, numerical ratings are transformed into two labels (like, dislike). Then, 500 test and 1,000 training users who have rated at least 80 movies from MLM are randomly selected. Also, 500 test and 1,000 training users who have rated at least 60 jokes from Jester are randomly selected. Finally, 60 rated items for MLM and 40 for Jester as a training set, and 20 items for MLM and Jester as a test set are randomly selected. For each a from the test set, referrals are found randomly selected 5 rated items. Each time, an item from test set is selected, and a prediction on masked data is found. Since data is disguised based on the relation between the random numbers and the θ, data disguising is run 10 times and 10 referrals are found on masked data. The final results for a are then averaged over all trials. Finally, the average value over 500 active users is found and it is displayed. It is hypothesized that privacy and accuracy depend on n, θ, M, d, and f.

To show how number of features affects the result, trials are performed while changing n from 100 to 1,000 for both data sets. θ is fixed at 0.70 and employed three-group scheme. Since CA and F1 values are similar, only CAs are shown in Table 2.2. As expected, the results become better with increasing n values.

Table 2.2 CA with Varying *n* Values

	Jester				MLM			
n	100	200	500	1,000	100	200	500	1,000
Original Data	68.28	68.56	69.45	69.48	74.24	77.30	79.80	80.28
Masked Data	58.45	61.23	63.92	65.56	72.40	75.34	78.40	79.58

They also converge to the results on original data with increasing n because aggregate data can be estimated with decent accuracy if enough data is available.

Accuracy varies for different θ values because randomness differs. Trials are performed, where 200 training users from Jester and MLM are used. M is set at 3, where θ is varied from 0.51 to 1.00 because complementary θ values give the same results. CAs and F1s are shown in Table 2.3. When θ is 1, the same accuracy with original data is achieved because users send true data. However, when θ is 0.51, largest randomness is added; and with decreasing θ values towards 0.51, accuracy worsens. Accuracy is more likely to improve when more features are used because 200 features are only employed.

Table 2.3 Accuracy with Varying θ Values

	Jester				MLM			
θ	0.51	0.70	0.85	1.00	0.51	0.70	0.85	1.00
CA (%)	55.52	61.23	63.23	68.56	75.00	75.34	76.96	77.30
F1 (%)	57.98	62.45	62.89	73.68	85.27	86.94	89.78	90.89

To show how data partition affects the results, experiments are performed with varying M. 200 training users from Jester and MLM are used, where $\theta = 0.70$. The experiments are performed for up to five-group scheme. Since the results show similar trends for both data sets, only MLM's results are shown in Table 2.4. As seen from the table, the results become better with decreasing M values because less randomness is added to original data. Up to five-group scheme, it is still possible to provide accurate private referrals.

Table 2.4 Accuracy with Varying M Values

	CA (%)				F1 (%)			
M	1	2	3	5	1	2	3	5
	77.30	77.12	75.34	65.45	90.89	89.54	86.94	76.34

To show how various numbers of rated items (*d*) values affect the results, experiments are performed with varying *d*. 200 training users from Jester and MLM are used. The θ value is set at 0.70 and *M* at 3. Since CAs and F1s are similar, only CAs is shown in Table 2.5. With increasing *d*, number of data involved in referral computations increases. That makes accuracy better. Again, when large enough data available for CF purposes, it is possible to produce accurate referrals estimated from masked data. As expected and seen from Table 2.5, the results improve with increasing ratings provided by *a*.

Table 2.5 Accuracy with Varying *d* Values

	Jester				MLM				
d	< 25	25 < *d* < 40	40 < *d* < 60	> 60	< 25	25 < *d* < 40	40 < *d* < 60	60 < *d* < 80	<80
CA (%)	53.72	55.86	58.60	61.23	65.67	67.10	70.14	72.56	75.34

To show how accuracy changes with varying number of randomly filled cells (*f*), experiments are performed while varying γ from 0 to 100. With increasing γ, *f* increases; thus, more randomness is added. 500 training users from Jester and MLM are used, where θ is set at 0.70 and *M* is set at 3. Only F1*s* values are shown in Table 2.6. When γ is 0, empty cells are not filled with fake ratings. As expected, accuracy worsens with increasing γ due to increasing randomness. However, accurate recommendations are still provided using proposed schemes. The parameters can be adjusted to achieve required level of accuracy. Generally speaking, the results for Jester seem to be worse than the results for MLM because Jester has limited number of items (only 100 items).

2.7 Conclusions

Solutions to achieving private referrals on the NBC using RRT are represented. The solutions make it possible for servers to collect private data without greatly compromising users' privacy. Experiment results show that the schemes allow providing referrals with decent accuracy. To obtain a balance between accuracy, privacy, and efficiency, the parameters of the schemes can be adjusted.

Table 2.6 Accuracy with Varying *f* Values

	Jester				MLM			
γ	0	30	50	70	0	30	50	70
F1 (%)	63.49	62.59	61.19	59.49	89.63	84.04	83.25	82.27

Each parameter has different effects on accuracy and privacy. For θ, if more private recommendations must be produced it must be 0.51. If θ increases from 0.51 to 1 accuracy increases but privacy decreases. It gives the same result when decreases from 0.51 to 0. *n* has an effect on accuracy; if it increases, more accurate predictions are produced. *M* has effect on both privacy and accuracy and also it has effect on efficiency. If server divides users into more groups, it produces more private recommendations, but the accuracy of produced recommendations decreases. Also, with increasing *M*, computational time of producing recommendations increases. According to experiment results, *f* has the same effect on accuracy and privacy like *M*. *d* has an effect on accuracy, as well. If it increases, accuracy increases, too.

3. PROVIDING NAÏVE BAYESIAN CLASSIFIER–BASED PRIVATE RECOMMENDATION ON PARTITIONED DATA

Providing private CF services on partitioned data is becoming imperative. Data collected for CF purposes might be split between various parties even competing companies. Integrating such data is helpful for both e-companies and customers due to mutual advantageous. However, due to privacy, financial, and legal reasons, data owners do not want to disclose their data. In this chapter, it is hypothesized that if privacy measures are provided, data holders might decide to integrate their data to perform richer CF services and overcome the problems caused by inadequate data and/or sparseness. How to achieve NBC-based CF tasks on partitioned (horizontally or vertically) data with privacy is investigated. Randomized schemes are proposed to achieve privacy. Several experiments are performed on real data to evaluate the schemes' overall performance. Finally, experimental outcomes are analyzed and some suggestions are provided.

3.1 Introduction

With the evolution of the Internet, the number of users accessing the Internet and the number of products available online is rapidly increasing. To reach the most valuable and interesting information is very important. Customers want to buy products that they might like over the Internet and wish for selecting such products without wasting too much time. On the other hand, e-companies want to keep their existing customers and recruit new ones. One way to achieve such goals for both customers and e-commerce sites is to use recommender systems. Customers get recommendations about products they want to purchase, while e-companies might increase their sales by providing truthful referrals. CF techniques are used by online vendors for filtering and recommendation purposes. It has important applications in e-commerce, search engines, and direct referrals. Users can get recommendations about their activities using CF. The goal is to predict the preferences of *a*, based on a database consisting of a set of votes corresponding to the ratings of users on items [4, 46]. CF systems provide either predictions for single items or TN.

To provide more truthful and dependable referrals, data collected for CF purposes should be large enough. It is impossible to produce recommendations from insufficient data. To produce accurate and trustworthy referrals, there should be enough neighbors that are selected based on sufficient commonly rated items. With increasing available data (increasing number of users and items), it is more likely to have enough neighbors and matching between *a* and her neighbors. Many online vendors, especially those newly established ones, might not have enough data for CF purposes. Therefore, they might face with the cold start problem and are able to produce referrals for only a limited number of products. When there are a limited number of users, it becomes a challenge to form a large enough neighborhood. Moreover, some vendors might own ratings for a limited number of items; and that makes it harder to compute the similarities between *a* and other users because such values are computed on commonly rated items.

Data collected for CF might be partitioned horizontally or vertically between various parties, even competing companies. In horizontal partitioning, data owners hold disjoint sets of users' preferences for the same items. However, in vertical partitioning, they own disjoint sets of items' ratings collected from the same users. Combining horizontally partitioned data (HPD) is helpful when CF systems own a low number of users. Integrating vertically partitioned data (VPD) is advantageous when data holders have ratings for a limited number of items. Some users buy books from Amazon.com, while others get them from Barnes & Noble.com. Amazon.com's and Barnes & Noble.com's databases including ratings for the same books recorded from disjoint sets of users, can be jointly used for better referrals. Moreover, an individual's ratings for products might be split among different online vendors such as Amazon.com and MovieFinder.com. Amazon's and MovieFinder's databases including ratings for books and movies, respectively, recorded from the same customers, can be jointly used to produce better predictions. A referral computed from the joint data is likely more accurate and reliable than the one calculated from one of the disjoint data sets alone. However, due to privacy concerns, legal issues,

and financial reasons, data owners do not want to collaborate and disclose their data to each other.

In this chapter, it is explored how to provide CF services from partitioned data between two parties, without greatly exposing their privacy, using the NBC-based CF algorithm proposed. This chapter's goal is to provide accurate referrals efficiently from partitioned data with privacy, as follows: First, data holders should not be able to figure out the true ratings and rated items in each other's databases. Second, the referrals calculated from partitioned data with privacy concerns should be close to those referrals computed from combined data without privacy concerns. And finally, additional costs such as storage, communication, and computation costs, introduced due to privacy concerns, should be negligible and make it possible to provide referrals to many users in an acceptable time. As generally known, privacy, accuracy, and efficiency are conflicting goals.

