PRIVACY-PRESERVING COLLABORATIVE FILTERING ON ARBITRARILY PARTITIONED DATA

İbrahim YAKUT

Ph.D. Dissertation

Computer Engineering Program

June, 2012

This dissertation is supported by grant 108E221 from TÜBİTAK.

JÜRİ VE ENSTİTÜ ONAYI

İbrahim Yakut'un **"Privacy-Preserving Collaborative Filtering on Arbitrarily Partitioned Data"** başlıklı **Bilgisayar Mühendisliği** Anabilim Dalındaki, Doktora Tezi 07.06.2012 tarihinde, aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı-Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Doç. Dr. HÜSEYİN POLAT**	**…….**
Üye	**: Doç. Dr. ATAKAN DOĞAN**	**…….**
Üye	**: Yard. Doç. Dr. CÜNEYT AKINLAR**	**…….**
Üye	**: Yard. Doç. Dr. OSMAN ABUL**	**…….**
Üye	**: Yard. Doç. Dr. GÜRKAN ÖZTÜRK**	**…….**

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

Enstitü Müdürü

ABSTRACT

Ph.D. Dissertation

PRIVACY-PRESERVING COLLABORATIVE FILTERING ON ARBITRARILY PARTITIONED DATA

İbrahim YAKUT

Anadolu University
Graduate School of Sciences
Computer Engineering Program

Supervisor: Assoc. Prof. Dr. Hüseyin POLAT
2012, 148 pages

Data collected for collaborative filtering purposes might be arbitrarily partitioned between two parties, even rival companies. Online vendors might have insufficient user ratings. Scarce data then might cause offering inaccurate and unreliable recommendations. In order to supply trustworthy and dependable predictions, one solution for such companies might be cooperation on partitioned user preference data. However, it is still a challenge to convince e-commerce sites cooperate on partitioned data so that they can provide richer collaborative filtering services, due to privacy concerns. Unless confidentiality is protected, such companies are expected to face with serious legal and financial deadlocks in managerial operations.

This study aims to scrutinize how to estimate predictions based on arbitrarily partitioned data configurations between two e-commerce companies without deeply jeopardizing their privacy. Privacy-preserving schemes are proposed to offer numerical or binary recommendations using item-based, trust-based, and naïve Bayesian classifier-based prediction algorithms on arbitrarily partitioned data. Along the study, how two parties ended up with cross partitioned data can provide CF services using hybrid CF algorithm is also investigated. It is shown that each proposed method does not intensely violate data owners' confidentiality. The proposed schemes are also investigated in terms of supplementary computation, communication, and storage overheads. Experimental trials are conducted using real data sets to show how the quality of the predictions improves due to collaboration and privacy measures affect accuracy. All appraisements demonstrate that the proposed solutions are preferable for estimating higher quality predictions efficiently on partitioned data while preserving data holders' privacy.

Keywords: Privacy, Collaborative Filtering, Arbitrarily Partitioned Data, Performance, and Accuracy.

ÖZET

Doktora Tezi

GİZLİĞİ KORUYARAK RASTGELE BÖLÜNMÜŞ VERİ TABANLI ORTAK SÜZGEÇLEME

İbrahim YAKUT

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Hüseyin POLAT
2012, 148 sayfa

Ortak süzgeçleme amacıyla toplanan veriler iki firma hatta rakip şirketler arasında rastgele şekilde bölünmüş olabilir. Sanal alışveriş siteleri yetersiz kullanıcı oylarına sahip olabilirler. Yetersiz veri, hatalı ve güvenilir olmayan öneriler üretmeye sebep olabilir. Güvenilir öneriler sağlamak için bu şirketlerin parçalanmış tercih verileri üzerinden işbirliği yapmaları bir çözüm olabilir. Bununla birlikte gizlilik endişelerinden dolayı, e-ticaret sitelerinin bu şekilde dağılmış veri üzerinden daha iyi ortak süzgeçleme hizmetleri sağlamaları ciddi bir sorun teşkil etmektedir. Gizlilik sağlanmadığı takdirde, bu şirketlerin idari süreçlerinde ciddi hukuki ve finansal çıkmazlarla karşı karşıya gelmesi durumu söz konusudur.

Bu çalışma iki sanal alışveriş sitesinin gizliliklerini tehlikeye atmadan rastgele bölünmüş veri üzerinden nasıl öneri üretebileceklerini incelemeyi amaçlamaktadır. Ürün-tabanlı, güven-tabanlı ve basit Bayes sınıflandırıcı-tabanlı algoritmalar kullanılarak rastgele bölünmüş veriler üzerinden nümerik ve ikili öneriler üreten gizlilik korumalı yöntemler önerilecektir. Çalışmada ayrıca çapraz bölünmüş verilere sahip iki şirketin hibrit ortak süzgeçleme algoritmaları kullanarak nasıl öneriler üreteceği ele alınmıştır. Önerilen her bir metod gizlilik açısından irdelenecektir. Ayrıca, önerilen yöntemler ilave hesaplama, haberleşme ve saklama yükleri açısından da incelenecektir. İşbirliğinin öneri kalitesini nasıl artırdığını ve gizlilik ölçütlerinin doğruluğu nasıl etkilediğini göstermek için gerçek verilerle deneyler yapılacaktır. Bütün incelemeler ve deney sonuçları önerilen çözümlerin rastgele bölünmüş veriler üzerinden e-ticaret sitelerinin gizliliklerini ihlal etmeden ve etkin bir şekilde daha kaliteli öneriler üretmek için tercih edilebileceğini göstermiştir.

Anahtar Kelimeler: Gizlilik, Ortak Süzgeçleme, Rastgele Bölünmüş Veri, Performans ve Doğruluk.

ACKNOWLEDGEMENTS

I would like to thank Assoc. Prof. Dr. Hüseyin Polat for his support, guidance, and acceleration to my research. Also, his vision, wisdom, and patience significantly promote the quality of the study. It was a privilege and pleasure to work with him since my Master of Science thesis. I have taken his great amount of time and effort during research studies up-to-now.

I would also like to thank Assoc. Prof. Dr. Atakan Doğan, Assist. Prof. Dr. Cüneyt Akınlar, Assist. Prof. Dr. Osman Abul, and Assist. Prof. Dr. Gürkan Öztürk for serving on my dissertation committee and for their valuable contributions.

I would also like to thank my colleagues, Cihan Kaleli and Alper Bilge, for their scientific support.

A heartful thanks go to my parents. I am very grateful for their devotion and efforts on me.

Finally, I would like to thank my wife for her substantial support and patience during all my studies.

İbrahim Yakut

June, 2012

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

a	:	Active User
APD	:	Arbitrarily Partitioned Data
CA	:	Classification Accuracy
CF	:	Collaborative Filtering
CPD	:	Cross Partitioned Data
d	:	Density
D	:	User-Item Matrix
DC	:	Data Controller
DLCP	:	Denominator of Likelihood Computation Protocol
E(x)	:	Expected Value of x
F1	:	F-Measure
HDD	:	Horizontally Distributed Data
HE	:	Homomorphic Encryption
HPD	:	Horizontally Partitioned Data
k-nn CF	:	k-Nearest Neighbor-based Collaborative Filtering
m	:	Number of Items
MAE	:	Mean Absolute Error
MLM	:	MovieLens Million
MLP	:	MovieLens Public
MP	:	Master Party
n	:	Number of Users
NBC	:	Naïve Bayesian Classifier
NLCP	:	Numerator of Likelihood Computation Protocol
NMAE	:	Normalized Mean Absolute Error
OREP	:	Online Recommendation Estimation Protocol
OT	:	Oblivious Transfer
PACEP	:	Private Adjusted Cosine Estimation Protocol
p_{aq}	:	Prediction on Item q for User a
PGP	:	Prediction Generation Protocol
PPCF	:	Privacy-Preserving Collaborative Filtering

PPEP : Private Priori Estimation Protocol

P2D2M : Privacy-Preserving Distributed Data Mining

P2P : Peer to Peer

P3CF : Privacy-Preserving Partitioned Collaborative Filtering

RMSE : Root Mean Square Error

RPT : Randomized Perturbation Techniques

RRT : Randomized Response Techniques

q : Target Item

SMC : Secure Multi-party Computation

STCP : Secure Trust Computation Protocol

$t_{a \to u}$: Trust between Users a and u

v_d : Default Rating

VDD : Vertically Distributed Data

v_f : Fake Rating

VPD : Vertically Partitioned Data

$\xi_K(x)$: Encrypted Value of x with Key K

τ : Threshold

1. INTRODUCTION

Collaborative filtering (CF) algorithms are widely used by online vendors to provide predictions to their customers. It is possible to increase sales and/or profits through successful CF systems. E-commerce companies effectively use them either providing predictions or top-*N* lists directly or presenting product-related information fitting best to the customers' taste while surfing over their pages. CF-based recommender systems not only guide the users about products such as books, movies, music CDs, restaurants, and so on, which they have not sufficient information on yet, but also promote sales and/or visiting hits of e-commerce sites. Such systems also provide web-based personalization on a range of products.

In case of inadequate data, it becomes a challenge to produce recommendations for all items; which leads to very low coverage. Some vendors especially newly established ones, might face with the cold start problem. In other words, they are not able to provide satisfactory predictions in quality and/or quantity due to insufficient data. Since similarities between users are computed over commonly rated items, *data scarcity* makes it difficult to find large enough commonly rated items. The similarities, computed over a small number of commonly rated items, then can be considered untrustworthy. Furthermore, to have a large enough neighborhood, data owners should have sufficient number of users. Since interrelated customers' preferences about various products may be available in partitioned manner between two CF provider parties, in order to overcome data scarcity problem in recommender systems, such parties can cooperate on held data. However, due to privacy, legal, and financial reasons, they might not want to reveal their data to each other. If they are assured about the privacy of each own data, they can decide on cooperation over partitioned data to promote CF services.

Depending on the availability of customer data, two data holder parties can end up with different configurations of partitioned data. While this partition can be horizontal or vertical, in practice, it is more likely to be arbitrarily. This study focuses on *how CF services can be provided on arbitrarily partitioned data configurations between two parties while ensuring their privacy*. After proposing

solutions in such problematic framework, the proposals are going to be justified about privacy, efficiency, and output quality via theoretical and empirical analysis. It is shown that the offered schemes can be solutions against data scarcity problem in CF recommender systems.

This chapter is structured, as follows: CF and the related terminology are presented in Section 1.1. After privacy-preserving collaborative filtering (PPCF) schemes are pointed out in the following section, arbitrarily partitioned data is defined in Section 1.3. Then, privacy preservation issues and the contributions are presented, respectively. At the end of the chapter, the organization of this thesis is introduced.

1.1. Collaborative Filtering

Since CF's conceptual introduction with *Tapestry* mail filtering system (Goldberg et al., 1992), there have already been various collaborative recommender proposals in the state-of-the-art. Some algorithms focus on improving *scalability* of CF systems by using dimensionality reduction tools such as singular value decomposition (SVD) (Sarwar et al., 2000), principal component analysis (PCA) (Goldberg et al., 2001), and clustering methods (Breese et al., 1998). There are also studies examining how to tackle with *data sparsity* (Papagelis et al., 2005; Kaya and Alpaslan, 2010) and *cold start* problems (Ahn, 2008; Li et al., 2009). *Shilling attack* scenarios are proposed in order to manipulate recommendation lists, maliciously (Ray and Mahanti, 2009). Some techniques are also offered to make CF systems robust against such attacks (Chirita et al., 2005; Ji et al., 2007). *Novelty* and *diversity* are other issues that take attention of CF researchers (Hurley and Zhang, 2011).

In addition to CF, *content-based filtering* methods can be utilized for information filtering and recommendation purposes. While content-based filtering methods work on the inputs such as genres and synopses of movies, books, etc., CF operates on the user-item profile data such as ratings, preferences, and transactions. In contrast to content-based filtering, CF serendipitously provides filtering of items whose contents are too complex to be analyzed by computers (Herlocker et al., 1999). In addition to Tapestry (Goldberg et al., 1992), there are also hybrid information filtering solutions integrating content-based and CF

models (Balabanovic and Shoham, 1997). Vozalis and Margaritis (2007) contribute demographic data into CF process to enhance recommendations.

In a typical CF process, as shown in Fig. 1.1, the key elements are an $n \times m$ user-item preference matrix, an active user a, and a target item q. The main idea behind CF is that a will prefer those items that like-minded users prefer, or that dissimilar users do not. CF systems provide a prediction to user a about a q based on the preferences of a community of users (Herlocker et al., 1999). Usually, a has insufficient idea about q and CF contributes a's decision process. Alternatively, CF algorithms recommend top-N lists with respect to a's rating profile and similarities with a set of users. After collecting customers' preferences about products they have already experienced, CF input data is constructed as $n \times m$ user-item matrix (D), where n and m represent the number of users and items, respectively. The systems then provide prediction p_{aq} for a about q grounded on D, a's known ratings, and her prediction query. Generally speaking, CF process consists of the following three main steps (Herlocker et al., 1999):

i. *Similarity Estimation:* Determine similarities between any two entities (users or items) with the same type using a similarity measure.
ii. *Neighborhood Formation:* For a target entity, choose the best similar entities as neighbors either off-line or online.
iii. *Recommendation Computation:* Estimate a prediction from the neighbors' data using a CF algorithm; and return it as a recommendation to a.

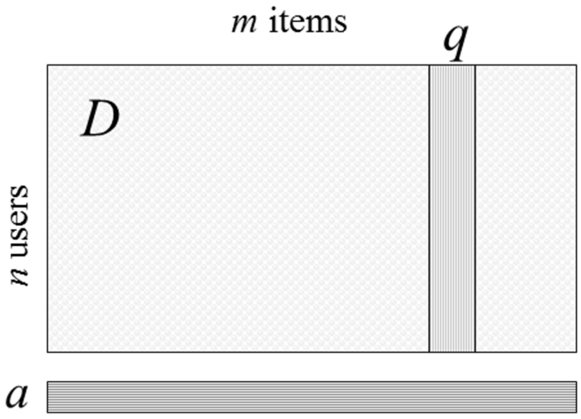

Figure 1.1. Key Elements in CF Process

Online prediction process can be performed either directly on *D,* or pre-constructed model. The former kinds of algorithms are called *memory-based* algorithms while the latter type ones are referred to as *model-based* algorithms. Conventional CF algorithms are memory-based and have some scalability problems. Since the main computations are realized on commonly rated items, when data are very sparse such algorithms significantly degrade prediction accuracy and coverage. To overcome the deficiencies of memory-based CF algorithms, model-based algorithms featuring off-line computation process are proposed. However, in general, memory-based CF produces more accurate predictions over model-based ones. To benefit from both kinds of algorithms, *hybrid* schemes are also proposed. Moreover, CF algorithms can be classified with respect to entities, too. Some methods compute similarities between users, hence known as *user-based CF* algorithms. If such relations are constructed based on items, then they are called *item-based CF* method. Lastly, rating data type in CF can be *numerical* or *binary* values. Numerical ratings can be *discrete*, *continuous*, or even *subzero* depending on data collection mechanism.

1.2. Privacy-Preserving Collaborative Filtering

Privacy has many definitions according to different perspectives in historical background and it is hard to define it concisely. In information theoretic framework, a systematic definition is firstly propounded by Warren and Brandeis (Judith, 2008). Warren and Brandeis (1890) consider privacy as "right to liberty secures the exercise of extensive civil privileges". According to Westin (1967), privacy can be defined as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others". However, as the other information-based systems, emerging collaborative recommendation service technologies threaten this right or claim for both individuals and institutions. Let us consider individual case. In order to benefit from recommendation services, a user must provide some personal data especially preference or taste data to such systems, i.e. *data controller* (DC)[1]. By knowing these, DC might figure out the user and based on her profile, DC or its collaborator companies may disturb her by spam

[1] OECD, Guidelines on the Protection of Privacy and Transborder Flows of Personal Data, 2005.

conversations, e. g. mails, phone calls, and so on for unsolicited marketing. They may also utilize such profile to offer higher prices for enjoyed items or services in price discrimination manner. Cranor (2004) extensively examines the possible privacy risks caused by e-commerce personalization. Such privacy concerns make users not to share opinions and taste information with DCs without being sure of the confidentiality of personal data. Lam et al. (2006) discuss probable exposure risks in case of "undesired access to personal user information" in the framework of CF recommender systems.

There are also studies introducing attacks to infer private information from recommender systems. The first attack scenario trying to show the privacy risks in recommender systems is presented by Ramakrishnan et al. (2001). Their scenario assumes an attacker has anonymized version of the rating database and based especially on *straddlers* having diverse tastes and ratings over different genres of items. Calandrino et al. (2011) investigate how to infer individual user profile from the aggregated outputs and individuals' auxiliary information such as a subset of her transactions, Facebook user profile, cited text from which books are tweeted by Amazon Kindle. They publish experimental results of their inference attack and conclude that with the help of some auxiliary information, user profiles can be deduced via inference attack on public outputs such as item similarity lists, item-to-item covariance, and/or relative popularity items. Cheng and Hurley (2009) argue informed attacks against particular vulnerability for model-based CF algorithms. In particular, they show the risk that such attacks can be applied on peer-to-peer (P2P) recommendation algorithms.

In addition to user-to-DC transactions, there are some situations, where two parties may need to exchange some personal data. However, such DC-to-DC data exchange may cause serious complications related to data confidentiality. First of all, users' preferences about different products may help companies to profile their customers in such a way to increase their sales and/or profits. Online vendors offer different discounts and coupons to their customers based on unrated items. Since revealing such information may cause financial losses, data collected for CF purposes are considered valuable asset. Secondly, each company is responsible for protecting the collected ratings about their customers and data transfers may not

be possible due to legal reasons. Kobsa (2007) points out that users are uncomfortable about a DC sharing their personal data with other DCs. Organization for Economic Co-operation and Development (OECD) stated that exposure of customers' personal data is serious issue and DCs must protect such data (OECD, 2000; OECD, 2005). Finally, customers' preferences about products held by companies are considered online vendors' confidential data.

Based on user-to-DC and DC-to-DC privacy issues mentioned above and as schematized in Fig. 1.2, two privacy definitions can be defined in the context of CF systems, as follows:

Individual Privacy: No exposure of any rating value and any information about which items are rated in each user profile. No doubt about the other personal data because CF mainly operates on ratings rather than demographic data (Cranor, 2004). In addition to rating values and rated items, Lathia et al. (2007) consider the mean rating for any user and the total number of items rated by any user confidential.

Corporate Privacy: No transactions causing information leakage conflicting individuals' privacy between two or more DCs that are responsible for protecting privacy of users. They should hold the user profiles firmly so that they would not support their competitors in personalization power.

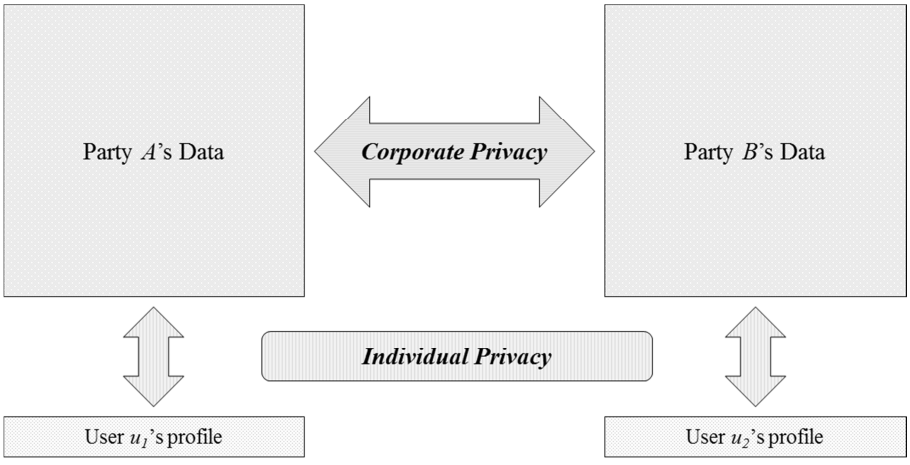

Figure 1.2. Privacy Issues

PPCF schemes can be grouped into two major classes with respect to privacy definitions, as shown in Fig. 1.3. PPCF works conducted so far can be

discussed in such classification framework. According to the framework, PPCF schemes on APD between two parties are discussed in this thesis.

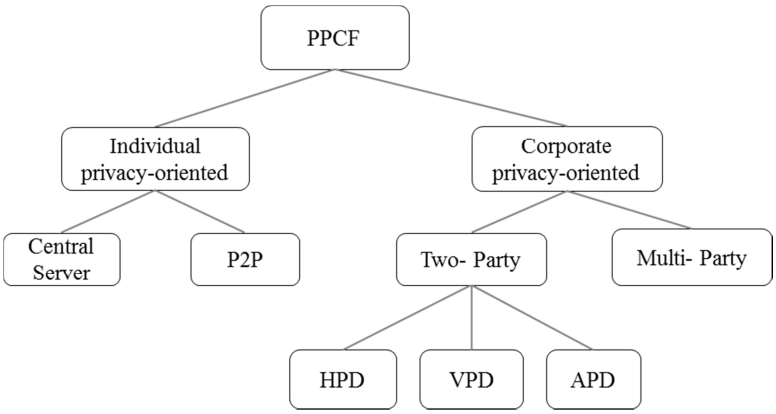

Figure 1.3. State-of-the-Art in Privacy-Preserving Collaborative Filtering

1.2.1. Individual Privacy-oriented Schemes

Individual privacy takes attention of CF research community. In state-of-the-art, there are techniques focusing on how to collect user profiles while satisfying individual privacy protection. Agent-based architectures are also proposed to ensure confidentiality of individuals. Moreover, to get rid of concerning about data collection in the centralized manner, P2P network solutions are also applied. Such classes of solutions are given in detail in the following.

Since privacy metrics make sense for significant rate of users (Ackerman et al., 1999), DCs must verify to users that their recommendation system ensures their privacy. By the way, they promote own data quality and quantity, too. One solution for providing such privacy metrics is *randomized perturbation techniques* (RPT). Aggrawal and Srikant (2000) show that aggregates obtained by obfuscating sensitive values using particular randomized processes remains still valuable for data mining applications. Polat and Du (2005a) applies RPT to both correlation and SVD-based CF algorithms; and propose privacy-preserving solutions for recommendation services. Their empirical findings show that result predictions have satisfying accuracy with some losses due to random masking of private data. Berkovsky et al. (2007) examine how users approach to data obfuscation in PPCF and find that obfuscation-based privacy protection makes the Internet users more willing to share their personal information with DCs.

Yakut and Polat (2007b) apply disguising framework to the linear time recommendation algorithm Eigentaste (Goldberg et al., 2001). Moreover, Bilge and Polat (2012) utilize similar disguising framework on discrete wavelet transform (DWT)-based CF algorithm. They also enhance prediction quality using two different item ordering methods. To overcome scalability and sparsity problems in privacy-preserving conventional correlation-based CF system, Bilge and Polat (2011) offer two distinct preprocessing schemes such as using a novel content-based profiling of users to estimate similarities on a reduced data for better performance and pseudo-prediction protocol to surmount sparsity. Recently, Basu et al. (2012b) introduce privacy-preserving Slope One predictor scheme using additive and multiplicative perturbation techniques. They obtain satisfactory results because the Slope One predictor's invertible affine transformation property is robust to certain types of noise.

In another approach, to fulfill the individual privacy demands in central data-based PPCF, Parameswaran and Blough (2007) merge the nearest neighborhood data substitution (NeNDS) and geometric transformations such as rotation, scaling, and translation by taking advantages of both methods and present hybrid NeNDS-based obfuscation scheme to ensure privacy of personal rating data. According to their results, there are 5% losses of utility of ranking order in obfuscated data with respect to original data.

Cissée and Albayrak (2007) focus on the recommender system functionality and propose an approach utilizing multi-agent system architecture to realize a privacy-preserving recommender. Their proposal is based on fundamental features of agents such as autonomy, adaptability, and the ability to communicate. Aïmeur et al. (2008) introduce a theoretical framework named ALAMBIC in order to achieve users' privacy-protection objectives in a hybrid recommender system combining content-based, demographic, and CF techniques. Using agent-based architectures, their scheme splits customer data between DC and a semi-trusted third party (STTP), so that neither can derive sensitive information from their share alone. To ensure privacy in ALAMBIC, one must be sure about that DC and STTP are not colluding.

Verhaegh et al. (2004) offer a solution handling similarity computation and prediction estimation on encrypted rating profiles to protect individual privacy in memory-based CF system. Katzenbeisser and Petkovic (2008) consider patient privacy in medical recommendation system and provide cryptographic solutions ensuring patients' confidentiality. Hoens et al. (2010b) point out privacy issues in similar application area and introduce a privacy-preserving solution based on data masking and anonymization procedures. Brun and Boyer (2009) focus on the study of how sequential association rules and Markov models can be adapted to obtain privacy compliant recommender system serving anytime. They respect privacy via executing recommender algorithm on input of anonymized user traces without using any other personal information. To address the scalability issues, Tada et al. (2010) introduce item-based solution with ensuring individual privacy via homomorphic encryption (HE) schemes. In order to facilitate the computations, they also contemplate the postulation that item similarities can be publicly available rather than user-user similarities.

There are also central data-based PPCF solutions on binary ratings. Polat and Du (2006) propose a framework that collects binary data privately by *randomized response techniques* (RRT). They showed the possibility of providing recommendations by item-based CF algorithm on perturbed data via RRT. Kaleli and Polat (2007b) apply RRT to protect users' privacy while producing accurate referrals using naïve Bayesian classifier (NBC). They also showed that it is possible to improve the overall performance of NBC-based CF with aid of *k*-modes clustering while preserving users' privacy including active users (Kaleli and Polat, 2009). Their experimental results demonstrate that their scheme not only significantly reduces online time but also enhances accuracy of referrals slightly. To provide more truthful recommendations by privacy-preserving NBC-based CF, Bilge and Polat (2010) offer preprocessing steps such as selecting the best similar products to each item and filling the unrated cells with personalized ratings. They experimentally show that their modifications enhance the prediction quality while slightly worsen the efficiency.

Surveys conducted on the Internet users show that individuals either being aware of privacy risks or not perceive privacy concept in different levels; and they

behave based on their privacy perceptions on the Internet (Ackerman et al., 1999; Spiekermann et al., 2001). Both surveys determine general clusters about privacy concerns about users. In order to adapt their findings on privacy-preserving recommender system, Aïmeur et al. (2008) classify users into four levels and define *no*, *soft*, *hard*, and *full* privacy. In *no* privacy case, a user does not care about the privacy of her personal information. Users having *soft* privacy profile behave like identity concerned as Spiekermann's definition (Spiekermann et al., 2001). While *hard* privacy allows DC about browsing behavior and actual purchasing information, *full* privacy anticipates keeping secret whole of personally direct/indirect information from DC. Polat and Du (2007) offer inconsistently masking data procedure as a solution to meet the diverse privacy expectations of users. For this reason, they discuss all possible privacy cases for RPTs. They also investigate how to provide predictions using correlation-based CF algorithm on *inconsistently* masked data. Yakut and Polat (2007a) propose a model-based CF solution on this variant. Both of the solutions give the Internet users specify masking parameters from a particular range.

In the context of provision of individual's privacy, realizing secure computation protocols on fully distributed environments have been placed as a solution, too. Canny (2002a) propose a scheme in which users control all of their log data. In his algorithm, rather than processing individual-wise data, via some encryption schemes, a community of users can compute a public "aggregate" of their data, which allows personalized SVD-based recommendations to be computed by members of the community, or by outsiders. Canny (2002b) also introduces P2P recommender system based on factor analysis.

Miller et al. (2004) propose a recommender system providing referrals to users in P2P network without storing user rating profile in conjunction with user identity and place. Berkovsky et al. (2005) offer data obfuscation in decentralized recommender system and set up experiments for a range of obfuscation policies to show the applicability of their proposal. The authors in (Berkovsky and Kuflik, 2006) include hierarchy in peer neighborhood formation to avoid sharing obfuscated user profiles among all peers.

Lathia et al. (2007) present concordance metrics to evaluate similarities between peers rather than direct computations on rating values. Shokri et al. (2009) propose hybrid architecture that utilizes both server and distributed peer network. Each user stores her profile off-line, modifies it by partly merging it with the profile of similar users through direct contact with them, and only then periodically uploads her profile to the server. Kaleli and Polat (2010) investigate how to provide referrals on binary data in P2P manner and propose NBC-based solution in this variant. Their privacy-preservation is based on RRT. Ahn and Amtriain (2010) present an implementation of expert CF in fully distributed settings. They develop rich internet application (RIA) by combining RESTful architectural style with Linked Data's basic principles, where REST and Linked data are latest web technologies.

There also P2P proposals in such trendy topics as cloud and ubiquitous computing environments. Ahmad and Khokhar (2007) introduce bi-clustering-based PPCF solution for ubiquitous computing infrastructure. While distributed bi-clustering excludes the need for trusted servers, the privacy preservation is based on HE. Basu et al. (2011b) focus on how to provide recommendations over cloud acting as P2P network and propose practical implementation of privacy-preserving weighted Slope One predictor on a real world computing platform.

Clients' privacy issues in the nowadays' Internet hit social network recommendation systems take care of PPCF research community. Chen and Williams (2010) point out the key problems that arise from the privacy dimension of social recommendations; and present an architecture to develop privacy-aware cooperative social recommender systems. Dokoohaki et al. (2010) integrate both trust-aware recommenders and privacy needs in social network; and introduce a privacy-preserving trust-aware recommender framework for social networks. Hoens et al. (2010a) develop a recommendation system for social networks, which protects the privacy of user profile while allowing them to learn aggregate results about ratings. Erkin et al. (2011) design privacy-enhanced recommender system for a social trust network. While they apply HE and secure multi-party computation techniques to ensure privacy, they improve the efficiency through computation and communication costs by packing data. Machanavajjhala et al.

(2011) examine privacy-utility trade-off for graph-based social recommender system. YANA (Li et al., 2011a) and Pistis (Li et al., 2011b) are two privacy-preserving recommender proposals for online social communities.

1.2.2. Corporate Privacy-oriented Schemes

Data scarcity problems faced by DCs bring about *privacy-preserving distributed data mining* (P2D2M) solutions. However, corporate privacy considerations are the foremost challenge of such solutions and DCs. In other words, e-commerce companies can only cooperate when privacy measures are satisfied. In this variant, two parties can end up with three kinds of data configurations: horizontal, vertical, or arbitrary. Such configurations can be briefly defined in the jargon of e-commerce, as follows: In horizontally partitioned data (HPD), two parties have the same item portfolio and the disjoint set of users while in vertically partitioned data (VPD), they end up with the same set of users and the disjoint set of items. In practice, the first one may be available when parties in the same commercial area open a stall for different regions as the latter one occurs for parties selling different set of products and having the same customer profile. Intuitively speaking, while HPD is profitable when there is insufficient number of users, VPD is advantageous when data holders have ratings belong to the limited number of items. Lastly, arbitrarily partitioned data cases can be defined as availability of records belongs to similar set of users for the similar set of items in arbitrary manner rather than purely HPD or VPD. Such configurations are familiarized in detail in Section 1.3.