PPDM on partitioned data has been receiving increasing attention. Sanil et al. [57] describe an algorithm to conduct a linear regression analysis based on VPD. Vaidya and Clifton present privacy-preserving methods for association rule mining [67], *k*-means clustering [68], and NBC [69], on VPD. Although such approaches are based on VPD, both VPD- and HPD-based CF with privacy using NBC are studied. PPCF on VPD problem is discussed in [49]. Unlike their study in which they show how to achieve predictions from numerical ratings, in this chapter, it is investigated how to provide CF tasks based on VPD and HPD using binary ratings employing NBC.

Privacy-preserving NBC for HPD is discussed in [32]. They show that using secure summation and logarithm, they can learn distributed NBC securely. Kantarcioglu and Clifton [30] discuss privacy-preserving association rules on HPD. They address secure mining of association rules over HPD while incorporating cryptographic techniques to minimize the shared data. Polat and Du [50] discuss PPCF on HPD using item-based algorithms. Unlike these works, it is explored partitioned data-based CF with privacy employing NBC, where users' preferences are represented with binary ratings. Moreover, the schemes can be easily extended to

multi-party schemes. Unlike the works studied so far, it is investigated both VPD and HPD-based CF services (predictions and TN recommendations) using NBC, where users' preferences are represented with binary ratings.

3.2 Partitioned Data-based PPCF Using NBC

Without privacy as a concern, two vendors can integrate their data to perform richer CF services. However, due to privacy concerns, they do not want to reveal their data. The challenge is how to achieve CF tasks privately from split data. Data owners should not be able to learn the true rating values and the rated items in each other's databases, while they are able to provide CF services on the integrated data. PPCF schemes to achieve CF tasks using NBC from partitioned data are proposed. It is assumed that the parties communicate through *a* during providing recommendations online. Also it is assumed that one of the parties acts as a master site to produce the recommendations after getting required data from the other party and such task can be swapped between them.

To derive information from each other's databases, data holders can employ different attacks. The proposed privacy-preserving schemes should be secure against such attacks, which can be explained, as follows: Data owners can act as an *a* in several times. The party acting as an *a* employs the same ratings vector during the all recommendation computation processes, manipulating only one rating value each time. Since it gets some conditional probability values computed using its ratings and the users' ratings in the other party's database, the party acting as an *a* can easily figure out the differences between such probabilities computed successively. Based on such differences, it is able to find out the ratings of the item for which the rating is manipulated or it can learn whether such item is rated or not. The proposed schemes, therefore, should be secure against such attacks, which might come from both parties.

Data holders can offer some incentives (discounts or coupons) or bribery to the users who provided data for filtering services. They then can obtain some data from users and try to derive more information about each other's databases. Since both parties can bribe the same users to derive data or to manipulate each other's

data, the required data through such bribed users may not be true or trusted. These users can employ such offers against the other party to get more discounts or coupons. This kind of attack becomes expensive and the derived data through this attack become questionable and doubtful.

3.3 Privacy-Preserving HPD-based Schemes

In horizontal partitioning, the companies hold disjoint sets of users' preferences for the same products. Two vendors, *A* and *B*, n_A and n_B users' ratings, respectively, of the same *m* items. They perform CF tasks using the joint data, which is an $(n_A + n_B) * m$ matrix while preserving their privacy. It would be difficult to find out whether two users from different online vendors refer to the same person or not. This can be solved by using some unique identities provided by e-companies to customers for online shopping. The identities of users can be exchanged offline. Since data is partitioned between *A* and *B*, Eq. (3.1) can be written, as follows, where *n* is the number of users and $n = n_A + n_B$:

$$p(c_j|f_1,\ldots,f_n) \propto p(c_j) * P_{Aj} * P_{Bj} = p(c_j) * \sum_{i=1}^{nA} p(f_i|c_j) * \sum_{i=nA+1}^{n} p(f_i|c_j) \quad (3.1)$$

where P_{Aj} and P_{Bj} represent the products of conditional probabilities computed from data belonging to *A* and *B*, respectively. When *B* acts as a master site, *a* computes the required data, P_{Aj} values, and sends it to *B* through *a*. HPD-based scheme with privacy can be explained, as follows:

1. *a* sends her data to both *A* and *B*. Since *B* is the master site and has *a*'s data, it can compute $p(c_j)$ values.
2. Since both *A* and *B* own the feature ratings of *q*, they can compute the conditional probabilities for classes like and dislike.
3. *A* then computes P_{Aj} values and sends them to *B* through *a*, while *B* computes P_{Bj} values.
4. Finally, *B* can find the probabilities of *q* belonging to c_j using Eq. (3.1).

B will not learn the true ratings and the rated items in *A*'s database, because it only gets P_{Aj} values, which are products of n_A values, from *A*. When *A* has only one

known feature value available for some items, *B* might be able to derive data by acting as an *a* in multiple scenarios. However, *B* does not know which feature value is known and how many known features *A* has. Even if *B* is able to learn such information, to prevent *B* from deriving data, *A* can introduce bogus known features (insert 1 or 0 into randomly selected cells of *q*). To further improve privacy, before sending P_{Aj} values to *B*, *A* multiplies such values with the same value r_A, where r_A is a random number generated by *A*. Since both values are multiplied by the same number, the comparison between $p(c_j|f_1, f_2. . . , f_n)$ values will not be changed for *j* being like or dislike.

3.4 Privacy-Preserving VPD-based Schemes

In vertical partitioning, the vendors own disjoint sets of items' ratings collected from the same users. *A* and *B* hold m_A and m_B items' ratings, respectively, where $m = m_A + m_B$. To make the data sharing possible, the identity of the products should be established across the data holders' databases. This data exchange can be achieved between vendors offline.

In VPD-based schemes, *a* sends the corresponding data to *A* and *B*. However, even if *a* does that, one party can act as an *a* in multiple scenarios to derive data from other party's data set. Therefore, the proposed VPD-based schemes should be secure against such attacks and it can be assumed that *a* sends her entire data to the master site or the site having *q*. Since the master site needs *a*'s known ratings to compute $p(c_j)$ values, instead of sending all known ratings, *a* can compute such values and sends them to the master site together with the corresponding ratings. Ratings of *a* are held by one of the vendors, because data is split vertically. Therefore, the party, which does not have *q*, should conduct the required computations and send the results to the company that owns *q*; and such party acts as a master site. The party not having *q* should be able to compute corresponding results required to find the conditional probabilities in such a way to prevent the master site deriving information from its data set. Since class probabilities are known by the master site, it needs to compute the conditional probabilities, as follows:

$$p(fi|c_j) = \frac{\#(f_i \mid c_j)}{\#(c_j)}, \tag{3.2}$$

where $\#(fj|c_j)$ shows the total number of similarly rated items of c_j as the feature value of q for corresponding user; and $\#(c_j)$ represents the total number of commonly rated items as j, where $j \in$ {like; dislike}. Since data is partitioned between A and B vertically, the master site gets the results from other party to find the conditional probabilities. Therefore, Eq. (3.2) can be written, as follows:

$$p(fi|c_j) = \frac{\#_A(f_i|\ c_j) + \#_B(f_i|\ c_j)}{\#_A(c_j) + \#_B(c_j)}, \tag{3.3}$$

where A and B compute the corresponding parts of $\#(fi|cj)$ and $\#(c_j)$ values. Assume that B owns q and acts as a master site. A then should compute $\#_A(f_i|c_j)$ and $\#_A(c_j)$ values for all $i=1,2,\ldots,n$ and j being like (1) or dislike (0); and sends them to B. VPD-based scheme with privacy can be explained, as follows:

1. a sends her corresponding data to A and B. a also computes $p(c_j)$ values and sends them to the master site, B.
2. Since q is held by B, A does not know which features of q are known; and therefore, it computes the corresponding parts of conditional probabilities for all features. Moreover, since A does not know feature values of q, it should compute such values twice, one for assuming $f_i = 1$ and one for assuming $f_i = 0$. However, A needs to compute $\#_A(f_i|c_j)$ values for classes like and dislike for only f_i being 1 or 0 because $p(f_i = 1|c_j) + p(f_i = 0|c_j) = 1$. After receiving such values from A, B will be able to select and/or find the required data to find the conditional probabilities because it knows the known features of q and their values.
3. Since B gets $p(c_j)$ values from a, it then can figure out how many 1s, 0s, and empty cells are in a's vector. Such information may help B derive data from A's database. Moreover, B can act as an a in multiple scenarios. Therefore, A should compute $\#_A(f_i|c_j)$ and $\#_A(c_j)$ values in such a way to prevent B deriving data from its database. To do so, A employs the following steps: It first finds

the number of empty cells (m_{ae}) in corresponding part of a's vector. A then uniformly randomly selects a value, R_A, over the range [1, 100]. A then can fill randomly selected R_A percent of these m_{ae} empty cells ($f = m_{ae} * R_A/100$) with random ratings (1s and 0s). However, with increasing randomness, accuracy diminishes. Instead of filling empty cells with random ratings, A can fill them with default votes (v_ds) of items it holds. Therefore, A finds the v_ds for m_A items it holds. Finding such ratings is explained in the following sub-section. A finally fills empty cells with the corresponding v_ds. A is able to randomly selects empty cells in such a way that p(c_j) values will not be changed. The number of empty cells to be filled depends on how much privacy and accuracy the parties want. With increasing numbers of filled cells, randomness increases; thus, accuracy diminishes.

4. A then computes the corresponding parts of conditional probability values $\#_A(f_i|c_j)$ and $\#_A(c_j)$ values) based on a's new or filled ratings vector.
5. Since B does not know how many and which empty cells are selected to be filled, it cannot derive information from the received data. Moreover, since empty cells are filled with non-personalized ratings, which are only known by A, B does not know such values, either.
6. After B gets the required data, it finds the final conditional probabilities, the probabilities for q belonging to c_j , and finally sends the prediction to a.

The new schemes can be extended to provide TN. The master site computes class probabilities for N_a items, where $N < N_a < m - m_r$ and m_r is the number of items rated by a. It selects those items will be liked by a, sorts them decreasingly according to class probabilities, and provides the first N items to a as TN.

3.5 Finding Default Votes

Both parties own the all ratings for items they hold. Therefore, they can compute non-personalized votes for the items they hold without the help of each other, as follows: For each item's column, they find the total number of 1s (l) and 0s (d). They then compare l and d values for each item. If $l > d$, then default vote (v_d) for

that item is 1, it is 0 otherwise. Both parties finally store non-personalized ratings and later use them for data disguising. Such ratings are computed offline.

3.6 Overhead Costs and Privacy Analysis

Proposed schemes are analyzed in terms of additional costs due to privacy concerns. The extra storage cost is negligible because *A* and *B* need to store $v_d s$ into 1 * m_A and 1 * m_B matrices, respectively. As expected, partitioned data-based schemes introduce additional communication costs in terms of both number of communications and amount of data. For single predictions, in HPD- and VPD-based schemes, additional number of communications is only 3 because *a* sends her data to both parties and one party sends the required data to the master site through *a*. Moreover, the amount of data sent also increases because one party sends either aggregate values in HPD-based schemes or two vectors of length *n* including the corresponding parts of conditional probabilities, where *n* is the number of features. The HPD-based schemes do not introduce additional computation costs. However, VPD-based schemes introduce extra computation costs due to randomly inserted non-personalized ratings. Number of comparisons increases because more ratings are available after inserting $v_d s$ into *a*'s vector. Computing $v_d s$ is done offline, which is not critical for overall performance.

The HPD-based schemes are secure due to the following reasons: *B* will not be able to learn the true ratings and the rated items even if it acts as an *a* in multiple scenarios, because it receives two aggregate values, which are products of n_A values. VPD-based schemes are also secure. Even if the master site knows *a*'s ratings, since only commonly rated items between *a* and other users are used for recommendation computations, it will not be able to derive data from other party's data. Finding $v_d s$ is secure because the parties do not need each other's data to find them. They will not learn such values held by each other. Due to randomly inserted $v_d s$, *B* will not be able to derive data from the corresponding parts of conditional probability values. The parties are able to disguise *a*'s data in such a way to achieve required levels of privacy and accuracy.

The master site does not know the rated items and the true rating values due to randomly selected empty cells and $v_d s$. However, it can guess the randomly selected unrated items. The probabilities of guessing the correct *RA* and m_{ae} are 1 out of 100 and 1 out of m_A, respectively. After guessing them, it can compute *f*. The probability of guessing the *f* randomly selected cells among m_{ae} empty cells is 1 out of C^{mae}_{f}, where C^{g}_{h} represents the number of ways of picking *h* unordered outcomes from *g* possibilities. Since the master site does not know the $v_d s$, the probability of guessing the inserted $v_d s$ for one item is 1 out of 2. Thus, the probability of guessing the randomly selected empty cells and their ratings is 1 out of $(100 * m_{A*}(1/2)^f * C^{mae}_{f})$.

3.7 Experiments

To evaluate the overall performance of the new schemes, experiments are performed using two well-known data sets, Jester, and EachMovie (EM). The DEC Systems Research Center collected EM. It contains ratings of 72,916 users for 1,628 movies. User ratings are recorded on a numeric six-point scale, ranging from 0 to 1. CA and F1 are employed to measure accuracy. Coverage is also used as a metrical indicator to show the effectiveness of the NBC-based CF algorithm with combining various amounts of data. A basic measure of coverage is the percentage of items for which predictions are available [22]. Low number of users and neighbors results in low coverage.

Firstly, the numerical ratings are transformed into binary ones because NBC employs binary ratings rather than numerical ones. For EM data set, items are labeled as 1 if the numerical rating for the item is bigger than 0.5 or 0 otherwise in EM. For Jester data set, items are labeled as 1 if the numerical rating for the item is above 2.0 or 0 otherwise in Jester. For train and test sets, 3,000 and 2,000 users are selected randomly, respectively, among those users who have rated at least 50 and 60 items from Jester and EM, respectively. 5 rated items are randomly selected from test users' ratings vectors as test items. The number of users and/or items to be selected varies for various experiment sets. CF tasks are performed using the training sets to provide referrals to test users for test items. The selected rated items' votes are withheld, their

entries are replaced with null, and tried to predict their values. Predictions are compared for them with their withheld true votes. Experiments are run for split sets alone and combined data; and found average CAs and F1s.

It is hypothesized that accuracy, privacy, and efficiency depend on various factors. Since combining partitioned data increases the available data, it is expected that this might improve accuracy while increasing computation time. Therefore, experiments are performed using the disjoint data sets alone and the integrated data. Then their outcomes are compared. Since HPD and VPD are both considered, the number of items (*m*) and users or features (*n*) are varied to show how various sizes of disjoint and integrated data sets affect the results. Moreover, since default votes (v_d*s*) are inserted randomly selected cells, trials are performed to show how different numbers of randomly selected cells (*f*) affect accuracy. Also, computation times are computed. The experiments are run using MATLAB 7.3.0 on a computer, which is Pentium 4, 3.00 GHz with 1 GB RAM. The following experiments are performed:

Due to insufficient data, CF systems, especially those newly established ones, are able to provide referrals for only a limited number of items and they might face cold start problem. It is expected to increase the coverage by integrating split data. Combining VPD makes it more likely to find reliable matching between users. However, since number of users involving in recommendation process increases, integrating HPD improves coverage. It is assumed that if there is one or more available ratings for *q*, the CF system could provide referrals for *q*. Coverage values are found for data owners on data they owned and the combined data. Since Jester is much denser than EM, for Jester, when *n* is 50, the coverage is 99.5% and 100% for split and combined data, respectively. When *n* is bigger than 100, coverage is 100% for both split and integrated data. For EM, *n* is varied from 50 to 1,250 to show how coverage changes with combining different sizes of split data and outcomes are shown as percentages in Table 3.1, where split and combined data contain *n* and 2 * *n* users' ratings, respectively. As seen from Table 3.1, coverage increases with combined data and increasing *n*. Therefore, combining HPD improves coverage and helps overcome the cold start problem.

Table 3.1 Coverage with Combined Data

n	50	125	300	750	1250
Split Data	45.76	63.14	72.72	83.53	88.08
Combined Data	74.26	85.31	87.65	91.64	96.25

Experiments performed with varying *n* values to show how combining different amounts of HPD affect accuracy and recommendation computation time (CT) in seconds. It is more likely to find large enough neighborhoods for more accurate and reliable referrals by combining HPD. Training users are randomly selected while varying *n* from 50 to 1,250 and 1,000 test users are randomly selected from train and test sets, respectively. Using the new scheme, referrals are found for randomly selected 5 rated items from each test user's ratings vector based on disjoint data sets alone and combined data. Then predictions are compared with true ratings, the average outcomes are calculated and they are displayed in Table 3.2, where combined data contains 2 * *n* users' data.

Table 3.2 Overall Performance with Combining Varying Amounts of HPD

		Jester					EM				
n		50	125	300	750	1,250	50	125	300	750	1,250
Split Data	CA (%)	64.86	66.55	67.37	68.07	69.73	70.96	72.95	74.29	74.88	75.14
	F1 (%)	63.42	64.77	65.81	66.40	66.64	78.04	79.77	80.85	81.23	81.46
	CT (secs)	15	35	104	345	706	48	127	315	909	1,302
Combined Data	CA (%)	66.14	67.22	69.16	70.15	71.40	73.12	74.62	75.28	75.50	75.86
	F1 (%)	64.50	65.76	66.08	67.57	68.12	79.74	81.02	81.56	81.69	81.79
	CT (secs)	21	82	277	926	1,930	83	224	582	1,680	2,986

As seen from Table 3.2, the accuracy of the referrals becomes better both with combined data and increasing *n* values. Although accuracy is improved by combining HPD, as expected, time to provide recommendations increases. CTs represent the times to produce 5,000 referrals based on various amounts of data. Therefore, combining HPD improves accuracy while sacrificing on time.

When VPD is combined, the number of available items increases. It helps find more reliable matching between users and it is expected improvements in referral qualities, while expecting an increase in CTs because more comparisons are done due to increasing data. To show how overall performance changes with integrating varying amounts of VPD, experiments are conducted while varying *m*. Since Jester has only limited number of items, only EM is employed in these experiments. For this purpose, 1,000 training users are randomly selected and the same 1,000 test users used. Referrals are computed for 5 test items randomly selected among the rated items of test users' ratings vectors. After finding referrals using the new scheme with varying *m* values, they are compared with true ratings, and the CAs and the F1s calculated. The CTs are also computed and the results are shown in Table 3.3.

Table 3.3 Overall Performance with Combining Varying Amounts of VPD

	Split Data					Combined Data				
m	200	350	500	650	814	400	700	100	1300	1628
CA (%)	63.27	65.12	66.16	67.16	67.33	65.96	67.52	68.04	70.94	71.26
F1 (%)	75.99	77.23	78.09	78.91	79.57	78.48	78.84	78.87	79.98	80.84
CT (secs)	218	452	582	667	811	561	655	896	1,093	1,260

By combining VPD, it is more likely to find reliable matchings between users and have sufficient commonly rated items. That is why, as seen from Table 3.3, accuracy improves with both combining VPD and increasing *m* values. However, as expected, CTs increase due to the same reasons explained previously. More importantly, recommendations are calculated based on integrated data are more

reliable than the ones computed on disjoint data sets alone because reliable matching can be found between users.