In the context of P2D2M, HPD is firstly studied by Kantarcioglu and Clifton (2004). They address the secure mining of association rules over HPD using commutative encryption (Pohlig and Hellman, 1978) and Yao's secure function evaluation (Yao, 1986). Kantarcioglu and Vaidya (2003) present schemes for learning NBC on HPD securely. Yi and Zhang (2009) consider privacy-preserving NBC for HPD and propose both two-party and multi-party protocols to achieve it. In another study, Inan et al. (2007) study how to construct dissimilarity matrix privately on HPD among different sites. Yang and Huang (2008) present a clustering method for horizontally distributed multi-party data sets with privacy based on the orthogonal transformation and perturbation techniques. Emekci et al.

(2007) propose a method to build decision trees over HPD among multiple parties up to thousands of private data sources. Kaya et al. (2009) present a distributed clustering protocol for HPD based on a very efficient homomorphic additive secret sharing scheme with privacy.

As HPD, VPD extensively investigated by P2D2M researchers, too. Vaidya et al. (2008a) introduce a generalized privacy-preserving variant of the ID3 algorithm for VPD over two or more parties. Along with the algorithm, the authors prove that its security gives a tight bound on the information revealed. Oliveria and Zaïane (2007) use random projection techniques in clustering toward secure and effective data analysis for business collaboration. They offer solutions for vertically distributed configuration and also for centralized data. Skillicorn and McConnell (2008) present a simpler prediction approach for VPD. The method works in distributed computing manner in which work load is shared by all parties. Rozenberg and Gudes (2006) deal with the problem of association rule mining from VPD with the goal of preserving the confidentiality of each database and offer two different solutions. Yi and Zhang (2007) propose a privacy-preserving association rule mining protocol based on a new semi-trusted mixer model for VPD.

Polat and Du (2005b; 2005c) introduce *privacy-preserving partitioned collaborative filtering* (P3CF) problem in two different studies. In P3CF, the key question is "how can two e-commerce companies offer CF on partitioned data without disclosing their data to each other?" In (Polat and Du, 2005c), the authors investigate both threshold and best-*N* neighborhood determinations on obtained similarity values using modified Tanimoto metric and propose top-*N* recommendation solutions for horizontally partitioned binary data. The same authors examine how to realize recommendations using correlation-based CF algorithm on VPD (Polat and Du, 2005b). Kaleli and Polat (2007a) propose P3CF solutions for NBC-based binary referral estimation on both HPD and VPD, alternatively. Yakut and Polat (2010) examine how to estimate SVD-based predictions while guaranteeing corporate privacy of HPD and VPD; and introduce a model-based P3CF scheme. Hsieh et al. (2008) focus on correlation-based CF on HPD with corporate privacy; and propose a P3CF framework utilizing El

Gamal-based HE. Zhan et al. (2008) investigate empirically efficiency issues in P3CF on HPD by comparing computation and transportation time costs of El Gamal-, commodity-, and their revised commodity-based approach; and their experimental findings show that the revised approach outperforms the others.

There are also PPCF studies caring about the problematic cases, which more than two DCs exist. Kaleli and Polat (2012b) discuss the challenges of vertically distributed data (VDD) among multi-party and propose a privacy-preserving self-organizing map (SOM) algorithm for multi-party collaboration. The same authors also consider horizontally distributed data (HDD) for SOM-based CF algorithms (Kaleli and Polat, 2012a) and introduce a solution. In another study, Kaleli and Polat (2011) investigate how to provide trust-based recommendations on VDD while preserving corporate privacy.

Basu et al. (2011a) present a privacy-preserving item-based CF scheme through the use of an additively homomorphic public-key cryptosystem on the weighted Slope One predictor; and show its applicability on both HDD and VDD. Basu et al. (2012a) implement the some components of a PPCF method in Java on the Google App Engine, which provides cloud computing platform for web applications in Google-driven data hubs. They observe the feasibility of PPCF services on cloud platform. Their results demonstrate that such engine can have significant performance bottlenecks to realize PPCF services in decent time.

1.3. Arbitrarily Partitioned Data

Two online vendors, *A* and *B*, can end up with APD, as shown in Fig. 1.4. Some ratings are held by *A* while others held by *B*. Since users do not rate all items, there are unrated cells, as well. When *A* and *B* sell similar set of products for the same customers, they might end up with APD. Any user may provide ratings for some items to *A* while she might rate some items held by *B*. As shown in Fig. 1.4, there might be non-common customers and products. Although there is no order of placement of ratings, it is assumed that each user provides one rating only for any item. Thus, there are no overlapping ratings. In other words, users' preferences held by each party are distinct sets.

Jagannathan and Wright (2005) introduce the concept of so called APD. The authors present a privacy-preserving protocol for *k*-means clustering based on

APD. For similar clustering goal, Prasad and Rangan (2007) develop a privacy-preserving BIRCH algorithm on APD. They also introduce secure protocols for distance metrics and give a procedure for using these metrics in securely computing clusters over APD. Han and Ng (2007) propose a privacy-preserving decision tree induction algorithm on APD among multiple parties. Since secure scalar product is a core operation in decision tree induction, their main contribution in this work is a more efficient method to perform the secure scalar product operation on APD among multiple parties. In another study, privacy-preserving support vector machine (SVM) classifier solution on APD is proposed by Yunhong et al. (2010). Despite the fact that the SVM classifier is public in their proposal, it does not divulge any privately held data. According to their empirical analysis on real world data, accuracy results are satisfactory with respect to ordinary SVM classifier.

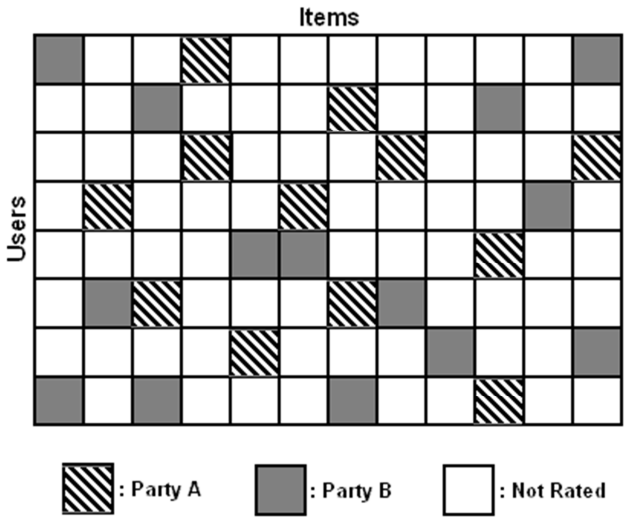

Figure 1.4. APD: Arbitrarily Partitioned Data

Bansal et al. (2010) present a privacy-preserving algorithm for neural network learning when the data are arbitrarily partitioned between two parties. They show that their algorithm leaks no knowledge about other party's data except the final weights learned by the network at the end of training. Upmanyu et al. (2010) propose a solution based on "cloud computing" using the paradigm of "secret sharing" to privately cluster an APD. Li et al. (2011c) offer privacy-preserving distance-based outlier detection protocol on APD.

In addition to APD, data owners might end up with CPD, as shown in Fig. 1.5, where $n_1 + n_2 = n$ and $m_1 + m_2 = m$. Suppose that there are two e-commerce companies, *A* and *B*, where both companies sell the same products to the same community of customers. As seen in Fig. 1.5, *A* holds the ratings of the users from u_1 to u_{n_1} for the items from i_1 to i_{m_1}; and the ratings of the users from u_{n_1+1} to u_n for the items from i_{m_1+1} to i_m, while *B* owns the remaining ratings. *A* and *B* can end up with CPD under one of the following conditions:

i. *A* makes discounts for the items from i_1 to i_{m_1}, while, at the same time, *B* makes discounts for the items from i_{m_1+1} to i_m. During these sales campaigns, users from u_1 to u_{n_1} buy and rate corresponding discounted items from *A* and *B*, respectively. After these discounts are over, *A* makes discounts for the items from i_{m_1+1} to i_m, while, at the same time, *B* makes discounts for the items from i_1 to i_{m_1}. Another group of customers, users from u_{n_1+1} to u_n, then buy and rate corresponding discounted items from *A* and *B*, respectively. Such sales offerings then lead to CPD between *A* and *B*, as shown in Fig. 1.5.

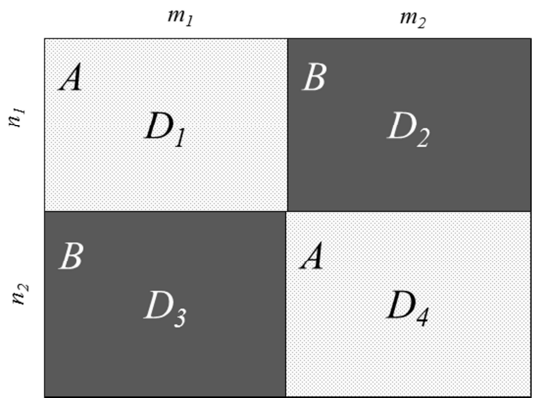

Figure 1.5. CPD: Cross Partitioned Data

ii. Customers choose vendors to buy various products depending on some parameters and intuition (Van den Poel and Buckinx, 2005; Chang et al., 2007). This fact causes profiling of users. There are two types of user profiles, U_1 and U_2, which have different purchasing behaviors. Users in U_1 (including n_1 users from u_1 to u_{n_1}) select companies *A* and *B* to buy

items from i_1 to i_{m_1} and items from i_{m_1+1} to i_m, respectively. Unlike users in U_1, users in U_2 (including n_2 users from u_{n_1+1} to u_n) select e-commerce sites A and B to buy items from i_{m_1+1} to i_m and items from i_1 to i_{m_1}, respectively. Such purchasing profiles then result CPD between A and B, as shown in Fig. 1.5.

iii. Huang et al. (2007) and Shapira et al. (2005) show that web users' online exploration behavior is highly correlated and informative. Thus, customers may select different e-commerce sites at different times to purchase various products. They do not want to give their data to one online vendor only. In this way, they can have more control over their purchasing history, ratings, interested items, viewed items, view duration, and so on. A group of customers (n_1 users) give their ratings for m_1 and m_2 items to A and B, respectively, while another group of users (n_2 customers) provide their ratings for m_2 and m_1 items to A and B, respectively. Such types of privacy concerns lead to CPD between A and B, as shown in Fig. 1.5.

1.4. Privacy Preservation Framework

After giving essential definitions in CF and state-of-art in PPCF and directing sight to the arbitrarily partitioned data configurations, it is now fitting place to express issues related to privacy preservation specifically for this dissertation.

1.4.1. Privacy Constraints

In the generic problem of P2D2M, user-item data can only be mined through collaborating parties without invading corporate privacy of DCs. For this study, the main objective of privacy preservation focuses on hiding the ratings and the rated items. Thus, actual rating values and the rated items are considered confidential data; and the *principal privacy constraint* implies that the proposed scheme cannot allow any leakage inferring the confidential data. During collaborative work, intermediate results are exchanged between aiding parties. Thus, the proposed method cannot involve any exchange of intermediate computation value allowing parties to infer confidential values, either. The *auxiliary privacy constraint* can be stated as "there are no transactions, which

conflict the principal privacy constraint." Since user and item IDs can be thought as public values, sharing them does not violate neither principal nor auxiliary privacy constraints. It is assumed that the collaborating parties are *semi-honest* that obey the defined protocol; however, they might want to process any obtained data either intermediate or final result to learn private values. Moreover, each party can act as an active user in multiple scenarios to derive useful information about confidential data. Hence, the proposed schemes should prevent cooperating companies from learning confidential data and do not allow them to jeopardize the principal and the auxiliary privacy constraints.

1.4.2. Privacy-Preserving Methods

To achieve privacy, the proposed schemes mainly exploit randomization-based techniques, encryption schemes with homomorphic property, and oblivious transfer. Since randomization-based techniques take shape depending on the usage, they are explained in detail in the dissertation, where it is proposed to apply. The homomorphic cryptosystems are useful to perform addition and multiplication operations based on private data. Since the first introduction of homomorphic cryptosystems by Goldwasser and Micali (1984), several such systems have been proposed (Naccache and Stern, 1998; Paillier, 1999). Since Paillier cryptosystem avoids many of the drawbacks of the earlier homomorphic cryptosystems and provides faster encryption and decryption comparing to its alternatives (Pedersen et al., 2007), it is preferred to be utilized, where HE schemes are required along this dissertation.

Based on public cryptosystems infrastructures, Paillier (1999) proposes an additive HE method with self-blinding property. Via the specified method, cipher-texts can be processed to get encrypted version of their sums. Suppose that x and y are two numbers while ξ_K is encryption function with key, K. Then, encrypted versions of the numbers are $\xi_K(x)$ and $\xi_K(y)$. According to Paillier's scheme, multiplication of these cipher-texts results cipher-text of their sum; in other words, $\xi_K(x) \times \xi_K(y) = \xi_K(x + y)$. Based on this rule, his scheme also supports multiplication, which can be performed as analogous manner: $\xi_K(x)^y = \xi_K(xy)$. Moreover, self-blinding property allows publicly change of cipher-text into another one without affecting the plaintext. This can be achieved by multiplying

cipher-text with R^N, where R is random integer value and N is modulus of the operated public cryptosystem.

Rabin (1981) introduce *oblivious transfer* (OT) concept that provides platform to share desired information by receiver among a range of private information of sender while sender is oblivious about what is received. Principally, it must satisfy three key requirements such as correctness of the received value, confidentiality of n-1 values, and privacy of which one is received. 1-out-of-n OT refers to a protocol, where at the beginning of the protocol one party, Bob has n inputs $X_1, X_2, \ldots, X_n$ 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 (Even et al., 1985). In this study, to afford privacy constraints, OT is included into proposals according to the one proposed by Naor and Pinkas (2001).

1.5. Contributions

Along the study, the generic question focused is that "when data collected for CF purposes are arbitrarily partitioned between two DCs, how do they offer recommendations based on APD without violating their corporate privacies?" Individually, the research problems can be listed, as follows:

i. *Item-based CF on APD*: Two parties want to provide CF services via item-based algorithm on APD.
ii. *Trust-based CF on APD:* Using trust-based CF metrics, how two parties realize CF services based on APD.
iii. *NBC-based CF on APD*: How can a NBC-based CF algorithm be realized on arbitrarily partitioned binary data along two parties?
iv. *Hybrid CF on CPD*: Two online vendors, holding data as CPD, want to provide CF services using hybrid CF methods.

Each solution according to the listed problems must satisfy the abovementioned privacy constraints focusing to ensure hiding value of individuals' ratings and which items are rated. Since privacy, accuracy, and efficiency are conflicting goals, it is expected that privacy preservation brings about some computation, communication, and storage overheads. Randomization-based techniques are also expected to make accuracy worse. The solutions to be

proposed should bring reasonable amount of such listed overheads while promoting the quality of predictions. In other words, they will overcome data scarcity of DCs in realizable way.