To prevent data holders from deriving data by acting as an *a* in multiple scenarios, the party that needs to send the conditional probabilities to the master site, insert v_d*s* into randomly selected empty cells of *a*'s ratings vector. Although v_d*s* are non-personalized ratings, inserting them into empty cells affects accuracy. To show how different *f* values affect the quality of the referrals, experiments are performed using both data sets, where 1,000 train users and the same 1,000 test users are used. x_f is defined as a percentage of empty cells to be filled and x_f is varied from 0 to 100. Data disguising is run 10 times and CA, F1 and CT values are computed. Since the results are similar, only F1 and CT values are displayed in Table 3.4.

Table 3.4 Overall Performance with Varying *f* Values

	Jester				EM			
x_f (%)	0	30	60	100	0	30	60	100
F1 (%)	67.76	63.36	60.96	58.19	81.53	81.01	80.70	80.57
CT (secs)	603	645	656	665	1,118	1,269	1,446	1,623

As seen from Table 3.4, inserting v_d*s* into randomly selected cells affects accuracy and the times required to provide recommendations. Although accuracy worsens by inserting v_d*s*, the results are still promising even if x_f is 100, where all empty cells are filled with non-personalized ratings. With increasing x_f, accuracy becomes worse and CTs increase. On the other hand, data owners protect their privacy by adding randomness to the private data. Data holders can adjust x_f to achieve required levels of privacy, accuracy, and efficiency.

3.8 Conclusions

Partitioned data-based CF with privacy is receiving increasing attention lately. NBC is one of the most successful algorithms on many classification domains and

widely used in CF. It is shown that it is still possible to provide accurate recommendations efficiently based on partitioned data between online vendors, even competing companies, without greatly jeopardizing their privacy. The new schemes are evaluated in terms of accuracy and computation costs by conducting experiments based on well-known real data sets. The experiment results show that the outcomes are promising and the proposed schemes allow online vendors to provide accurate referrals efficiently on partitioned data. The schemes are analyzed in terms of privacy and it is shown that they are secure.

4. NBC-BASED COLLABORATIVE FILTERING USING CLUSTERING WITH PRIVACY

In order to be successful, CF systems, which are widely utilized by many online vendors, are expected to provide accurate recommendations efficiently without deeply violating users' privacy. Customers prefer those sites that offer accurate predictions efficiently while preserving their privacy. However, with increasing numbers of users accessing the Internet and products available online, it becomes difficult to offer referrals to loads of users in a limited time. Providing predictions with decent accuracy is another challenge. Moreover, many CF systems fail to protect users' privacy. Therefore, it becomes a demanding goal to provide accurate referrals efficiently with privacy.

In this chapter, how to improve NBC-based CF systems' performance using clustering is studied when binary ratings are utilized to offer predictions. RRT is proposed to protect users' privacy while still providing accurate referrals. Various experiments are performed on real data to show how accurate predictions are, how much online recommendation computation times are improved, and how much accuracy worsens due to privacy concerns. Finally, the outcomes are demonstrated and suggestions are provided. The results show that the proposed schemes improve performance and allow producing accurate predictions even with privacy concerns.

4.1 Introduction

E-commerce is increasingly becoming popular. Many people trade over the Internet. Numbers of users accessing the Internet and items available online are rapidly escalating. Shoppers want to buy products that they might like without wasting too much time, while online vendors desire to keep current customers and recruit new ones. Several methods have been employed to achieve such goals. To serve users better, information filtering and recommendation schemes become imperative. CF techniques are among such schemes and widely used by many companies to offer predictions using other users' data.

CF systems collect users' preferences about products they bought or showed interest. They then produce referrals based on such collected ratings by matching together users who share the same tastes. With the help of CF, users can get recommendations about movies, books, news, music CDs, restaurants, bars, and other categories. To perform CF, ratings from users for items are collected. Such ratings can be numerical or binary; and they can be collected explicitly or implicitly.

Ratings collected for CF are generally numeric; however, in some cases, CF systems collect binary ratings for their applications. For example, for market basket data analysis and document clustering, binary ratings are collected. Users' preferences can be collected in binary ratings showing whether they like an item or not rather than showing how much they like or dislike a product. When binary ratings are available and collecting ratings as binary is inevitable, to perform CF services efficiently, the most appropriate algorithms should be utilized.

To generate high quality predictions, various CF algorithms have been proposed. Such algorithms can be categorized as memory- or model-based [5]. Although there are many CF algorithms, there is no perfect algorithm. Each algorithm has its own advantages and disadvantages. Memory-based algorithms achieve higher accuracy, while online time is not convincing. On the other hand, model-based approaches achieve better online performance; however, accuracy diminishes. With increasing number of users accessing the Internet and products available online, CF systems fail to offer accurate referrals efficiently. CF systems should be able to generate accurate referrals efficiently. Otherwise, it makes no sense to use such algorithms for recommendation purposes.

To increase the performance of CF systems, various approaches have been suggested. Clustering is one of such approaches and applied to CF. Using clustering, data is grouped into several clusters; and predictions then can be independently computed in each group. Since predictions are generated from each cluster independently and data in each cluster is a subset of the entire data, online computation time necessary to offer referrals might significantly degrade. In order to produce predictions from binary ratings, Miyahara and Pazzani [40] propose to

employ NBC, which is one of the most successful algorithms in many classification domains. It is simple and it is shown to be competitive with other approaches.

Today's CF systems have various problems. The first one is generating loads of recommendations to many users during an online interaction. The second problem is providing referrals with decent accuracy. Finally, the last problem is producing truthful predictions while protecting users' privacy. To get the most appropriate items, customers ask predictions before they decide to choose a product to buy. Generating such referrals to users is vital for both customers and e-commerce sites. Recommendations provided to customers should be accurate and dependable. Otherwise, inaccurate and untrustworthy predictions lead angry customers who might decide to buy products through other online vendors. To keep the current customers and recruit the new ones, it is vital to offer referrals with decent accuracy. It is an easy task to offer predictions to users if the numbers of users and items are small. However, it becomes tiresome to produce such referrals with increasing numbers of users and items. Without privacy protection measures, CF systems are serious threat to privacy. They pose several privacy risks [14]. Due to privacy concerns, users might decide to give false data or refuse to contribute data at all. It then becomes a problem to provide predictions on false and/or insufficient data. If their privacy is protected, users might feel more comfortable to give their true data. Therefore, protecting users' privacy is vital for the success of CF systems.

It is possible to take advantages of memory- and model-based CF schemes. While model-based ones offer better performance since model generation is done off-line, accuracy is worse. Memory-based ones achieve better accuracy; however, their online performances degrade with increasing available data. The goal is to conduct some computations off-line like in model-based schemes to achieve better online performance and employ memory-based approaches to improve accuracy.

According to the survey conducted by Cranor et al. [15], great majority of people have concerns about their privacy. Since CF systems collect data from many customers and they are a serious threat to privacy, customers do not feel comfortable to disclose their private data. They might send false data or refuse to give data at all.

Without introducing privacy-preserving measures, it is difficult to convince users about giving their true preferences about items. The outcomes generated from false and/or not enough data then are most likely to be untrustworthy and inaccurate. It is hypothesized that if privacy measures are provided, customers might give more truthful data; and that might improve accuracy. In order to protect users' privacy, RRT is proposed to use. Such techniques can be utilized to perturb users' binary preferences. Using RRT, the private data is masked in such a way that certain computations can be done without jeopardizing users' privacy. It can be still possible to estimate aggregate information with decent accuracy from data disguised by using RRT if there are enough data available. Since CF is based on aggregate computations, CF services can be performed on perturbed data.

To improve the performance of CF systems, various approaches have been employed. Clustering is one of such approaches and applied to CF. Ungar and Foster [65] present clustering methods for CF, where they group people into clusters based on the items they have bought. Instead of partitioning users into clusters, the set of items are partitioned into clusters based on user rating data [42]. Predictions then can be independently computed in each group. Lin et al. [38] generate clusters from training data and such clusters form the basis for similar user selection. In [24], Hu et al. propose an approach, which is a hybrid model of user and item-based CF. By clustering data using *k*-means, the authors want to improve the performance of CF systems. Also, using item-based CF algorithm in their method, they smooth sparse data. Srinivasa and Medasani [63] propose an approach, which is active in that it can rapidly adopt user interest changes. They present fuzzy clustering approach, where they are able to clustering at document content level, user group level, and document clustering. In the proposed approach, only users are clustered based on cluster membership values and real data-based testing results are presented to show how effective the schemes are. Rashid et al. [53] propose a new algorithm consisting of both memory- and model-based CF algorithms. They build a model offline using *k*-means clustering algorithm on user preference data. They then provide predictions online using a simple nearest-neighbor approach. In addition to employing clustering

methods to CF, as mentioned previously, other methods like SVD and Eigentaste have been also employed for CF to improve performance. Although, clustering methods especially *k*-means clustering is applied to CF for improving its performance, these clustering methodologies are applied for numerical ratings. As mentioned above, data in CF systems can be in binary form and the applied clustering algorithms are not suitable for binary data.

The proposed work here differs from the aforementioned works: Firstly, it is investigated how to improve NBC-based CF algorithms using clustering, where binary ratings are used for predictions. Then, clustering users based on their disguised data is scrutinized, where RRT is used for data masking. Finally, how to provide referrals based on masked data in each cluster independently is studied. Real data-based experiments are performed, their outcomes are analyzed and shown, and suggestions are presented.

In this chapter, how to improve NBC-based CF on binary ratings using clustering methods are scrutinized. The proposed schemes should enhance online computation time and accuracy. It is also investigated whether it is still possible or not to offer predictions using the improved schemes while preserving users' privacy. Since there is a trade-off among accuracy, privacy, and performance, solutions, which are able to find a good balance between them, are offered.

4.2 NBC-based Collaborative Filtering with Clustering

With increasing number of users and items, NBC-based CF systems' online performance degrades. Since memory- and model-based algorithms have their own advantages, an approach, which can leverage the advantages of both kinds of algorithms, is proposed to provide accurate recommendations efficiently using NBC-based CF systems. For this purpose, firstly, clustering is employed to users' data; and then predictions are provided based on the data in each cluster independently using NBC-based CF algorithms.

Clustering selects subsets of users. This selection helps CF systems choose the most similar users and put them into the same clusters. Since predictions are

calculated based on the data in each cluster alone, online performance improves. In this study, it is proposed to use clustering to improve the efficiency of NBC-based CF schemes. Although there are various clustering algorithms, *k*-means is one of the most popular algorithms. It is one of the widely used algorithms to cluster numeric data. However, it is not suitable for binary ratings. Therefore, *k*-modes (KM) clustering algorithm, which is a variant of *k*-means algorithm to cluster categorical data, is proposed to use. Clustering algorithms usually place each object in a single cluster. However, in some cases, an object can belong to more than one cluster or it might improve accuracy to place an object into more than one cluster. For this purpose, the idea of fuzzy clustering is utilized. Unlike other clustering algorithms, fuzzy clustering algorithms return cluster membership values rather than clustering objects. Fuzzy C-means and fuzzy C-modes algorithms are used to cluster numeric and categorical data [3, 29]. In this study, how to improve NBC-based CF using KM clustering algorithm is studied. Moreover, the idea of fuzzy clustering is applied to KM to be able to put users into more than one cluster. Then, how to provide predictions based on such algorithms without violating users' privacy is investigated. The computations contain online and off-line phases. Clustering is done off-line while predictions are provided online.

The KM algorithm takes the input parameter *k* and partitions a set of *n* objects into *k* clusters. Cluster center is measured in regard to the mode value of the objects in the cluster [22]. As explained previously, data collected for CF purposes might be binary. In such cases, those algorithms suitable for categorical data should be utilized. To find the similarities between objects for KM clustering, it is needed to use a similarity measure to calculate the likeness between two users represented with binary ratings. For this purpose, the variant of Tanimoto coefficient [50] is proposed to use, as follows:

$$w_{au} = (t(y_s) - t(y_d)) / t(y), \quad (4.1)$$

where $t(y_s)$ and $t(y_d)$ represent the number of similarly and dissimilarly rated items by users *u* and *a*, respectively, $t(y)$ is the number of commonly rated items by them, and w_{au} shows the similarity between users *u* and *a*. Similarities range from -1 to 1. If w_{au}

> 0, users *a* and *u* are similar; otherwise, they are dissimilar. When w_{au} is 0, they are not correlated at all.

4.2.1 Providing Recommendations

During online recommendation generation, an *a* should be first assigned to a cluster. For selecting *a*'s cluster, similarities between *a* and clusters' centers are calculated similarly. *a* then is assigned to the closest cluster. Note that cluster centers are also represented with binary values. The following methods are proposed to use to generate predictions:

Basic *k*-Modes (BKM). In this method, users are first grouped into clusters using KM. Each user can belong to at most one cluster. After placing users in clusters, the most similar cluster to *a* is determined. Finally, CF services for *a* are performed based on the data in that cluster only.

Extended *k*-Modes (EKM). The key idea behind CF is that *a* will prefer those items that like-minded users prefer, or that dissimilar users do not [46]. Therefore, it might improve accuracy to perform CF services for a based on those users' data who are the most similar and dissimilar users to *a*. For this purpose, after finding the most similar or the closest cluster to *a*, also the most dissimilar or the furthest cluster to *a* is found. We then compute predictions based on the data from these two clusters. Note that one of the clusters contain the most similar users to *a*, while the other includes the most dissimilar users to *a*. The predictions then are determined based on the most similar and dissimilar users' data.

Fuzzy *k*-Modes (FKM). Conventional clustering algorithms place each object into a single cluster. Unlike other algorithms, fuzzy clustering returns cluster membership values, rather than putting users in clusters. Users then can be clustered based on such values, where one user can belong to more than one cluster. When each user belongs to a single cluster, useful information might be losed. In some cases, some users might belong to more than one cluster. Therefore, fuzzy clustering to KM is applied, as follows: Similarities (w_{uk}) between a user *u* and each cluster *k* based on binary ratings firstly computed, as explained previously. The bigger the w_{uk} is the

closer the user *u* to cluster *k*. After finding such similarities (or distances), users are placed into those clusters whose similarities are bigger than or equal to a predefined threshold (τ). Note that it is critical to select the optimum τ, which can be determined experimentally. When a lower value is selected, it is more likely to put dissimilar users into the same clusters. That can make both accuracy and online performance worse because the number of users in one cluster increases and clusters may include unlike users. If the threshold value is set too high, useful information might be losed.

4.2.2 Evaluating NBC-based CF Schemes with Clustering

The proposed schemes should be able to provide accurate predictions efficiently. Accuracy can be defined, as follows: Recommendations produced based on the proposed schemes should be as close as the true withheld ratings. More formally, the proposed methods should achieve higher CA and F1 values. The higher the CA and F1, the better the schemes are. Efficiency or online performance represents the online computation time required to produce recommendations.

To evaluate how clustering affects the overall performance of NBC-based CF, experiments are performed on Jester and EM data sets. Although there are other data sets available for CF, the results based on these two sets can be generalized. Compared to EM, Jester is denser and almost 50% of all ratings are available. To measure the accuracy of our schemes, CA and F1 are employed.

Higher CA and F1 indicate better recommendations. The higher the CA and F1, the better the results are. Besides evaluating the schemes in terms of accuracy, it is also wanted to assess them in terms of online time to provide predictions. For this purpose, *T* is defined in seconds as online time required offering recommendations. The smaller the *T*, the better the schemes are.

Firstly, numerical ratings are transformed into binary. Using the similar methodology in [40], items are labeled as 1, if the numerical rating for the item is bigger than 0.5, or 0 otherwise in EM, while they are labeled as 1, if the numerical rating for the item is above 2.0, or 0 otherwise in Jester. Users who rated more than 60 items are selected from both data sets. Each data set is randomly divided into two

disjoint sets, train and test. For each experiment, the required number of train and test users are randomly selected from train and test sets, respectively, based on the experiment requirements. For each test user, 5 rated items are randomly picked, replaced their entries with null, and tried to predict their votes. Predicted votes are compared with true withheld ratings. After computing CA and F1, the final overall outcomes are demonstrated. *T* is also calculated for each set of experiments and the results are shown. The experiments are run using MATLAB 7.3.0 on a computer, which is Intel Core2Duo, 2.2 GHz with 2 GB RAM.

Firstly, experiments are performed using both data sets to show how BKM affects the results with varying numbers of clusters. *k* is varied from 1 to 15, where 1,000 and 500 train and test users are used, respectively. In these experiments, each user belongs to a single cluster. For each test user or *a*, 5 recommendations are produced for withheld items based on the data in the closest cluster to *a*. *T*, CA, and F1 values are calculated. Since the results are similar, CA and *T* values only are shown in Table 4.1. Note that *k* is 1 means that all users are grouped into one cluster or there is no clustering.

Table 4.1 Effects of BKM with Varying *k*

	k	1	2	3	5	7	10	13	15
Jester	CA (%)	67.80	68.36	68.72	68.56	68.48	68.96	68.04	68.36
	T (secs)	99	25	21	13	10	7	7	5
EM	CA (%)	69.92	70.28	69.92	68.88	68.48	68.48	67.92	68.52
	T (secs)	250	113	73	49	37	21	19	18

As seen from Table 4.1, with increasing *k*, *T* significantly improves. With increasing *k*, number of users in each cluster becomes smaller; and that makes *T* better. Since number of items in EM is bigger than Jester, *T* values for EM are worse. For Jester, accuracy slightly improves with clustering. When *k* is 10, accuracy improves by 1.16%. For EM, accuracy improves when *k* is 2 only. Although accuracy

slightly becomes worse when *k* is bigger than 2, accuracy losses are small. In the worst case, accuracy becomes 1.44% worse. However, in the same case, *T* becomes better by almost 14 times. When *k* is 2, *T* gets better by 2.21 times. Due to the sparseness of EM, accuracy losses due to clustering could be expected. With decreasing number of users, it becomes a challenge to have large enough commonly rated items between users. To sum up, however, the improvements in *T* are significant and they might outweigh the losses in accuracy in sparse data. For dense sets, both accuracy and *T* improve with clustering.

To evaluate how overall performance changes with EKM method, experiments are performed using both data sets. The same 1,000 and 500 train and test users are used, respectively. *k* is varied from 1 to 15. Referrals are produced for 5 withheld items for each test user. After computing *T*, CA, and F1 values, the outcomes are demonstrated in Table 4.2. Since the outcomes are similar, F1 and *T* values only are shown.