The first contribution of this dissertation is to identify and study the problem of P3CF on arbitrarily partitioned data, which have not been studied in the literature. As state-of-art is discussed in earlier text, there are some P3CF solutions on HPD or VPD; however, this work focuses on APD. A couple of solutions about how to provide APD-based private predictions on item-based algorithms, trust-based, and NBC-based CF algorithms have proposed. Since there is no proposal covering PPCF on APD in the state-of-the-art, it is the first study focusing on APD in the context of P3CF. Second, this work is the first to introduce and study CPD concept in general. By the way, a P3CF scheme is proposed on CPD for the first time. Finally, by this dissertation, novel PPCF schemes are proposed on both numerical and binary ratings data. Some protocols, which are proposed to handle the problems pointed out, can be utilized in different tasks of PPDM literature.

1.6. Organization of the Dissertation

In the following chapter, privacy-preserving item-based CF on APD is scrutinized. While how trust-based CF can be realized on APD with privacy is discussed in Chapter 3, privacy-preserving NBC-based CF on arbitrarily partitioned binary data is presented in Chapter 4. How referrals can be estimated over CPD with privacy is the concentration of Chapter 5, where a solution exploiting hybrid CF algorithm is introduced. Finally, conclusions are drawn and future research directions are pointed out in Chapter 6.

2. PRIVACY-PRESERVING ITEM-BASED COLLABORATIVE FILTERING ON ARBITRARILY PARTITIONED DATA

In this chapter, it is scrutinized how to estimate item-based predictions on APD between two e-commerce sites without deeply jeopardizing their privacy. The proposed scheme is analyzed in terms of privacy; and it is demonstrated that the method does not intensely violate data owners' confidentiality. Experiments are conducted using real data sets to show how coverage and quality of the predictions improve due to collaboration. The proposed scheme is also investigated in terms of online performance; and it is justified that supplementary online costs caused by privacy measures are negligible. Moreover, some trials are performed to show how privacy concerns affect accuracy. The empirical results show that accuracy and coverage improve due to collaboration; and the proposed scheme is still able to offer truthful predictions with privacy concerns.

2.1. Introduction

With millions of customers and products, a typical web-based recommender system running conventional memory-based algorithms suffers serious scalability problems (Sarwar et al., 2000). To improve scalability, model- or item-based schemes are preferred over memory- or user-based ones because amount of online computations is relatively less in such approaches. Sarwar et al. (2001) propose an item-based algorithm to enhance scalability. Their algorithm can be considered as a hybrid one including both off-line computations (model construction) and online computations (prediction estimation).

Suppose that data collected for CF purposes are arbitrarily partitioned between two parties, *A* and *B*. Such vendors want to provide recommendations on their integrated data without revealing their confidential data to each other. Moreover, they want to estimate predictions with decent accuracy. Finally, online performance must allow them to offer referrals to their customers efficiently. Thus, the problem is how to offer item-based recommendations with decent accuracy on APD efficiently while achieving data owners' confidentiality. Since privacy, accuracy, and performance are conflicting goals, the proposed scheme should provide equilibrium among them.

CF with privacy on partitioned data is an interesting topic for CF researchers. Polat and Du (2008) show how to offer top-*N* recommendations based on horizontally or vertically partitioned data between two parties without deeply violating the data owners' privacy. They provide sorted list of referrals on binary rating data. In another study, Polat and Du (2005b) discuss how to provide predictions for single items based on VPD between two parties while preserving their privacy. They consider all users in the database as neighbors and utilize the entire users' data for prediction computations. Kaleli and Polat (2007a) investigate how to achieve NBC-based CF tasks on partitioned data with privacy. The authors employ binary ratings and the NBC-based CF algorithm to generate referrals, where the scheme determines whether a will like q or not. Yakut and Polat (2010) scrutinize how to provide SVD-based referrals on partitioned data without greatly jeopardizing data holders' privacy. They offer two different solutions for horizontal or vertical partitioning cases. They show that their scheme does not introduce extra online costs.

Although there are various studies scrutinizing CF on distributed data with privacy, this study is the first one for providing predictions on APD with confidentiality. Since APD is the most probable and practical configuration over HPD and VPD, a novel scheme is going to be proposed in order to offer APD-based referrals while preserving the data owners' privacy. APD promises more useful prediction system and the solution for APD is expected to be more complicated. As recommendation generation algorithm, this study focuses on the item-based algorithm enabling pre-computing of item-item similarities proposed by Sarwar et al. (2001). Similarity computations and neighborhood formations, performed off-line, can eliminate the significant bottleneck of scalability. Rather than generating top-*N* list as in (Kaleli and Polat, 2007a; Polat and Du, 2008), the proposed solution yields predictions for single items. In (Kaleli and Polat, 2007a), the proposed scheme operates over the data featured as binary while this study is based on the numeric data in a specific range. Unlike all related work, data partitioning model concerned in this study is novel.

2.2. Item-based Collaborative Filtering

Item-based CF techniques are proposed to overcome scalability and sparsity challenges of user-based techniques. Since item-item relationships are much more static than the relationship between users, item-based CF enables pre-computation of item-item similarities off-line. Therefore, prediction process consists of only a table look up for similarities and computation of a weighted sum. Sarwar et al. (2001) propose an item-based algorithm, which looks into the set of items that *a* has rated and calculates how similar they are to *q*. The most *k* similar items are chosen as neighbor off-line (referred to as model construction). The referrals are then computed online by taking the weighted average of *a*'s ratings on these similar items. In their scheme, to compute similarities between items, they use cosine-based, adjusted cosine, and correlation-based similarity metrics. They experimentally found that the adjusted cosine similarity, given in Eq. (2.1), performs the best:

$$sim_{ij} = \frac{\sum_{u \in U} (v_{ui} - \overline{v_u})(v_{uj} - \overline{v_u})}{\sqrt{\sum_{u \in U} (v_{ui} - \overline{v_u})^2} \sqrt{\sum_{u \in U} (v_{uj} - \overline{v_u})^2}} \tag{2.1}$$

in which $sim_{i,j}$ is the similarity weight between items *i* and *j*, U is the set of users, v_{uy} is the rating of *u* for item *y*, and $\overline{v_u}$ is the average of user *u*'s ratings. After finding the similarities, final prediction is estimated online as a weighted sum of *a*'s ratings for similar items, as follows:

$$p_{aq} = \frac{\sum_{j \in Neighborhood} (sim_{jq} * v_{aj})}{\sum_{j \in Neighborhood} |sim_{jq}|} \tag{2.2}$$

in which v_{aj} represents a rating of *a* for item *j*.

2.3. Item-based Predictions on Arbitrarily Partitioned Data with Privacy

In this section, the proposed scheme is introduced in detail. An overview of the proposed method is given in Fig. 2.1. As seen from the figure, this approach includes off-line and online phases. In off-line phase, after data masking and preprocessing steps, item vectors' lengths and item-item similarities between each pair of items are estimated using *private vector length estimation protocol* and *private adjusted cosine estimation protocol* (PACEP), respectively. Then, each

party j constructs its own model, $Model_j$. Online phase is triggered by a, where she sends her rating vector and a query (target item q). After normalization and data masking processes, two parties collaborate securely to estimate final prediction value p_{aq}, where they utilize pre-constructed models in online phase.

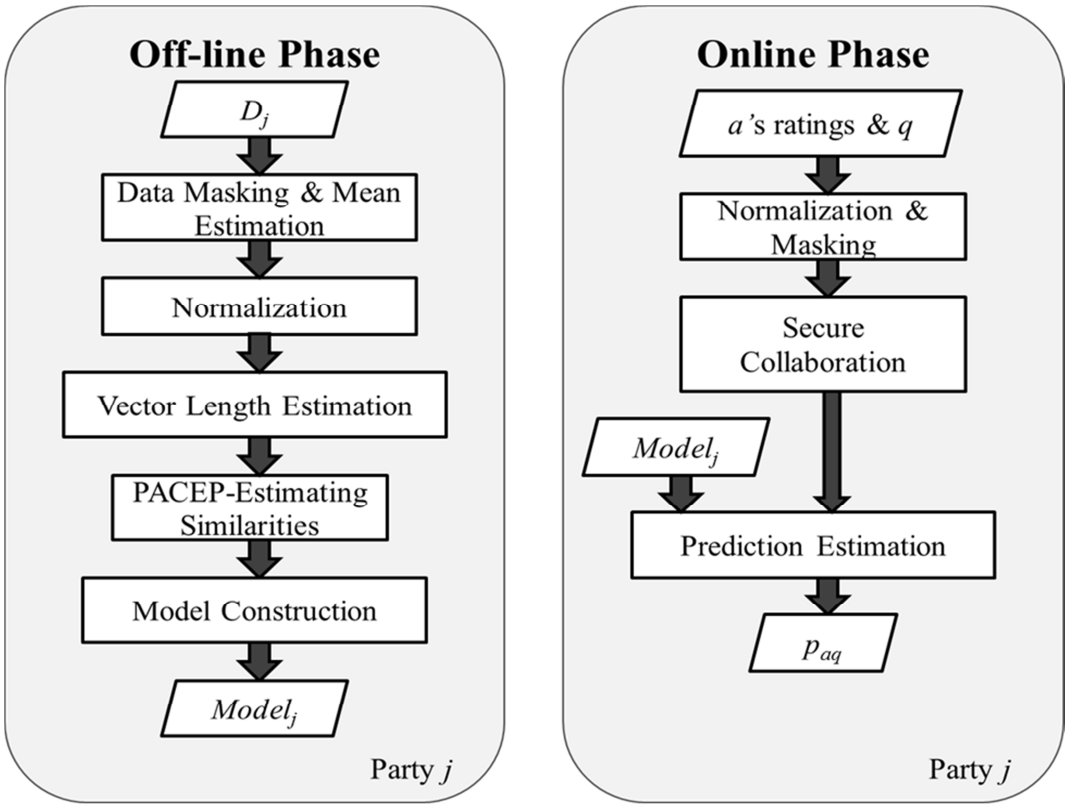

Figure 2.1. Overview of the Proposed Scheme

The user-item matrix D might be arbitrarily partitioned between A and B, as shown in Fig. 1.2., where D_A and D_B represent the sets owned by A and B, respectively. The adjusted cosine similarity is used to compute similarities between items, as proposed by Sarwar et al. (2001). Given two items, i and j, the similarity (w_{ij}) between them can be computed, as follows:

$$w_{ij} = \frac{\overrightarrow{v_i'} \cdot \overrightarrow{v_j'}}{\left\|\overrightarrow{v_i'}\right\| \times \left\|\overrightarrow{v_j'}\right\|}, \tag{2.3}$$

where $\overrightarrow{v_i'}$ and $\overrightarrow{v_j'}$ are two vectors including the mean normalized ratings by subtracting corresponding user average from own ratings, $\cdot$ denotes the dot-product of the two vectors; and $\left\|\overrightarrow{v_i'}\right\|$ and $\left\|\overrightarrow{v_j'}\right\|$ show vector lengths of $\overrightarrow{v_i'}$ and $\overrightarrow{v_j'}$,

respectively. To determine the item q's neighbors, threshold method is used where those items whose similarity with q is bigger than a pre-defined threshold are selected as neighbors. In order to estimate recommendations, the item-based algorithm is employed proposed by (Sarwar et al., 2001), where z-scores are used rather than ratings, as follows:

$$p_{aq} = \overline{v_a} + \sigma_a \times P_{aq} = \overline{v_a} + \sigma_a \times \frac{\sum_{j \in J} w_{qj} \times z_{aj}}{\sum_{j \in J} w_{qj}}, \tag{2.4}$$

where remember that p_{aq} is the prediction for an active user a on a target item q, $\overline{v_a}$ and σ_a represent the mean rating and the standard deviation of a's ratings, J is the set of q's neighbors, w_{qj} is the similarity weight between items q and j, and z_{aj} is the z-score of a's rating on item j. Given a's rating on item j (v_{aj}), $\overline{v_a}$, and σ_a; z_{aj} can be computed, as follows: $z_{aj} = \frac{v_{aj} - \overline{v_a}}{\sigma_a}$. The z-scores need to be computed for a only. Thus, the master party (MP) can easily calculate them after receiving the required data from a where MP is one of the companies asked for prediction by a.

CF process can be divided into off-line and online phases. Since off-line costs are not critical for overall success, computations should be done off-line as much as possible. In the item-based algorithm (Sarwar et al., 2001), similarity weights between items and neighborhood formations can be conducted off-line. Determining each item's neighbors is called model generation. After constructing the model off-line, recommendations can be estimated online based on the model and a's data. The parties should perform such computations in such a way so that they do not reveal their data to each other.

2.3.1. Off-line Phase

In order to protect their data, the parties first mask their private data sets, namely D_A and D_B. They then can estimate the model based on perturbed data. The parties can disguise their data, as follows:

i. Each party j determines the number of unrated item cells in their databases.

ii. They then uniformly randomly choose a value, θ_j, over the range $(1, \beta_j]$.

iii. Next, each party j uniformly randomly selects θ_j percent of their empty cells.

iv. A and B finally fill such cells with fake ratings (v_f); and obtain the masked databases, D'_A and D'_B, respectively.

β_j and v_f can be called as performance parameters because their values definitely affect the overall performance of the scheme. There are various factors that might help the parties determine β_j value like density/sparsity ratio, accuracy and privacy levels, number of cells with double ratings, and the originality of the collected data. With increasing sparsity, number of filled cells increases. Likewise, accuracy might get worse with augmenting randomness while privacy enhances. Due to the nature of data distribution, number of commonly filled cells by both vendors increases with increasing β_j value. Finally, amount of inserted fake ratings affects the originality of the true data. Besides the value of β_j, fake ratings are another factor especially effecting accuracy. In order to determine fake ratings, various techniques can be used. They can be grouped into three main classes, as follows:

i. *Non-personalized ratings*: Utilize corresponding user, item, or overall average ratings, i.e. default ratings (v_ds), computed based on available data as v_f. In other words, each data holder can compute them without help of the other party using its available ratings. For an entity j (user or item), if there are m_j ratings, then the average vote is $\overline{v_j} = \sum_i v_i / m_j$, where i = 1, 2, …, m_j. Similarly, overall average can be estimated, as follows: If there are m_A number of ratings in A's database, then the overall average vote is $\overline{v_A} = \sum_i v_i / m_A$ in which i = 1, 2, …, m_A. After calculating such ratings, uniformly randomly chosen empty cells can be filled with corresponding votes. For example, if any party decides to use user mean votes for data disguising, it fills randomly selected empty cells with corresponding users' rating averages.

ii. *Ratings distribution*: Determine v_f based on available users' ratings distribution. After determining each user's ratings distribution and the values of its parameters (such distribution is usually Gaussian

distribution with μ and σ), data holders generate fake ratings for each user using the determined distribution and the values. They then fill uniformly randomly selected empty cells with related v_f values.

iii. Personalized ratings: Use the most probable values estimated from available data using a prediction algorithm as v_f. In this method, each party first estimates v_f values based on its available data using a *k*-nn CF algorithm (employing Eq. (2.4)). They then filled randomly chosen cells with corresponding personalized ratings.