Table 4.2 Effects of EKM with Varying *k*

	k	1	3	5	7	10	13	15
Jester	F1 (%)	66.64	67.60	68.74	68.93	68.02	68.70	68.37
	T (secs)	99	53	32	23	18	15	12
EM	F1 (%)	70.02	69.92	69.25	68.58	69.37	68.22	68.40
	T (secs)	250	166	81	75	48	41	40

As seen from Table 4.2, EKM improves both accuracy and efficiency for Jester. For EM, although efficiency gets better with increasing *k*, accuracy slightly degrades. However, accuracy losses are negligible compared to the gains in *T*. When the results on EKM with the results on BKM are compared, for Jester, EKM achieves better results in terms of CA and F1 values. As expected, *T* values become worse in EKM due to the increasing number of users. Remember that it is considered both the most similar and dissimilar users in EKM. For EM data set, CA and F1 values on

EKM get slightly worse. This phenomenon could be explained with the sparseness of EM. However, since referrals are generated based on both the most similar and dissimilar users' data, such predictions might be more dependable and trustworthy. In terms of *T*, CA, and F1 values, the optimum *k* values for Jester and EM are 13 and 5 are concluded, respectively. However, CF systems are able to determine *k* values according to their preferences over accuracy and efficiency.

To show how FKM affects the results, trials are conducted using both data sets. The same 1,000 and 500 train and test users are again used, respectively, where predictions are sought for randomly selected 5 rated items for each test user. In order to determine the optimum value of τ and how accuracy changes with different τ, values of τ is varied from 0.30 to 0.75. Although experiments are performed while changing *k* from 1 to 15, the outcomes are demonstrated for *k* being 13 and 5 for Jester and EM, respectively. After calculating CA and F1 values, the results are displayed in Table 4.3.

Table 4.3 Effects of FKM with Varying τ

	τ	0.30	0.50	0.65	0.75
Jester	CA (%)	68.84	68.12	68.28	67.68
	F1 (%)	67.73	67.16	67.21	66.51
EM	CA (%)	69.40	68.00	68.40	68.16
	F1 (%)	68.42	67.40	67.96	67.26

For Jester, remember that when *k* is 1 or no clustering, CA and F1 values are 67.80 and 66.64, respectively. For EM, they are 69.92 and 70.02, respectively. As seen from Table 4.3, the results for various τ values are better than the results for *k* being 1 for Jester. In terms of accuracy, the outcomes are the best when τ is 0.30 for Jester. However, with decreasing τ values, since each cluster is more likely to contain more users, online computation times are expected to increase. With increasing τ values, *T* is expected to improve. Therefore, in terms of overall performance, 0.65 can

be selected as the optimum value of τ for Jester. For EM, due to its sparseness, the outcomes are slightly worse compared to the base results. For EM, as for Jester, 0.65 can be chosen as the optimum value of τ because it happens to give the best results in terms of accuracy and efficiency.

To sum up, clustering significantly improves efficiency. With clustering, it becomes easier to provide loads of referrals to many users in a limited time during an online interaction. Since the more clusters have, the less users each cluster includes, online performance improves with increasing number of clusters. For dense data sets, clustering makes accuracy better. For sparse data sets, however, clustering slightly degrades accuracy. The gains in online performance due to clustering compensate the losses in accuracy.

4.3 Privacy-Preserving NBC-based CF with Clustering

Privacy has been increasingly receiving attention. Although it is not easy to define privacy succinctly, privacy can be defined in this context, as follows: CF systems should not be able to learn the true values of users' ratings. Moreover, it sometimes might be more dangerous for people to disclose that they rated or bought certain items. Therefore, besides preventing CF systems learning true rating values, CF systems should not be allowed to learn the rated and/or unrated items by each user.

With the evolution of the Internet and e-commerce, collecting customers' private data becomes easier. Due to privacy concerns, many users do not want to reveal their data. Today's CF systems are advantageous; however, they are serious threat to individual privacy. They pose several privacy risks such as unsolicited marketing, price discrimination, being subject to government surveillance, users' profiles might be used in a criminal case, and so on [14]. Customer data is considered valuable and can be transferred. Due to such risks, users do not feel comfortable to disclose their preferences. They sometimes refuse to provide data at all or might give false data. Recommendations then on such insufficient and false are more likely to be

inaccurate and untrustworthy. If users' privacy is protected, they might feel more comfortable to give their data and it becomes easy to collect more truthful data.

Users' privacy is protected while still providing accurate predictions using clustering-based CF systems. Users disguise their ratings before sending them to a server. The data perturbation techniques should be able to prevent the server from learning true ratings and rated items. Moreover, they should be able to allow providing accurate referrals efficiently. RRT is proposed to use to achieve privacy. As stated previously, it sometimes might be more dangerous for people to disclose that they rated or bought certain items. Therefore, they might to hide their rated items besides hiding true ratings. To prevent the server from learning rated items, users can fill some of their empty cells with fake ratings or default votes. As investigated by [28], users can fill empty cells in such a way to achieve required levels of accuracy and privacy. RRT makes it possible to estimate aggregate data items with decent accuracy. Although we cannot do anything with individual user's masked data, it is still possible to estimate aggregate data. Since clustering and NBC-based CF are based on aggregate data, meaningful outcomes can be still generated from perturbed data. When there are enough users and/or items, the contribution of faked or default votes to similarity and prediction computations will be close to zero. Therefore, NBC-based CF can be combined with RRT to provide predictions with privacy.

Multi-group schemes [28] are utilized to mask private data. After data clustering, similar schemes employed by [28] can be utilized to produce recommendations using NBC-based CF algorithm. However, data clustering can be accomplished based on disguised data. Since users disguise their rating vectors by dividing them into multiple groups (let say M groups, M-group scheme), the server must be consider all possibilities to estimate the similarities between disguised vectors because it does not know if the received data items are true or false. Since similarities are calculated between two masked vectors, the server must consider all 2^{2M} possibilities when ratings are grouped into M groups. Therefore, to find similarities, Eq. (4.1) should be modified in such a way to estimate similarities

between disguised vectors. The server can estimate such similarities by modifying Eq. (4.1), as follows:

$$w'_{ij} = \sum_{z=1}^{2^{2M}} w'_{ijz} * p_z \tag{4.2}$$

where w'_{ij} is the estimated similarity between masked ratings vectors i and j, M shows number of groups, p_z represents the probability of occurrence of z^{th} possibility, and w'_{ij} is the estimated similarity between masked ratings vectors i and j in the case of the z^{th} possibility. For example, if users disguise their data by using 2-group scheme, the similarity between two disguised vectors can be estimated, as follows, by considering all 2^4 possibilities:

$$w'_{ij} = \sum_{z=1}^{16} w'_{ijz} * p_z = w'_{ij1} * p_1 + w'_{ij2} * p_2 + \ldots + w'_{ij16} * p_{16} \tag{4.3}$$

In addition to protecting users' privacy, the proposed schemes should preserve active users' privacy, as well. As explained in Section 2. 4. 4, *a*'s privacy can be achieved using the 1-out-of-*n* Oblivious Transfer protocol.

4.3.1 Evaluating Privacy-Preserving NBC-based CF with Clustering

The proposed privacy-preserving schemes should allow CF systems to offer accurate referrals while preserving privacy. Furthermore, they should not introduce significant overhead costs due to privacy concerns. As expected, privacy protection measures make accuracy worse because accuracy and privacy are conflicting goals. However, accuracy losses due to privacy concerns should be acceptable.

To show how accuracy changes due to privacy protection measures, trials are performed using both data sets. 1,000 and 500 users for training and testing are used, respectively. 5 test items are again used for each test user. Data disguising is done 100 times. In other words, experiments are conducted 100 times while each time disguising data independently. After computing CA and F1 values, the overall averages are displayed. Since how M and θ affect accuracy is already discussed in Section 2.6, M is set at 3 and θ at 0.60 for the experiments. With increasing M and θ

values, randomness increases while accuracy diminishes. Moreover, privacy improves due to increasing randomness. To hide rated/unrated items, randomly selected 50% of empty cells were filled with fake ratings. EKM is employed in these experiments, where *k* is set at 13 and 5 for Jester and EM, respectively. The outcomes are displayed in Table 4.4.

As seen from Table 4.4 and expected, accuracy degrades with privacy concerns for both data sets. On average, accuracy decreases by 4% in terms of CA and F1 for both data sets. As generally known, privacy and accuracy are conflicting goals. Therefore, it is expected that accuracy becomes worse due to privacy protection measures. It is vital, however, that such losses should not be significant. The results show that it is still possible to offer predictions with decent accuracy without greatly violating users' privacy. Moreover, users and CF systems can determine the values of M and θ, and decide how many empty cells to disguise in such a way to achieve required levels of privacy and accuracy.

Table 4.4 EKM with Privacy

	τ	Without Privacy	With Privacy
Jester	CA (%)	69.44	65.34
	F1 (%)	68.70	64.05
EM	CA (%)	69.24	65.01
	F1 (%)	69.25	65.83

The proposed schemes do not introduce additional storage and communication costs in terms of number of communications. However, since users fill some of their empty cells, amount of data transferred increases due to privacy concerns. As explained previously, the schemes include both off-line and online computation costs. Note that clustering is done off-line. Without privacy concerns, online computation times improve because recommendations calculated on data in each cluster independently and entire data is grouped into clusters. Unlike online

costs, off-line costs are not critical for overall performance. Although online computation costs decrease due to clustering, it is expected that online costs increase due to privacy concerns. Remember that users perturb their data by dividing them into M groups and disguising each group independently. The server should consider all possibilities to offer predictions because it does not know whether the received data is true or false. Moreover, since a sends Y ratings vectors including her true ratings vector, the system should compute predictions based on all these vectors.

Privacy analysis can be similarly done as done in Section 2.5. Due to randomly inserted fake or default faults, the server will not be able to learn rated items. Since the server does not know the randomly generated r_u values, it does not know whether the received data is true or false. With increasing M values and increasing θ values towards 0.5, privacy improves as expected due to increasing randomness. However, accuracy diminishes. The users are able to select M and θ values to achieve required levels of privacy and accuracy.

4.4 Conclusions

Using clustering is proposed to improve the overall performance of NBC-based CF systems. With increasing available data, it becomes tiresome to generate loads of recommendations to many users. Clustering partitions data into subsets and predictions could be produced from data in each subset independently. Since NBC-based CF is based on binary ratings, k-modes clustering algorithm is proposed to utilize clustering to group binary data. In order to achieve better accuracy, fuzzy clustering is also tried to apply. To evaluate the overall performance of the schemes, real data-based experiments are performed. Experiment results show that clustering significantly improves online computation times. In addition, it increases accuracy for dense data sets. For sparse data sets, accuracy slightly diminishes with clustering. However, the gains in efficiency outweigh the losses in accuracy.

Besides accuracy and efficiency, privacy protection is another demanding goal of CF systems. We propose schemes to achieve NBC-based CF with clustering while preserving users' privacy including active users. Privacy, accuracy, and

performance are conflicting goals. Due to privacy concerns, we expect that accuracy and efficiency degrade. However, losses due to privacy-protection measures should be acceptable. The experiment results show that accuracy losses due to privacy protection are not significant. Although off-line additional costs increase due to privacy, they are not critical for overall performance. In order to achieve a good balance between privacy, accuracy, and performance, users and CF systems are able to determine the parameters of privacy protection measures.

5. CONCLUSIONS AND FUTURE WORK

In this thesis, approaches are proposed to overcome the challenges for NBC-based CF algorithm. The proposed schemes are analyzed in terms privacy, accuracy, and efficiency; and they are encouraged with real data-based experiments.

The experiments results show that it is possible to produce private recommendations using RRT with NBC. The proposed schema makes it possible for servers to collect private data without greatly compromising users' privacy. Experiments results show that the schemes allow providing referrals with decent accuracy. To obtain a balance between accuracy, privacy, and efficiency, the parameters of the schemes can be adjusted. According to experiments results, the proposed approach parameters have different effects on privacy, accuracy, and efficiency. If θ values increase from 0 to 0.5, privacy level increases and while accuracy decreases. If θ continues to increase from 0.5 to 1, privacy level decreases while accuracy increases. These results show that privacy and accuracy are conflicting goals. In addition, the group number parameter M has effect on privacy and accuracy. If M increases from 1 to 5, privacy level increases; however, accuracy decreases. For producing private and accurate predictions, θ and M must be adjusted.

Partitioned data-based CF with privacy is receiving increasing attention lately. It is shown that it is still possible to provide accurate recommendations efficiently based on partitioned data between online vendors, even competing companies, without greatly jeopardizing their privacy. Solutions are proposed to produce private referrals efficiently based on partitioned data using NBC. The experiments results show that the outcomes are promising and the proposed schemes allow online vendors to provide accurate referrals efficiently on distributed data without revealing their private data. The methods are analyzed in terms of privacy and the analysis shows that they are secure. The effects of integrating distributed data and privacy concerns on accuracy are scrutinized based on real data-based trials. Moreover, the solutions allow data holders to produce referrals efficiently.

Evolution of CF systems increases and the results of this evolution increase the runtime of CF algorithms. Lots of algorithms have been proposed to overcome

this challenge. In the fourth chapter, an approach is proposed to improve runtime of NBC and also privacy techniques are applied to achieve efficient CF systems with privacy. It is shown that online performance of NBC-based CF can be improved by using *k*-modes clustering. *k* has a vital effect on accuracy and computation time. Data owners must choose the optimum *k* for their data sets. With the optimum value of *k*, more accurate and efficient predictions can be generated. Experiment results show that clustering significantly improves online computation times. In addition, it increases accuracy for dense data sets. For sparse data sets, accuracy slightly diminishes with clustering. However, the gains in efficiency outweigh the losses in accuracy. The proposed schemes are evaluated based on experiments results. The outcomes show that it is possible to offer NBC-based CF services efficiently with decent accuracy using clustering methods. Users' privacy is preserved by using RRT. Accuracy losses due to privacy concerns are negligible.

Due to various privacy concerns, users might decide to hide their data variably. They can mask their private using different θ values and group schemes. If they differently perturb their data, it becomes a challenge to provide predictions from such inconsistently masked data. In the future, it will be studied whether it is still possible or not to provide accurate predictions, if users disguise their data variably. If users reveal some aggregate data whose disclosure does not violate their privacy, accuracy might improve. It should be deeply investigated whether such data closures are possible and they improve accuracy.

The proposed schemes for both HPD and VPD can be easily extended to multi-party schemes. With increasing number of parties involving CF process, computation and communication costs are expected to increase. Although combining distributed data makes accuracy better, additional costs due to privacy concerns should be scrutinized deeply. Therefore, there still remains work to be done about multi-party schemes.

There are remains works to be done in order to show why clustering makes accuracy slightly worse for sparse data sets. Although it is explained it due to sparsity, detail investigations should be performed. To improve the overall

performance of privacy-preserving schemes, aggregate data disclosures might be employed. How to utilize aggregate data disclosures to improve accuracy and efficiency will be scrutinized. To cluster binary data, other clustering algorithms could be used. It will be investigated whether overall performance can be improved or not by using another memory-based CF algorithms after clustering.

REFERENCES

[1] Basu, C., Hirsh, H., and Cohen, W. W., "Recommendation as classification: Using social and content-based information in recommendation", *Proceedings of the Recommender System Workshop'*98, 714-720, 1998.

[2] Berkovsky, S., Eytani, Y., Busetta, P., Kuflik, T., and Ricci, F., "Collaborative filtering over distributed environment", *Proceedings of the Workshop on Decentralized, Agent Based and Social Approaches to User Modeling, in conjunction with the 10th International Conference on User Modeling*, Edinburg, UK, 33-40, 2005.

[3] Bezdek, J.C., Fuzzy mathematics in pattern classification, *PhD thesis*, Cornell University, Ithaca, NY, USA, 1973.

[4] Billsus, D. and Pazzani, M. J., "Learning collaborative information filters", *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, USA, 46-54, 1998.

[5] Breese, J. S., Heckerman, D. and Kadie, C., "Empirical analysis of predictive algorithms for collaborative filtering", *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, Madison, WI, USA, 43-52, 1998.

[6] Bunn, P. and Ostrovsky, R., "Secure two-party *k*-means clustering", *Proceedings of the 14th ACM conference on Computer and Communications Security*, Alexandria, VA, USA, 486-497, 2007.

[7] Canny, J., "Collaborative filtering with privacy via factor analysis", *Proceedings of the 25th ACM SIGIR'02*, Tampere, Finland, 238-245, 2002.

[8] Canny, J., "Collaborative filtering with privacy", *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 45-57, 2002.

[9] Chandrashekhar, H., and Bhasker, B., "Collaborative filtering based on the entropy measure", *CEC'07 and EEE'07*, Tokyo, Japan, 203-210, 2007.

[10] Chen, A. Y., and McLeod D., "Collaborative filtering for information recommendation systems", *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*, Information Science Reference, **1**, 118-124, 2006.

[11] Chen, J., and Yin, J., "A collaborative filtering recommendation algorithm based on influence sets", *Ruan Jian Xue Bao/Journal of Software*, **18**, 1685-1694, 2007.

[12] Chen, Y., and Cheng, L., "A novel collaborative filtering approach for recommending ranked items", *Expert Systems with Applications*, **34(4)**, 2396-2405, 2007.

[13] Chen, Y., and George, E. I., "A Bayesian model for collaborative filtering", *Proceedings of the 7th International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 1999.

[14] Cranor, L. F., "`I didn't buy it for myself' privacy and e-commerce personalization", *Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society*, Washington, DC, USA, 111-117, 2003.

[15] Cranor, L. F., Reagle, J., and Ackerman, M. S., *Beyond concern: Understanding net users' attitudes about online privacy*, Technical report, AT&T Labs-Research, 1999.

[16] Du, W. and Zhan, Z., "Using randomized response techniques for privacy-preserving data mining", *Proceedings of the 9th ACM SIGKDD'03*, Washington, DC, USA, 505-510, 2003.

[17] Even, S., Goldreich, O., and Lempel, A., "A randomized protocol for signing contracts", *Communications of the ACM*, **28**, 637-647, 1985.

[18] Fisher, D., Hidrum, K, Hong, J., Newman, M., Thomas, M., and Vuduc, R., "SWAMI: Framework for collaborative filtering algorithm development and evaluation", *Proceedings of the 23rd Annual International SCM SIGIR Conference on Research and Development in Information Retrieval*, Athens, Greece, 366-368, 2000.

[19] Goldberg, K., Roeder, T., Gupta, D., and Perkins, C., "Eigenstaste: A constant time collaborative filtering algorithm", *Information Retrieval*, **4(2)**, 133-151, 2001.

[20] Grcar, M., "User profiling: collaborative filtering", *Proceedings of the SIKDD 2004 at Multiconference IS*, Ljubljana, Slovenia, 2004.

[21] Gupta D., Digiovanni M., Narita H., and Goldberg K. "Jester 2.0: A new linear-time collaborative filtering algorithm applied to jokes", *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 291-292, 1999.

[22] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. T., "An algorithmic framework for performing collaborative filtering", *Proceedings of the 22nd Annual International ACM SIGIR Conference*, Berkeley, CA, USA, 230-237, 1999.

[23] Hill, W., Stead, L., Rosenstein, M., and Furnas, G., "Recommendation and evaluating choices in a virtual community of use", *Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems*, Denver, CO, USA, 194-201, 1995.

[24] Hu, R. and Lu, Y., "A Hybrid user and item-based collaborative filtering with smoothing on sparse data", *Artificial Reality and Telexistence (ICAT'06)*, China, 184-189, 2006.