After masking their databases using fake ratings, the parties can construct a model. Model creation consists of computing the similarities and determining the neighbors. To calculate similarity weights based on normalized data, *A* and *B* should first find user average ratings and then calculate item vector lengths. How building blocks like mean and vector length can be estimated without violating data owners' privacy based on filled databases can be explained, as follows:

Private Mean Estimation Protocol: Given x_u ratings provided by a user u, their arithmetic average $\overline{v_u}$ can be computed, as follows:

$$\overline{v_u} = \frac{\sum_{j=1}^{x_u} v_{uj}}{x_u}, \tag{2.5}$$

where v_{uj} represents the user u's ratings for item j. Note that *sum* and *count* are examples of distributive measures, which can be computed by partitioning the data into smaller sets, computing each measure for each subset, and finally merging them to obtain the final value. Also note that since $\overline{v_u}$ = *sum*/*count*, it is an example of algebraic measure, which can be calculated by applying division function to distributive measures *sum* and *count*. Since the data are arbitrarily distributed between *A* and *B*, Eq. (2.5) can be written, as follows:

$$\overline{v_u} = \frac{\sum_{j=1}^{x_u} v_{uj}}{x_u} = \frac{\sum_{j=1}^{x_{uA}} v_{uj} + \sum_{j=1}^{x_{uB}} v_{uj}}{x_{uA} + x_{uB}}, \tag{2.6}$$

where x_u shows the number of ratings including the fake ones for user u, x_{uA} and x_{uB} represent the number of ratings including the fake ones held by *A* and *B*,

respectively. The parties can estimate $\overline{v_u}$ values for all users, u = 1, 2, …, n, based on the masked databases in a distributive manner, as follows:

i. A and B compute partial sum and count values based on their masked databases for all users.

ii. Then, A sends estimated sub-aggregates for those users with odd indices to B, while B sends estimated sub-aggregates for those users with even indices to A. In other words, the parties exchange sub-aggregates for half of the users.

iii. Next, A and B estimate user mean ratings for even and odd indexed users, respectively.

iv. Finally, they exchange the estimated mean ratings. Thus, at the end of this private protocol, each party ends up with the $\overline{v_u}$ values for all users.

Although it is not easy to perform computations in a distributive manner without sharing any information, the parties exchange smaller amount of data as much as possible. The smaller the amount of data shared, the more privacy they have. Notice that each party sends data to the other party for half of the users in this protocol. Therefore, the companies cannot figure out the sum of the ratings and the rated items for such users during this protocol.

Private Vector Length Estimation Protocol: After estimating the user means, the parties can normalize their ratings by subtracting the corresponding user mean ratings from each vote. The next task is determining item vector lengths based on filled and normalized databases. Since vector length is an example of distributive measure like mean, it can be estimated in a distributive manner, as follows:

$$\left\|\overrightarrow{v'_j}\right\| = \sqrt{\sum_{u \in x_u} (v_{uj} - \overline{v_u})^2} = \sqrt{\sum_{u \in x_{uA}} (v_{uj} - \overline{v_u})^2 + \sum_{u \in x_{uB}} (v_{uj} - \overline{v_u})^2} \qquad (2.7)$$

in which x_u is the set of users who provide ratings including the fake ones for item j, similarly, x_{uA} and x_{uB} represent the sets of users held by A and B, respectively, who provide ratings including the fake ones for item j. The parties can estimate the vector lengths for all items, j = 1, 2, …, m, based on masked databases in a distributive manner, as follows:

i. *A* and *B* first find deviation from mean values, compute their squares, and calculate the corresponding sum values for all items, respectively.

ii. Then, *A* sends estimated sub-aggregates for those items with odd indices to *B*, while *B* sends estimated sub-aggregates for those items with even indices to *A*. They basically exchange sub-aggregates for half of the items.

iii. Next, *A* and *B* estimate vector lengths for even and odd indexed users, respectively.

iv. Finally, they exchange estimated item vector lengths. Hence, at the end of this private protocol, each party ends up with the values for all items.

Private Adjusted Cosine Estimation Protocol (PACEP): The parties normalize their ratings by subtracting the corresponding user mean ratings from each rating and then dividing the result by the corresponding item vector lengths after they estimate the mean and the vector lengths using the proposed protocols, explained previously. Then, Eq. (2.3) can be written, as follows:

$$w_{ij} = \cos(\vec{i}', \vec{j}') = \sum_{u=1}^{n} v'_{ui} \times v'_{uj}, \tag{2.8}$$

which is basically a scalar dot-product between two vectors including normalized ratings and $v'_{ui} = (v_{ui} - \overline{v_u}) / \left\| \vec{v'_i} \right\|$. Since data are arbitrarily distributed, $X = \vec{v'_i}$ and $Y = \vec{v'_j}$ are partitioned between *A* and *B*. Therefore, scalar dot-product of *X* and *Y* can be written, as follows:

$$X \cdot Y = X_A \cdot Y_A + X_A \cdot Y_B + X_B \cdot Y_A + X_B \cdot Y_B. \tag{2.9}$$

A and *B* can estimate $X_A \cdot Y_A$ and $X_B \cdot Y_B$, respectively by themselves because they own such vectors. On the other hand, to compute $X_A \cdot Y_B$ and $X_B \cdot Y_A$, they need to collaborate. Notice again that the computations are based on filled matrices. In the following, how the parties can estimate $X_A \cdot Y_B$ can be explained in a private manner:

– *B* divides Y_B into *f* random vectors, where $Y_B = \sum_{i=1}^{f} Y_{Bi}$.

- *B* encrypts each value in each random vector with *KB* using an HE scheme, where *KB* is *B*'s public key.
- *B* then sends encrypted values to *A*.
- *A* similarly divides X_A into *g* random vectors, where $X_A = \sum_{i=1}^{g} X_{Ai}$.
- Then, *A* finds $\xi_{KB}(Y_{Bij})^{X_{Aij}} = \xi_{KB}(X_{Aij} \times Y_{Bij})$ values using HE property, where *E* represents encryption; and X_{Aij} and Y_{Bij} represent a value in a random vector. HE allows a multiplication operation to be conducted based on the encrypted data without decrypting them. An efficient HE scheme proposed by Paillier (1999) is utilized.
- *A* also divides the result of $X_A \cdot Y_A$ into a_1 random pieces and encrypts each piece with *KB* using the homomorphic scheme.
- *A* then permutes all encrypted values using a permutation function f_{pA}; and sends them to *B*.
- *B* decrypts them and adds them up.
- In order to estimate $X_B \cdot Y_A$, they follow the same steps by switching the roles.
- Finally, to find the similarity weights, they exchange such partial sums in such a way so that for half of the items, the similarity weights are kept by *A*; and for the others, they are kept by *B*.

Model Construction: After estimating the item-item similarities, the parties can now construct a model, which is basically determining the neighbors of each item. For each item, they can select those items satisfying a pre-determined threshold (τ) as neighbors. The optimum values of various controlling parameters like τ in traditional CF approaches are determined experimentally. Sarwar et al. (2001) perform some experimentation to determine the optimum values of some parameters including the neighborhood size. Thus, both parties can use the same threshold value previously determined experimentally. Note that, at the end of the PACEP, for half of the items, the similarity weights are kept by *A*; and for the others, they are kept by *B*. The parties then choose those items, for which they have the similarity weights, satisfying the τ as neighbors. In other words, the model is partitioned between *A* and *B* because for each item, some of

the neighbors and their similarity weights are held by A while the remaining are held by B.

2.3.2. Online Phase: Recommendation Estimation

Once the parties create the model off-line for prediction purposes, based on their distributed data with privacy, they then start providing recommendations online using it. Since the model is distributed between A and B, the parties can estimate predictions in a distributive manner using Eq. (2.4), as follows:

$$p_{aq} = \overline{v_a} + \sigma_a \times \frac{\sum_{j \in J_A} w_{qj} \times z_{aj} + \sum_{j \in J_B} w_{qj} \times z_{aj}}{\sum_{j \in J_A} w_{qj} + \sum_{j \in J_B} w_{qj}}, \tag{2.10}$$

where J_A and J_B are the set of q's neighbors held by A and B, respectively. Suppose that a sends a query to A that acts as a MP. The parties then follow the following steps:

- A finds $\overline{v_a}$ and σ_a values; and computes z_{aj} values.
- Then, A uniformly randomly selects some of a's unrated item cells; and fills them with default z-scores, estimated from available data off-line. A follows the similar steps as done while masking D_A.
- A then removes those items' z-scores from a's vector, which are q's neighbors and held by itself.
- Next, A encrypts the remaining z-scores using an HE scheme using its public key KA; and sends them to B together with the query.
- B divides w_{qj} values into random pieces w_{qjB}; computes $\xi_{KA}(z_{aj})^{w_{qjB}} (\text{mod } n^2) = \xi_{KA}(z_{aj} \times w_{qjB}) (\text{mod } n)$ using the HE property, where HE allows multiplication operation to be conducted on an encrypted value and a plain value without decrypting the encrypted value.
- B then permutes them using a permutation function f_B; and sends them A together with $\sum_{j \in J_B} w_{qj}$.
- A decrypts them and finds $\sum_{j \in J_B} w_{qj} \times z_{aj}$. She finally estimates p_{aq} and returns it to a.

The parties are able to construct a model off-line using the proposed scheme without deeply jeopardizing their confidentiality. After estimating the model, they then provide predictions based on the distributed model using the proposed privacy-preserving scheme. In the followings, the proposed scheme is analyzed in terms of performance, privacy, and accuracy.

2.4. Privacy Analysis

Privacy requirements state that the collaborating parties should not be able to learn true ratings and the rated items held by each other. Before utilizing various protocols, data holders perturb their data by inserting fake ratings. Notice that fake ratings are estimated on available data by each company without sharing any information. Moreover, they do not exchange such fake ratings. Also note that they do not use a single fake value for all selected empty cells. Number of different fake ratings depends on the method they utilize. For example, if they use personalized ratings, they use different votes for each chosen empty cell.

Since the proposed method includes various protocols, they are discussed separately. In the private mean estimation protocol, *A* and *B* exchange partial sum and count (denoted as *M*) values estimated on perturbed data for each user. Given the count values, the parties can guess the number of filled cells and the filled cells with some probabilities. For each party, the probability of guessing the correct θ_j is $1/\beta_j$; and the probability of guessing the correct β_j is 1/100. After guessing the correct θ_j, each party then can find out the number of truly rated items (m_r) from received count values, as follows: $m_r = (\theta_j \times M)/100$. Then, guessing the rated items for each user is 1 out of $C_{m_r}^{m}$, where C_Y^X represents the number of ways of picking *Y* unordered outcomes from *X* possibilities. Also note that $C_Y^X = C_{X-Y}^X = \frac{X!}{Y!(X-Y)!}$. Thus, the probability of guessing the rated items for each user is $1 \Big/ \left(100 \times \beta_j \times \frac{X!}{Y!(X-Y)!} \right)$. Given the partial sums, the parties cannot determine the true ratings as long as $M > 2$, because there are two known values only (sum and count (*M*)) and *M* unknown values. The parties can mask their data by inserting fake ratings in such a way so that $M > 2$ for each user. Note also that the parties do not know the fake ratings estimated by each other. Finally,

as explained before, since they exchange data for half of the users, they are not able to derive information about other users' data.

Unlike the private mean estimation protocol, in the private vector length estimation protocol, the parties exchange partial sum values only. To estimate vector lengths, they do not need count values. They do not know how many users rated each item held by each other. Also, due to filled cells with fake ratings (which are also unknown), normalization, and taking squares, it becomes difficult to guess true ratings given partial sums only. Moreover, the parties exchange data for half of the items only.

The protocol proposed to estimate similarities consists of distributed scalar product computations in which privacy is achieved through HE, permutation, and random division at the same time. Paillier (1999) shows that HE is semantically secure for inference of input values. In other words, the parties cannot derive any information from the exchanged encrypted values. Moreover, utilizing permutation prevents the parties from learning the correct order of such encrypted values. However, they can guess their actual order with some probabilities. If there are h encrypted values, correctly guessing their order is 1 out of $h!$, where value of h depends on f, g, a_1, and number of commonly rated items. Random division also enhances privacy. Even if one sub-vector is determined, others are still private. And finally, as explained previously, the parties hold half of the similarity weights only for each item. Hence, the parties are not able to derive information about each other data while conducting the PACEP. During model construction performed off-line, the parties decide neighbors for each item based on the similarity weights they hold and the τ value. Since they do not exchange anything during model creation, nothing can be inferred.

Online phase includes how to estimate prediction based on the model and a's data. Recommendation estimation, as explained previously, consists of masking a's data like train data masking, using HE, permutation, and random division. Since the computations are similar to the ones conducted in the PACEP, the parties cannot obtain information about each other's data while offering predictions online due to the same reasons described previously. To sum up, the

proposed method allows data owners provide recommendations while preserving their privacy during both off-line online phases.

2.5. Supplementary Costs Analysis

Unlike off-line performance, online efficiency is critical for the overall success of CF systems. Since privacy and efficiency are conflicting goals, supplementary costs are inevitable due to privacy concerns. The proposed scheme is first analyzed in terms of additional online costs and then off-line costs are explained.

To evaluate computational complexity of CF schemes, online phase is much more vital than off-line phase because the algorithm must respond to the thousands of prediction requests in a few seconds. In the traditional item-based algorithm proposed by (Sarwar et al., 2001), online process includes prediction generation step, which involves only a table look up for the similarity values and the computation of the weighted sum. In the proposed scheme, since the model is partitioned between A and B, extra computations are inevitable. Due to random selection, filling, removing, division, and permutation, additional costs are negligible. However, due to encryption, computation costs increase. Suppose that A is the MP. It performs m_{aB} encryptions, where m_{aB} shows the number of z-scores sent to B. It also performs $m_{aB} \times w_{qjB}$ decryptions. To determine the running times of cryptographic algorithms, benchmarks for the CRYPTO++ toolkit from http://www.cryptopp.com/ can be used. Similarly, due to random division and HE, the number of exponentiations conducted by B increase by w_{qjB} times.

In order to estimate online times spent for providing a single prediction, experiments are performed using a computer, which is Intel Core2Duo, 2.4GHz with 4GB RAM. According to study conducted by Goldberg et al. (2001), it takes 350 ms for providing a single prediction using k-nn algorithm with k = 80. Item-based algorithm proposed by Sarwar et al. (2001) spends 237 ms for producing a single prediction. Without privacy concerns and communication overheads, item-based scheme on partitioned data is expected to spend about 237/2 ms for estimating a single recommendation due to parallel computations. In the proposed scheme, dominant supplementary costs are due to HE and decryption. Assuming that there are 20 commonly rated items involve in recommendation process during online phase (that number is usually less than 20) and each similarity weight is

divided into five random pieces, on average, there are 20/2 encryption and 5 × 10 = 50 decryptions. The benchmarks are used given above and it is determined that an encryption and decryption take 80 milliseconds. Thus, due to cryptographic computations, the parties spend about 2.4 seconds for each prediction during online phase. Compared to traditional CF schemes, 2.4 seconds for a single prediction can seem to be a large value. However, that number can be improved if e-commerce sites utilize enhanced computing machines because note that the computer used in the trials is Intel Core2Duo, 2.4GHz with 4GB RAM. Similarly, with rapidly evolving hardware, software, and information technologies, the online process can be realized in less time. The parties can also utilize parallel computation techniques to improve online time.