[25] Hurt, A., Bauer, M., and Breytmann, B., *Collaborative filtering in a distributed environment: An agent-based approach*, Technical report, University of Applied Sciences Wurzburg, Germany, 2000.

[26] Ioandinis, A., Grama, A., and Atallah, M., "A secure protocol for computing dot-products in clustered and distributed environment", *Proceedings of the 2002 International Conference on Parallel Processing*, Canada, 379-394, 2002.

[27] Kaleli, C. and Polat, H., "Providing naïve Bayesian classifier-based private recommendations on partitioned data", *Lecture Notes in Computer Science*, **4702**, 515-522, 2007.

[28] Kaleli, C. and Polat, H., "Providing private recommendations using naïve Bayesian classifier", *Advances in Intelligent Web Mastering*, **43**, 168-173, 2007.

[29] Kam, M. N. and Huang, Z., "A fuzzy *k*-modes algorithm for clustering categorical data", *IEEE Transactions on Fuzzy Systems*, **7(4)**, 446-452, 1999.

[30] Kantarcioglu, M. and Clifton, C., "Privacy-preserving distributed mining of association rules on horizontally partitioned data", *Transactions on Knowledge and Data Engineering*, **16(9)**, 1026-1037, 2004.

[31] Kantarcioglu, M. and Clifton, C., "Privately computing a distributed k-*nn* classifier", *Proceedings of the 8th European Conference on Principle and Practice of Knowledge Discovery in Databases*, Pisa, Italy, 279-290, 2004.

[32] Kantarcioglu, M. and Vaidya, J. S., "Privacy-preserving naïve Bayes classifier for horizontally partitioned data", *Proceedings of the IEEE ICDM Workshop on PPDM*, Melbourne, FL, USA, 3-9, 2003.

[33] Kaya, S. V., Pedersen, T. B., Savas, E., and Saygin, Y., "Efficient privacy-preserving distributed clustering based on secret sharing", *Lecture Notes in Artificial Intelligence*, **4819**, 280-291, 2007.

[34] Kleinberg, J. and Sandler, M., "Using mixture models for collaborative filtering", *Proceedings of the 36th ACM Symposium on Theory of Computing*, Chicago, IL, USA, 569-578, 2004.

[35] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Rield, J. T., "GroupLens: Applying collaborative filtering to Usenet news", *Communications of the ACM*, **40(3)**, 77-87, 1997.

[36] Lekakos, G. and Giaglis, G. M., "A hybrid approach for improving predictive accuracy of collaborative filtering algorithms", *Electronic Commerce Research*, **17**, 5-40, 2007.

[37] Lemire, D. and Maclachlan, A., "Slope one predictors for online rating-based collaborative filtering", *Proceedings of the 2005 SIAM Data Mining*, Newport Beach, CA, USA, 2005.

[38] Lin, C., Xue, G., Yang, Q., Xi, W., Zeng, H., Yu, Y., and Chen, Z., "Scalable collaborative filtering using cluster based smoothing", *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, 114-121, 2005.

[39] Meruge, S. and Ghosh, J., "Privacy-preserving distributed clustering using generative models", *Proceedings of the 3rd IEEE International Conference on Data Mining*, USA, 211-218, 2003.

[40] Miyahara, K. and Pazzani, M. J., "Improvement of collaborative filtering with the simple Bayesian classifier", *Transactions of Information Processing Society of Japan*, **43(11)**, 3429-3437, 2002.

[41] Naor, M. and Pinkas, B., "Oblivious transfer and polynomial evaluation", *Proceedings of the 31st ACM Symposium on Theory of Computing*, Atlanta, GA, USA, 245-254, 1999.

[42] O'Connor, M. and Herlocker, J. L., "Clustering items for collaborative filtering", *Proceedings of SIGIR 2001 Workshop on Recommender Systems*, New Orleans, LA, USA, 2001.

[43] Oliveira, S. R. M. and Zaiane, O. R., "Achieving privacy preservation when sharing data for clustering", *Proceedings of the International Workshop on Secure Data Management in a Connected World in conjunction with VLDB*, Canada, 67-82, 2004.

[44] Ouyang, W. and Huang, Q., "Privacy-preserving association rules mining based on secure two-party computation", *Lecture Notes in Control and Information Sciences*, **344**, 969-975, 2006.

[45] Parameswaran, R. and Blough, D. M., "Privacy-preserving collaborative filtering using data obfuscation", *IEEE International Conference on Granular Computing*, Silicon Valley, USA, 380-387, 2007.

[46] Pennock, D. M., Horvitz, E., Lawrence, S., and Giles, C. L., "Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach", *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, Stanford, CA, USA, 473-480, 2000.

[47] Polat, H. and Du, W., "Achieving private recommendations using randomized response techniques", *Advances in Knowledge Discovery and Data Mining*, **3918** 637-646, 2006.

[48] Polat, H. and Du, W., "Privacy-preserving collaborative filtering" *International Journal of Electronic Commerce*, **9(4)**, 9-36, 2005.

[49] Polat, H. and Du, W., "Privacy-preserving collaborative filtering on vertically partitioned data", *Lecture Notes in Computer Science*, **3721**, 651-658, 2005.

[50] Polat, H. and Du, W., "Privacy-preserving top-*N* recommendation on horizontally partitioned data", *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, Paris, France, 725-731, 2005.

[51] Polat, H. and Du, W., "Effects of inconsistently masked data using RPT on CF with privacy", *Proceedings of the 2007 ACM Symposium on Applied Computing*, Seoul, Korea, 649-653, 2007.

[52] Popescul, A., Ungar, L. H., Pennock, D. M., and Lawrence, S., "Probabilistic models for unified collaborative and content-based recommendation in sparse environments", *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, Seattle, WA, USA, 437-444, 2001.

[53] Rashid, A. L., Lam, S. K., Karypis, G., and Riedl, J. T., "ClustKNN: A highly scalable hybrid model- & memory-based collaborative filtering algorithm", *Proceedings of the WebKDD, Web Mining and Web Usage Analysis*, Pennsylvania, USA, 2006.

[54] Resnick, P. and Varian, H. R., "Recommender systems", *Communications of the ACM*, **40(3)**, 56-58, 1997.

[55] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. T., "GroupLens: An open architecture for collaborative filtering of netnews", *Proceedings of the ACM Conference on Computer Supported Cooperative Work*, Chapel Hill, NC, USA, 175-186, 1994.

[56] Rizvi, S. J. and Haritsa, J. R., "Maintaining data privacy in association rule mining", *Proceedings of the 28th Very Large Data Bases (VLDB) Conference*, Hong Kong, China, 682-693, 2002.

[57] Sanil, A. P., Karr, A. F., Lin, X., and Peiter, J. P., "Privacy-preserving regression modeling via distributed computation", *Proceedings of the 10th International ACM SIGKDD Conference*, Seattle, WA, USA, 677-682, 2004.

[58] Sarwar, B. M., Konstan, J. A., Borches, A., Herlocker, J. L., Miller, B. N., and Riedl, J. T., "Using filtering agents to improve prediction quality in the GroupLens research collaborative filtering system", *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, Seattle, WA, USA, 345-354, 1998.

[59] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., "Analysis of recommendation algorithms for e-commerce", *Proceedings of the 2nd ACM conference on Electronic commerce*, Minnesota, USA, 158-163, 2000.

[60] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., "Application of dimensionality reduction in recommender system: A case study", *Proceedings of the ACM WebKDD 2000 Web Mining for E-commerce Workshop*, Boston, MA, USA, 682-693, 2000.

[61] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J., "Item-based collaborative filtering recommendation algorithms", *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, 285 – 295, 2001.

[62] Shardanand, U. and Maes, P., "Social information filtering: Algorithms for automating "word of mount"", *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 210-217, 1997.

[63] Srinivasa, N. and Medasani, S., "Active fuzzy clustering for collaborative filtering", *Proceedings of the 2004 IEEE International Conference on Fuzzy Systems*, Budapest, Hungary, 1697-1702, 2004.

[64] Su, X., Greiner, R., Khoshgoftaar, T. M., and Zhu, X., "Hybrid collaborative filtering algorithms using a mixture of experts", *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, USA, 645-649, 2007.

[65] Ungar, L. H., and Foster, D. P., "Clustering methods for collaborative filtering", *Proceedings of the Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence*. Menlo Park, CA, USA, 1998.

[66] Vaidya, J. and Clifton, C., "Privacy-preserving decision trees over vertically partitioned data", *Data and Applications Security XIX 2005*, USA, 139-152, 2005.

[67] Vaidya, J. S. and Clifton, C., "Privacy-preserving association rule mining in vertically partitioned data", *Proceedings of the 8th International ACM SIGKDD Conference*, Edmonton, Alberta, Canada, 639-644, 2002.

[68] Vaidya, J. S. and Clifton, C., "Privacy-preserving *k*-means clustering over vertically partitioned data", *Proceedings of the 2004 SIAM Conference on Data Mining*, Lake Buena Vista, FL, USA, 206-215, 2003.

[69] Vaidya, J. S. and Clifton, C., "Privacy-preserving naive Bayes classifier for vertically partitioned data", *Proceedings of the 9th International ACM SIGKDD Conference*, Washington, DC, USA, 206-215, 2003.

[70] Warner, S. L., "Randomized response: A survey technique for eliminating evasive answer bias", *Journal of the American Statistical Association*, **60(309)**, 63-69, 1965.

[71] Westin, A. F., *Freebies and privacy: What net users think.* Technical report Opinion Research Corporation, 1999.

[72] Yu, H., Vaidya, J., and Jiang, X., "Privacy-preserving SVM classification on vertically partitioned data", *Lecture Notes in Computer Science*, **3918**, 647-656, 2006.