Like extra online computation costs, online communication costs are also expected to increase due to privacy concerns. In a traditional CF algorithm, *a* sends her rating vector and query to the system, which returns a prediction to *a*. Thus, number of communications is two only. In the proposed scheme, number of communications and amount of data to be transferred between *a* and the MP do not increase due to privacy measures. However, in the proposed scheme, the MP must communicate with the collaborating party in order to exchange data. Remember that the MP sends some encrypted z-scores and the query to the collaborating company, while it receives some encrypted values. Thus, since they perform two additional communications, online number of communications increases by two times. Without privacy concerns, when two parties want to collaborate, the MP sends *a*'s z-scores and the query to the collaborating party. Hence, amount of data to be transferred is about $(6 \times m_a + 2)$ bytes, where m_a represents number of *a*'s ratings, and it is assumed that four and two bytes are needed to store a z-score and its index, respectively. Thus, $6 \times m_a$ bytes are used for storing *a*'s z-scores and their indices; and two bytes for storing target item index. The collaborating party sends back to the MP two partial sums, which require eight bytes. Hence, amount of data to be transferred is about $(6 \times m_a + 10)$ bytes. In the proposed scheme, the MP fills $m_f = [(m - m_a) \times \theta_j]/100$ number of cells with fake ratings. After finding the z-scores, it removes those items' z-scores from *a*'s vector, which are *q*'s neighbors and held by itself. It then encrypts the

remaining z-scores. Thus, assuming that half of the z-scores are removed, on average, it encrypts $(m_a + m_f)/2$ number of z-scores. The size of the encrypted value produced by block cipher encryption can be computed as size of plain text + block size – (size of plain text mod block size) (Obviex, 2011). Assuming also that 16 bytes blocks or 128-bit key are used, for an encrypted value, there is need for 4 + 16 – (4 mod 16) = 16 bytes. Therefore, amount of data that the MP sends to the collaborating company is about $(m_a + m_f)/2 \times 16 + (m_a + m_f)/2 \times 2 + 2 = 9 \times (m_a + m_f) + 2$ bytes. The collaborating party B encrypts $(w_{qjB} + 1)$ values because remember that B divides similarity weights into w_{qjB} pieces and sends a partial sum to the MP A. Thus, amount of data to be sent by B to A is about $16 \times (w_{qjB} + 1)$ bytes. To sum up, amount of data to be transferred increases from $(6 \times m_a + 10)$ bytes to $(9 \times (m_a + m_f) + 16 \times w_{qjB} + 18)$ bytes.

Storage costs should also be analyzed. Storage overheads due to the privacy-preserving scheme are, as follows: Besides original user-item matrices, the parties need to keep filled and normalized user-item matrices (perturbed databases). Thus, additional storage costs due to such masked matrices are in the order of $O(nm)$. Similarly, the parties need to save default z-scores for all items to mask a's data. Therefore, extra storage costs due to them are in the order of $O(m)$. To sum up, due to privacy-preserving measures, like computation and communication overheads, extra storage costs are also expected.

Additional off-line costs are not that critical. Extra computation costs due to data masking can be considered negligible. However, if personalized ratings are used as fake ratings, supplementary costs are expected due to utilizing a traditional algorithm to estimate personalized ratings. In private mean and vector lengths computations, there are no extra computation costs. However, additional communication costs are inevitable. Although model generation does not introduce supplementary costs, due to PACEP, there are extra computation and communication costs. The parties perform encryptions, decryptions, and additional multiplications. Since they are conducted off-line, they do not affect the online performance. How often the model is updated is also vital for off-line performance. However, the model can be updated periodically and off-line

without interrupting online prediction process. In other words, the parties offer CF services to their customers using the existing model while updating in process.

2.6. Prediction Quality Analysis: Experiments

To examine how collaboration affects coverage and accuracy; and to demonstrate the effects of privacy-preserving measures on accuracy, several experiments are performed using real data sets.

Two different data sets, called MovieLens Public (MLP) and MLM, are utilized from GroupLens research community (GroupLens). Both data sets include personal taste evaluations on a range of movies. Users have rated the movies by selecting an integer between 1 and 5. In MLP, there are 100K ratings on 1,682 items of 943 users. MLM contains approximately one million ratings of approximately 3,592 movies made by 6,040 users. In the conducted experiments, the entire MLP data set is utilized, which has a density of 6.3%. However, from MLM data set, we first determined those users who rated at least 50 items. Their data is used in experimental trials, where density of the selected set is about 5.05%. Both sets can be considered as sparse data sets. Notice that the raw MLM set has 6,040 users who provided at least 20 ratings for about 3,592 movies. In order to obtain denser data set and improve the quality of the data, those users who rated less than 50 movies (about 950 users) filtered out. In such trials, available ratings are uniformly randomly divided into two disjoint sets, called training and testing. 80% of all ratings are used for item-based model construction (training), while the remaining votes are used for testing.

Examples of the most common criteria for CF like Mean Absolute Error (MAE) and Normalized Mean Absolute Error (NMAE) are used (Goldberg et al., 2001; Canny, 2002a) as evaluation criteria to evaluate the overall performance in terms of accuracy. MAE can be formulized, as follows:

$$MAE = \frac{1}{t}\sum_{i=1}^{t}\left|r_i - p_i\right|, \tag{2.11}$$

where t is the number of ratings in the test set, r_i and p_i are the original rating and the predicted output of the proposal, respectively. NMAE can be obtained by normalizing the MAE, as follows:

$$NMAE = \frac{1}{r_{\max} - r_{\min}} MAE, \tag{2.12}$$

where r_{max} and r_{min} are maximum and minimum boundary values of original ratings (five and one, respectively). We used NMAE in order to compare obtained results with the existing results, where data sets might include ratings from variable ranges. To scrutinize how collaboration affects coverage, the following metric is utilized,

$$Coverage = \frac{r_{res}}{r_{test}}, \tag{2.13}$$

where r_{res} and r_{test} stand for the number of predictions returned and the number of test ratings.

In order to demonstrate whether the improvements are statistically significant or they are occurred by chance, statistical *t*-tests are applied. *t* values are first computed on empirical results. A *p*-value is determined for each *t*-value from *t*-distribution table. If the *p*-value chosen for some significance level (α) (usually 0.05, 0.01, or 0.001) is less than the calculated *t* value, then it is concluded that the improvements are statistically significant and they are not happened by chance.

Experiment 1-Effects of supplementary ratings: Due to the nature of data partitioning and the proposed privacy-preserving scheme, some cells in user-item matrices might contain double ratings. The effects of such cases are first studied. In APD-based schemes, although it is assumed that the parties *A* and *B* hold disjoint sets of ratings, due to inserted fake ratings by both parties, they might fill the same cells or they may choose the filled cells by the other party to be filled. Thus, the parties may end up with cells with double ratings, which can be used for predictions. It is explored that the probability of having cells with double ratings and how that affects accuracy. Suppose that *A* and *B* have rating sets R_A and R_B, respectively, having values for disjoint cells. Let each rating set R_j having size of $n \times m$ and rating density of ρ_j, where *j* is *A* or *B*. Thus, the size of each rating set is $|R_j| = \frac{\rho_j}{100} nm$. According to disguising process, the parties

generate fake rating sets F_A and F_B, respectively. The number of fake ratings for each party j can be estimated as follows where θ_j is selected over the range $(0, \beta_j)$:

$$|F_j| = \frac{\theta_j}{100}(nm - |R_j|) = \frac{\mathrm{E}(\theta_j)}{100}(nm - \frac{\rho_j}{100}nm)$$
$$= \frac{\beta_j}{2\times 100}(nm - \frac{\rho_j}{100}nm) = \frac{nm}{200}\beta_j(1 - \frac{\rho_j}{100}) \quad (2.15)$$

where $\mathrm{E}(\theta_j)$ represents the expected value of θ_j, which equals $\beta_j/2$ due to uniform distribution over $(0, \beta_j)$. For any filled cell by A, the probability of being filled twice or being already filled cell (P_{oc}) can be estimated, as follows: $P_{oc} = \frac{|R_B| + |F_B|}{nm - |R_A|}$. Similarly, for A and B, the probability of having cells with double ratings (P_o) can be estimated, as follows: $P_o = \frac{|F_A|}{nm} \times \frac{|R_B| + |F_B|}{nm - R_A} + \frac{|F_B|}{nm} \times \frac{|R_A|}{nm - R_B}$. Although any rating is used only once, some cells might have two ratings, which are not necessarily the same. Thus, amount of ratings involved in prediction computations increases. Also note that fake ratings estimated from available data using a CF algorithm are most likely to represent users' true preferences.

Experiments are performed to show how varying β_j values affect number of cells with double ratings. MLP is used while varying β_j values from 3.125 to 100 to estimate P_o values. Trials are run for 100 times in which uniformly randomly selected different training sets are utilized for each experiment to make the results more statistically sound. The outcomes are displayed in Fig. 2.2. As seen from Fig. 2.2, P_o values increase with increasing β_j values. For smaller β_j values (less than 25), only about 4% of those cells with any ratings have double votes.

Trials are conducted to demonstrate how amount of such cells affect accuracy without privacy concerns. Both data sets are used. For MLM, 1,000 users' data is used. In order to determine fake ratings used to fill unrated cells, user, item, or overall mean votes, user distribution, and personalized ratings estimated from available data are utilized. β_j is set at 50. Predictions are first estimated on integrated data assuming that the cells with ratings contain single votes only

(Single Votes). Recommendations are then computed on integrated data, where some cells might have double ratings (Double Votes). The MAEs for both data sets are demonstrated in Table 2.1.

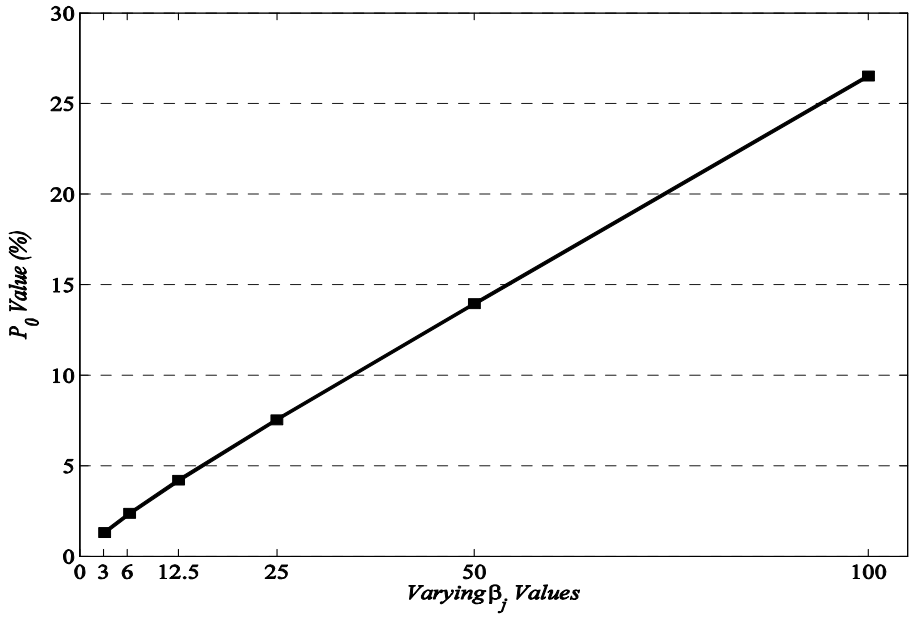

Figure 2.2. Percentages of Overlapped Cells (P_o) with Varying β_j Values

Accuracy changes due to supplementary votes for MLP are negligible, as seen from Table 2.1. For all methods used to determine fake ratings, the results are almost the same for both cases in experiments using MLP. Almost the same results are obtained for MLM. There are two exceptions. When user or item mean votes are used as fake ratings, allowing double votes improves accuracy. However, generally speaking, improvements in accuracy due to additional ratings can be considered insignificant for both data sets. As seen from Table 2.1, there are very little variations among methods and data sets. This phenomenon can be explained, as follows: First, as seen from Fig. 2.2, percentage of the overlapped ratings is about 14% when β_j at 50. Second, the outcomes are very similar for both data sets because they include movie ratings over the same range. Third, the results for single votes are slightly better for MLP due its somewhat higher density compared to the MLM set that is used. Forth, since density of the MLM set used in these trials is smaller, the improvements due to overlapping votes are larger for MLM. Finally, data normalization (using z-score normalization) smoothes the effects of different protocols used for data masking.

Table 2.1. Effects of Supplementary Ratings on Accuracy

Method for Fake Ratings		***User Mean***	***Item Mean***	***Overall Mean***	***Distribution***	***Personalized***
MLP	*Single Votes*	0.7331	0.7653	0.7336	0.7809	0.7508
	Double Votes	0.7333	0.7649	0.7330	0.7783	0.7507
MLM	*Single Votes*	0.7395	0.7869	0.7231	0.7741	0.7408
	Double Votes	0.7201	0.7510	0.7208	0.7670	0.7414

Experiment 2-Accuracy and coverage improvements due to collaboration: When data owners cooperate with each other, amount of ratings involved in prediction processes increases. That is why it is more likely to offer predictions for more items and provide high quality referrals. To verify how APD-based CF improves preciseness and coverage, experiments are performed. Such trials are run for 100 times while utilizing uniformly randomly selected different training sets for each experiment to make the results more statistically sound. Recall that *n* and *m* stands for number of users and items, respectively. First of all, accuracy changes are examined with varying *n* and *m* values during collaboration. For MLP data set, *m* is set at 1,682, while *n* is varied from 125 to 943. Similarly, for MLM data set, *m* is set at 3,591 and *n* is varied from 125 to 2,000. Similarly, *n* is set at 943 and 1,000 for MLP and MLM, respectively; and *m* is varied values. Predictions are first estimated for test data using partitioned data only. Recommendations are then produced for the same test data using the integrated data. After estimating overall averages, since the results are very similar, the outcomes are displayed for varying *m* values only in Table 2.2 for MLP while showing the results for varying *n* values only for MLM in Table 2.3.

Table 2.2. Accuracy Improvements due to Collaboration (MLP)

m		**200**	**400**	**800**	**1,682**
MAE	*Split*	**0.8662**	0.8223	0.7876	0.7620
	Integrated	**0.8168**	0.7782	0.7509	0.7337
Coverage	*Split*	**70.06**	78.04	83.16	86.40
	Integrated	**87.15**	93.27	97.17	98.22

It is hypothesized that coverage improves due to collaboration. To verify this, trials are conducted using both data sets while varying *n* and *m* values. Two criteria are defined to compute coverage values: (*i*) To estimate the similarity between two items, there must be at least two users who rated the both items and (*ii*) To provide prediction for *q*, *a* must provide ratings for at least two of the *q*'s neighbors. Coverage values are estimated for split data only and integrated data for both data sets with varying *n* and *m* values. Coverage values are displayed as percent in Table 2.2 and Table 2.3.

Table 2.3. Accuracy Improvements due to Collaboration (MLM)

n		**125**	**250**	**500**	**1,000**	**2,000**
MAE	*Split*	**0.7919**	0.7798	0.7625	0.7464	0.7318
	Integrated	**0.7689**	0.7553	0.7377	0.7240	0.7134
Coverage	*Split*	**56.48**	67.34	75.31	82.81	87.69
	Integrated	**67.15**	75.88	82.49	88.35	91.60

Experimental results show that collaboration between vendors definitely enhances both the quality of the recommendations and coverage, as seen from Table 2.2 and Table 2.3. Through partnership, amount of true ratings involved in prediction estimations increases. Similarities between various entities then can be estimated using more commonly rated entities. Similarly, number of neighbors joining in recommendation process online increases, as well. Thus, more truthful and dependable neighborhoods can be formed; and more precise and reliable referrals can be provided. For all varying values of *n* and *m* values, the results are better for integrated data than the outcomes on split data only, as expected. For MLP, when *m* is 200, accuracy improves by about 5.7% due to collaboration.

Such improvement is about 3.7% when m is 1,682. For the same cases, coverage values improve about 24% and 14%, respectively. The outcomes are similar for MLM, as seen from Table 2.3. When n is 125, accuracy enhances by about 2.9%. It is about 2.5% for n being 2.000. Due to collaboration, coverage enhances by about 21% when n is 125. Even if n is 2.000, coverage increases by about 4.5%. In terms of both coverage and preciseness, although the improvements become smaller with increasing n and m values, the outcomes on combined data still beat the results on split data only for both data sets. Relative improvements due to collaboration are significant and still promising for CF services. Therefore, collaboration is effective and vital for the success of recommendation systems. Empirical findings verify that collaboration contributes not only to the quality of the predictions but also query response rate.

Data collected for CF purposes might be unevenly partitioned between two parties. Hence, just to give an idea about if there is any benefit for the party with the larger portion of data, another set of experiments are conducted using MLP only, where entire data set is used. X_j is defined as the percentage of data held by the party j; and 1-X_j percent of the data held by the other party. X_j is varied from 10 to 100 in order to show how accuracy changes with varying amount of unevenly partitioned data. Note that when X_j is 100, it means that predictions are estimated on integrated data. The trials are run for 100 times, overall averages are computed, and the MAE values are displayed in Fig. 2.3.

As seen from Fig. 2.3, with increasing X_j values, benefits due to collaboration becomes smaller. Notice that with larger X_j values (bigger than 50), the outcomes become closer to the results on combined data because the results for X_j = 100 represent the outcomes on integrated data. Conversely, with decreasing amount of data, accuracy significantly becomes worse. Therefore, benefits due to collaboration are larger for those companies with smaller portion of data. Even if the results are close to the ones on integrated data for the company holding the larger portion of data, the party still benefit from collaboration. When comparing these results with the ones given in Table 2.7, it is observed that the company with the larger portion of data benefits from collaboration when X_j is smaller than 90 for small β_j values.

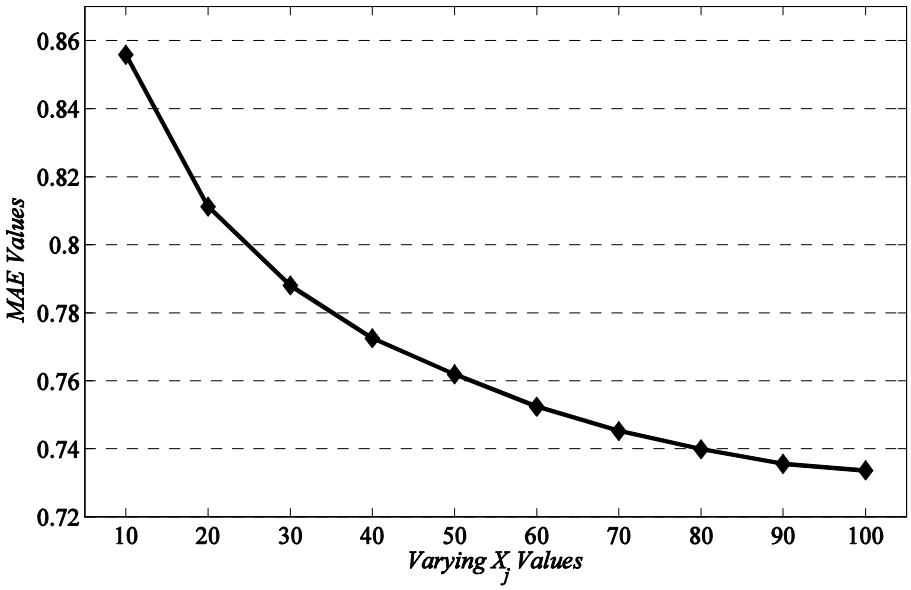

Figure 2.3. Effects of Unevenly Partitioned Data on Accuracy

Experiment 3-Methods for determining fake ratings: To observe how accuracy changes with different fake ratings, trials are performed using both sets. In these experiments, β_j is set at 50 and selected θ_j uniformly randomly over the range (1, 50). Data sets are used with sizes 943×1,682 and 1,000×3,591 for MLP and MLM, respectively. Since there are proposed three major methods (non-personalized ratings, ratings distribution, and personalized ratings) to find out fake ratings, they are used for determining the best one. The base results (results without privacy concerns) are first determined on integrated data and split data only. Predictions are then provided for the same test data based on masked data, which was disguised by filling fake ratings, estimated using various methods. Data disguising scheme is run for 100 times in order to make the outcomes more statistically sound and obtain results that are more dependable. After computing overall averages for both data sets, the outcomes are displayed in Table 2.4.

For MLP data set, user mean and overall mean methods slightly give better results compared to the outcomes on the integrated data, as seen from Table 2.4. Other methods make accuracy worse. Although personalized ratings method provides better results than the ones on split data only, accuracy losses are significant compared to the results on integrated data. The similar results are proposed for MLM. All methods, except overall mean method, make accuracy worse than the outcomes on integrated data. Since the overall mean method is the

only scheme, which provides enhanced outcomes for both data sets, it is chosen to determine fake ratings.

Table 2.4. Effects of Different Methods on Accuracy for Determining Fake Ratings

	No Masking		*Used Fake Rating Type*				
			Non-personalized Ratings			*Ratings Distribution*	*Personalized Ratings*
	Split Data	*Integrated Data*	*User Mean*	*Item Mean*	*Overall Mean*		
MLP	0.7620	0.7337	**0.7331**	0.7653	**0.7336**	0.7809	0.7508
MLM	0.7493	0.7276	0.7395	0.7869	**0.7231**	0.7741	0.7408

Experiment 4-Level of perturbation: Number of unrated cells to be filled is determined according to θ_j values, which are uniformly randomly chosen over the (1, β_j). With increasing β_j values, number of filled cells increases; that definitely affect accuracy. Although inserted votes increase the amount of data involved in prediction process, they may or may not represent users' true preferences. To assess the effects of varying β_j values, trials are done using both data sets, where overall mean method is utilized for determining fake ratings. Data sets are used with sizes 943×1,682 and 1,000×3,591 for MLP and MLM, respectively. Data disguising scheme is run for 100 times in order to make the outcomes more statistically sound and obtain results that are more dependable. After calculating overall averages, the MAE values are displayed in Table 2.5 for both data sets.

Table 2.5. Level of Perturbation vs. Accuracy

β	**0**	**3.125**	**6.25**	**12.5**	**25**	**50**	**100**
MLP	0.7337	0.7336	0.7334	**0.7333**	0.7335	0.7336	0.7336
MLM	0.7276	0.7230	0.7230	**0.7230**	0.7231	0.7231	0.7232

As seen from Table 2.5, accuracy slightly changes with varying β values for MLP. The quality of the referrals slightly becomes better with increasing β values from 0 to 25, while it becomes worse for larger β values. However, such changes are insignificant. For MLM, similar findings are obtained. Filling some of the unrated cells enhances accuracy. With increasing β values from 3.125 to 100,

accuracy slightly becomes worse. However, compared to the base result, the quality of the recommendations enhances faintly. Number of filled cells also affects overall performance because amount of data involved in the recommendation generation processes increases. Thus, 12.5 is selected as the optimum value of β for masking both data sets.

Like train data sets, active users' data are also masked utilizing the same scheme. The same methods can be utilized to find out fake z-scores or default z-scores. The similar experiments are conducted as determining the best method for estimating fake ratings to mask the train data. Since the similar outcomes are obtained, they are not shown. According to conducted experiments, personalized ratings method achieves the best results. Thus, that scheme is selected to produce default values for perturbing active users' data. Since accuracy is affected by varying perturbing levels, trials are also performed using both data sets while varying β values from 0 to 100. Data sets are similarly used with sizes 943×1,682 and 1,000×3,591 for MLP and MLM, respectively. The personalized ratings approach is utilized for finding fake z-scores. After running experiments 100 times, the overall averages of the MAEs are computed and displayed for both data sets in Table 2.6.

Table 2.6. Level of Perturbation vs. Accuracy (Masking *a*'s Data)

β	**0**	**3.125**	**6.25**	**12.5**	**25**	**50**	**100**
MLP	0.7337	**0.7397**	0.7430	0.7480	0.7553	0.7628	0.7699
MLM	0.7276	**0.7255**	0.7292	0.7329	0.7432	0.7502	0.7566

As seen from Table 2.6, the quality of the referrals becomes worse with increasing β values for both data sets. Although the results are worse than the base results for MLP, the outcomes are better than the base one for MLM even if they become worse with increasing β values. For MLP, accuracy losses due to inserted fake ratings are about less than 1% when β is set at 6.25. Since the best outcomes are obtained when β is 3.125, it is selected as the optimum value of β for disguising *a*'s data for both data sets.

Experiment 5-Overall performance: Some experiments finally performed to show the joint effects of the previously described factors. Both data

sets are used while varying *n* and *m* values. In the first set of experiments, *m* is fixed at 1,682 and 3,591 for MLP and MLM, respectively; and *n* values are varied. In the second set of trials, *n* is fixed at 943 and 1,000 for MLP and MLM, respectively; and *m* values are varied. Such trials are run for 100 times while utilizing uniformly randomly selected different training sets for each experiment to make the results more statistically sound. To mask train data sets using fake ratings, overall mean approach is utilized for finding such fake ratings, where we set β at 12.5. Similarly, in order to disguise *a*'s data, personalized votes scheme is employed for determining default values, where β equals 3.125. To obtain dependable outcomes, trials are conducted for 100 times. To compare the outcomes on split data only and the results on collaboration with privacy concerns, the MAEs for split data are displayed, as well. The outcomes of the first and the second sets of experiments are displayed as MAEs in Table 2.7 and Table 2.8 for both data sets, respectively.

Table 2.7. Overall Performance with Varying *n* Values

n		**125**	**250**	**500**	**943\1,000**
MLP	*Split*	0.8013	0.7901	0.7769	0.7620
	Proposed	0.8030	0.7766	0.7564	0.7380
	Gain (%)	-2.12	1.71	2.64	3.15
MLM	*Split*	0.7919	0.7798	0.7625	0.7464
	Proposed	0.7946	0.7883	0.7507	0.7221
	Gain (%)	-0.34	-1.09	1.55	3.26

Due to collaboration, accuracy improvements are expected, as shown previously. Since privacy and accuracy are conflicting goals, privacy-preserving measures might make accuracy worse. However, such losses should be small enough so that the gains due to partnership can compromise them. As seen from Table 2.7, proposed privacy-preserving scheme-based results are better than the outcomes on split data only for MLP when *n* is bigger than 125. Similarly, the proposed scheme provides improved results for MLM when *n* is larger than 250. It means that the improvements due to cooperation overweigh the downfalls on accuracy caused by privacy-preserving measures. With increasing *n* values,

improvements as a result of the proposed scheme usually increase for both data sets. When n is 1,000 for MLM, precision improves by 3.25%. For MLP, when n is 943, it enhances by 3.14%.

Table 2.8. Overall Performance with Varying m Values

m		**200**	**400**	**800**	**1,682\1,600**
MLP	*Split*	0.8662	0.8223	0.7876	0.7620
	Proposed	0.8233	0.7684	0.7468	0.7391
	Gain (%)	4.95	6.55	5.18	3.01
MLM	*Split*	0.9043	0.8497	0.8055	0.7703
	Proposed	0.8249	0.7822	0.7490	0.7452
	Gain (%)	8.78	7.94	7.01	3.26

The comparable results are obtained with varying m values, as seen from Table 2.8. Compared to the improvements with changing n values, enhancements with varying m values are more notable. For MLM, amount of improvements decreases with increasing m values. The trend is very similar for MLP, as well. However, the improvements reach their peak when m is 400; and the accuracy enhances by 6.55%. It picks up by 8.81% for MLM when m is 200. The significance of these improvements are also evaluated using the t-tests. Each of the improvements in Table 2.7 satisfies the t-test for $\alpha = 0.001$. For example, t-values are 20.84 and 16.39 for MLP and MLM, respectively, where n is 500. Hence, the proposed scheme significantly ensures the accuracy improvements for higher values of n. As seen from Table 2.8, for any number of items integrated data responses better results. By the t-tests, all the improvements are significant for α being 0.001. The t-values are 21.32 and 29.62 for MLP with m being 200 and for MLM with m being 1,600, respectively.

In order to show how overall performance change for larger values of β, another experiment is conducted with varying n and m values only using both data sets. We followed the same methodology and set β at 50. After estimating overall averages, the MAEs are displayed in Table 2.9 for varying n values only due to the similar trends.

Table 2.9. Overall Performance with Varying n Values for $\beta = 50$

n		**125**	**250**	**500**	**943\1,000**
MLP	*Split*	0.8228	0.8008	0.7791	0.7594
	Proposed	0.8116	0.7932	0.7860	0.7580
	Gain (%)	1.36	0.95	-0.89	0.18
MLM	*Split*	0.8132	0.8103	0.7737	0.7386
	Proposed	0.7861	0.7787	0.7794	0.7516
	Gain (%)	3.33	3.90	-0.74	-1.76

As seen from Table 2.9, accuracy changes due to privacy-preserving measures for MLP are smaller than the ones for MLM. For both data sets, accuracy slightly becomes worse when n is bigger than or equal to 500. Compared to the results for smaller β values displayed in Table 2.7, for n values less than 500, improvements are better for larger β value. However, they are worse for n values bigger than 250. Number of users is usually very large in the data sets constructed for CF purposes, utilizing larger β values makes improvements smaller. Since fake ratings might not represent the true users' preferences and increasing number of filled cells with fake ratings might damage quality of the data, data owners should select smaller β values. They also should consider privacy requirements discussed in Section 2.5.

We finally compared obtained results with other prediction methods in the state-of-the-art. For this purpose, NMAE measure is used. As seen from Table 2.7, the best outcome of proposed scheme in terms of NMAE is 0.1845 for MLP. Herlocker et al. (1999) propose a memory-based algorithm. Their scheme's NMAE value is around 0.1894. Another CF scheme proposed by Sarwar et al. (2000) utilizes dimensionality reduction methods. NMAE value achieved by that method is 0.1838. Bogdanova and Georgieva (2008) utilize error-correcting dependencies for CF and improve CF's performance. Their approach's performance in terms of NMAE is 0.1776. Lemire and Maclachlan (2005) propose a scheme based on average rating differential, which achieves the NMAE value of 0.1880. Compared to other approaches, this scheme provides promising and comparable results while preserving confidentiality.

2.7. Chapter Summary

A privacy-preserving scheme is proposed to offer recommendations on APD between two parties. This method can be considered as hybrid, combining model generation off-line and estimating referrals online. The proposed scheme consists of various protocols, which are explained in detail and justified about satisfaction over privacy constraints. The proposed method makes it possible to produce predictions on partitioned data between two parties. The vendors are able to offer richer and better CF services on integrated data without jeopardizing privacy. Although supplementary costs are inevitable caused by privacy concerns, such costs are negligible. Since some works are performed off-line (model generation) and the off-line costs are not that critical, additional off-line costs do not affect overall performance. In this chapter, it is also demonstrated that the scheme is secure and prevents the vendors from deriving the true ratings and the rated items held by each other. Real data-based experiments are conducted to evaluate the method in terms of accuracy. Attained experimental results show that collaboration significantly improves precision. Although privacy concerns cause accuracy losses, they are overweighed by the gains as a result of partnership. Therefore, the proposed scheme can be used to provide accurate recommendations efficiently while preserving privacy.

Optimum values of various controlling parameters are usually determined experimentally. According to empirical results in Section 2.6, the optimum values of β_j (and θ_j) are determined. These results based on MLP and MLM can be generalized. Like quality of the predictions, privacy level and preserving data originality are other factors that affect the choice of controlling parameters' values.

3. PRIVACY-PRESERVING TRUST-BASED COLLABORATIVE FILTERING ON ARBITRARILY PARTITIONED DATA

Trust issues discussed in the context of CF; and trust-based algorithms are shown to be successful for CF schemes. This study examines how to estimate trust-based predictions on APD while guaranteeing privacy of the enterprise data. For such CF problem, a privacy-preserving solution requiring plausible amount of resources and responding online in decent time is proposed. The proposal is also analyzed through dimensions of privacy preservation and extra resource usage. Moreover, to observe effects of APD on accuracy and coverage; and how the proposed privacy-preservation method affects prediction quality, a couple of experiments are performed. All conducted analyses demonstrate that the proposed scheme efficiently performs trust-based CF on APD while preserving confidentiality.

3.1. Introduction

Trust is so popular in social networks (Fogel and Nehmad, 2009) and also takes attention of e-commerce researches (Zhang et al., 2007). Trust metrics are applied into CF algorithms and satisfactory results are obtained (Hwang and Chen, 2007; Massa and Avesani, 2007). Massa and Bhattacharjee (2004) show that trust concept can be applied for CF recommender systems. In their model, trust values between users are determined based on *web of trust* composed of directly specification by users. To increase the coverage of trust-based CF system, Massa and Avesani (2007) offer *propagated trust metric*. Hwang and Chen (2007) present a CF method deriving both direct and propagated trust values from traditional rating profile data. They experimentally demonstrate that rating-based trust approach gives predictions having better accuracy over correlation-based CF methods. In this study, it is preferred to use the method proposed by Hwang and Chen (2007) due to the ease of availability of rating profile data.

In this chapter, it is investigated how to provide trust-based CF services on APD while protecting privacy of any party's data to the other. As stated before, focused APD configuration is non-overlapping. Privacy guarantee is main factor making parties cooperate, accuracy is key parameter to measure CF output

quality, and efficiency is core requirement for online responding information systems. At the same time, they are conflicting goals. Despite of such conflicting goals, a computationally achievable solution providing privacy requirements and responding quality predictions is going to be proposed. Such solution is also justified in privacy, supplementary overheads, and prediction quality via analyzing theoretically and empirically.

There are some privacy-preserving CF schemes caring about trust metrics. Dokoohaki et al. (2010) investigate optimal privacy in trust-aware social networks using randomized disguising techniques as a preprocessing step. While their scheme is a solution for such networks in P2P manner, in this proposal, there are two parties whose data constitute APD. Kaleli and Polat (2011) examine how to provide trust-based recommendations on VPD and present a solution in this variant. This work differs from theirs in data partitioning configuration and considers APD scenario for trust-based CF mechanism.

3.2. Trust-based Collaborative Filtering

Hwang and Chen (2007) define trust between users *a* and *u*, $t_{a\to u}$, which means how much *a* trusts *u*, or vice versa. The trust can be computed, as follows:

$$t_{a\to u} = \frac{1}{\left|I_a \cap I_u\right|} \sum_{j\in(I_a\cap I_u)} \left(1 - \frac{\left|p_{aj}^u - v_{aj}\right|}{\rho}\right), \tag{3.1}$$

where I_a and I_u stands for the rated item set of users *a* and *u*, respectively, and ρ is the range of the operated ratings, p_{aj}^u is prediction for trust computation and it can also be derived, as follows:

$$p_{aj}^u = \overline{v_a} + (v_{uj} - \overline{v_u}), \tag{3.2}$$

where v_{uj} is the rating of item *j* given by *u*, $\overline{v_a}$ and $\overline{v_u}$ are mean ratings of users *a* and *u*, respectively. Hwang and Chen (2007) also introduce trust propagation metric in order to evaluate trust values between users who have no commonly rated items, as shown in Eq. (3.3):

$$t_{s\xrightarrow[v]{}h} = t_{s\to v} \oplus t_{v\to h} = \frac{\left|I_s \cap I_v\right| \cdot t_{s\to v} + \left|I_v \cap I_h\right| \cdot t_{v\to h}}{\left|I_s \cap I_v\right| + \left|I_v \cap I_h\right|}, \tag{3.3}$$

where users s and h are non-commonly rated users but v has co-rated items with both of them. The final inferred trust $t_{s\to h}$ is the average of the values for each user v computed by Eq. (3.3). After computing trust between users, prediction p_{aq} for a for target item q can be computed, as follows:

$$p_{aq} = \overline{v_a} + \frac{\sum_{u\in S}(v_{uq} - \overline{v_u}) \times t_{a\to u}}{\sum_{u\in S} t_{a\to u}}, \tag{3.4}$$

where S stands for the users have rated q and in trust neighborhood of a, i.e. a's mostly trusted users.

3.3. Trust-based Predictions on Arbitrarily Partitioned Data with Privacy

In this section, the scheme proposed for privacy-preserving trust-based CF on APD is presented. The proposal consists of four different sub-processes. At first, preprocessing is performed to determine user means and normalize data held by each party. Secondly, secure trust computation process is done covering users having commonly rated items. Then, for user pairs having no commonly rated item, trust propagation computation is taken place. Finally, it is proposed how predictions are estimated online over the constructed models.

3.3.1. Preprocessing

To normalize user ratings using deviation from mean normalization method and compute prediction for trust values, the parties need user mean values. User or row mean can be expressed as *sum* over *count* of user ratings. Rather than directly sharing of such sum and count values for each user, it is more convenient to exchange such values after filling some unrated cells of original data set with default ratings v_d so that the original sum and count values are preserved. In default filling scheme, v_ds can be determined alternatively, as follows:

a. Using POP algorithm (Goldberg et al., 2001), for each item, each party computes v_d as average from the ratings available for that item.

b. As row-variant of the previous method, for each user, each party computes v_d as average from the ratings available for that user.

c. Overall mean of the available data can be utilized.

The other factor required to be defined for default filling scheme is θ_j as the amount of unrated cells to be filled. θ_j should be selected from the range (0,

β_j], where β_j is upper bound parameter of filling amount. Before filling process, each party should agree on specific β_j, which should be dependent on the density d_j of the available data set by each party j. Considering the originality of data, it can be offered that the maximum β_j as value of 100 correspond to the number of rated cells available by each party. After such default filling scheme, they can share sum and count values computed over own filled database and normalize such own database. Then, each party carries on the remaining tasks over their own filled databases.

3.3.2. Secure Trust Computation

Before computing trust values, Eq. (3.1) can be rearranged by using Eq. (3.2) and some simplifications, as follows:

$$t_{a\to u} = \frac{1}{c_{au}}\left(c_{au} - \sum_{j\in(I_a\cap I_u)} \frac{\left|(\overline{v_a} - v_{aj}) + (v_{uj} - \overline{v_u})\right|}{\rho} \right) \tag{3.5}$$

in which $c_{au} = \left|I_a \cap I_u\right|$. Considering APD;

i. ρ is obviously public. Moreover, c can be considered public because the data is filled and default ratings are included. In the following, trust computation scheme, c_{au} can be derived from the context.

ii. *Absolute difference* - $\left|(\overline{v_a} - v_{aj}) + (v_{uj} - \overline{v_u})\right|$: Recall that to compute trust value between users a and u, they must rate at least one item j commonly. If this condition is satisfied, there are two cases of availability of the ratings for item j. In the first case, *full availability* occurs as both ratings are available either in A or B. Second case is *cross-wise availability* in which one of the ratings is in A and the other is in B. There is no problem for determination of commonly rated ones and computation of this expression in one-side full availability case. However, cross-wise available ratings make trust computation task challenging. To compute trust values privately on APD, secure trust computation protocol (STCP) is introduced in the following. Considering Eq. (3.1) and Eq. (3.3), it can be said that trust values between two users are symmetric, i.e $t_{a\to u} = t_{u\to a}$ causing that trust values between users constitute upper-half triangular

matrix. It is proposed that such triangular trust matrix would be vertically shared among two parties. Note that, *half* of trust values in STCP stand for plausible and possible number of trust values providing the acute vertical division with respect to the number of users in APD.

Secure Trust Computation Protocol (STCP)

I. For the first half of the trust values

For each distinct user pair (a, u)

1. A and B compute and store absolute differences for fully available ratings own-side.
2. A and B continue ignoring set of ratings handled in S*tep* 1.
3. A encrypts all available $(\overline{v_a} - v_{aj})$ values using HE with its public key KA and obtains $\xi_{KA}(v_{aA})$ and generates X-1 random vectors and hides the vector holding the rated item indices of v_{aA} into such random vectors, then send $\xi_{KA}(v_{aA})$ and all X vectors to B.
4. B encrypts all available $(v_{uj} - \overline{v_u})$ values using HE with KA and obtains $\xi_{KA}(v_{uB})$.
5. B also performs $\xi_{KA}(v_{aA} + v_{uB}) = \xi_{KA}(v_{aA}) \times \xi_{KA}(v_{uB})$ for only commonly corresponding item indices for each X vectors.
6. For each different vector, B permutes each obtained values using its private permutation function f_B. Then, it sends all permuted values to A.
7. Using OT protocol, A takes the permuted set holding actual rated ones.
8. A decrypts them, takes absolute values and accumulates them. A also adds initially found absolute differences for fully available ones. A now has the *half-trust* values.
9. Switching their roles, applying *Step* 3-8, B also has the *complementary half-trust* values.
10. B sends such complementary values to A that obtains the final trust values.

II. For the remaining half, switching their roles, they perform *Step* 1-10 and B obtains final trust values.

3.3.3. Trust Propagation Computation

After performing the STCP, each party ends up with its corresponding trust values. However, some of them are null because the absence of co-rated items among any two users. Thus, they must utilize trust propagation metric given in Eq. (3.3) to determine trust values for such users. After establishing which of the values are needed for trust propagation calculation, the parties inform each other about such values. Then, for each required value, they exchange local aggregated numerator and denominator values of Eq. (3.3). They sum up such values and obtain propagated trust values. By the way, they get rid of the null values and update related trust values via Eq. (3.3).

3.3.4. Prediction Generation

To generate a prediction for a on q, Eq. (3.4) must be considered. The parties can follow the similar steps given in Section 3.3.2. The ratings of item q are held by the parties A and B can be labeled as v_{uqA} and v_{uqB}, respectively. Based on such arbitrarily distributed ratings and vertically distributed trust values, they can generate prediction using Prediction Generation Protocol (PGP) explained in the following. Assume that A acts as MP from whom a requests a prediction for q. Note that as in the STCP, A and B perform computations with fully available components and store such *fully available sub-aggregates* just after being informed about a and q and go on computations ignoring them unless specifying their contributions to PGP.

Prediction Generation Protocol

0. Active user a asks a prediction about q from A.
1. A zeroes all trust values below the threshold τ.
2. Using its own public key and self-blinding property, A encrypts all available trust values of a and sends them to B.
3. B multiplies just the rated ones of v_{uqB} using homomorphic property. B accumulates the results for numerator and it also accumulates corresponding trust values for denominator. It obtains $\xi_{KA}\left(\sum_{u\in S}(v_{uqB}-\overline{v_u})\times[t_{a\to u}]_A\right)$ and $\xi_{KA}\left(\sum_{u\in S}[t_{a\to u}]_A\right)$, respectively, where $[t_{a\to u}]_A$ is trust value of a held by A.

4. Switching their roles, they perform *Step* 1-3 for v_{uqA} and trust values of a held by B. A obtains $\xi_{KB}\left(\sum_{u\in S}(v_{uqA}-\overline{v_u})\times[t_{a\to u}]_B\right)$ and $\xi_{KB}\left(\sum_{u\in S}[t_{a\to u}]_B\right)$. After encrypting fully available sub-aggregates with KB, A adds corresponding parts of numerator and denominator and sends them to B.
5. B decrypts the obtained values in *Step* 4 and encrypts them with KA. Then, it also encrypts its fully available sub-aggregates similarly.
6. B adds correspondent obtained values in *Step* 3 and *Step* 5.
7. B returns $\xi_{KA}\left(\sum_{u\in S}(v_{uq}-\overline{v_u})\times t_{a\to u}\right)$ and $\xi_{KA}\left(\sum_{u\in S}t_{a\to u}\right)$ to A.
8. A determines p_{aq} using Eq. (3.4).

3.4. Privacy Analysis

Since the proposed scheme does not involve any direct exchange of information about individual rating values and which items are rated, principal privacy constraint is satisfied. However, there are some protocols prescribing change of aggregate values in secure manner. Such transactions should be examined whether they conflicts auxiliary privacy constraint or not. The proposed protocols' privacy protection is based on default ratings and cryptographic tools. Since Paillier (1999) justifies that his HE schemes are semantically secure and Naor and Pinkas (2001) examine the security of their OT protocols, the proposed protocols are secure in their anticipated framework. However, in privacy perspective, it is still interesting to investigate disclosed intermediary values, aggregates, and default ratings in addition to actual rating values. Considering such values, the proposed scheme is going to be analyzed in terms of inference probability rates and privacy enhancement.

In normalization, default ratings hide the total number of ratings of each user has already rated and avoid directly sharing actual local mean of each user. In the STCP, B can guess a subset of A's rated items. Let the size of this subset f, over random vectors, the probability of guessing such subset is $1/X$ at *Step 3*. Similarly, after switching their roles, A can also guess it similarly. The value of X should be set to proper value depending on sensitivity of items and privacy requirements. Again, in the same protocol, A obtains individual aggregates of commonly rated items for the subset of cross-wise components in *Step 8*. Let A

obtains *g* pieces of such aggregates. Then, *A* can infer the subset of the rated items with probability C_g^f. By switching their roles, *B* may also infer in the same possibility for the complementary cross-wise components.

In trust propagation, each party learns which trust value is null in the other party's trust sub-matrix and two sub-aggregate values; one for numerator and the other for denominator part of Eq. (3.3). In order to deduce trust values owned by the other party, how many values are included to compute such sub-aggregates should be known. However, it is unknown in the proposed scheme. After guessing such value, they are still conundrums that which trust values are included and what are such trust values. In prediction generation, *cooperator*, who is not *MP*, learns only sub-aggregates of final prediction value and *MP* learns only final prediction value. For both parties, the same applies as in trust propagation because there are sub-aggregates in similar computational manner.

Filling with default ratings and removing processed fully available components enhances privacy. Both operations decrease the rate of original rating components over totally contributed ones to generate aggregate values. Default ratings also give denial of possession of the rated items in case of inferring the other's ratings. To compare privacy preservation with respect to type of default ratings, POP algorithm can be considered to be the best over the others. Since the computation process is realized user by user and user-based aggregates are shared through the proposed protocols, using item mean for default ratings, each user rating vector is expected to have different default rating values. However, in user mean usage, filled default ratings are the same for each user and this may facilitate the inference manner of the other party. Also, row-variant POP preference is not suitable for applications, where local user means are sensitive about privacy because the actual mean is disclosed each other. Overall mean's handicap about privacy is that it is the same for all local data. This fact is also advantage to the party intending to deduce some extra sensitive information from the other's data. Despite such issues, the proposed default rating types can be still efficient solution for filling.

Conducted privacy analysis indicates that there is no conflict of both principal and auxiliary privacy constraints in the proposed approach. There are no

direct or indirect leakage of the parties' individual rating values and the rated items. However, the inference possibilities are scrutinized over the shared intermediate values. One additional issue is related to trust updates. To enhance privacy and complicate inference possibilities, the parties prefer to re-fill their original data with defaults for each update phase because default rating values and filled unrated cells are changed and different input data are obtained.

3.5. Supplementary Costs Analysis

The proposed scheme brings about some overheads in computation, communication, and storage. In this section, the proposal is examined in terms of such extra costs. First of all, considering computational resources, this scheme can be contemplated for implementation as two phases: off-line and online. Preprocessing and computations of direct and propagated trust values can be computed off-line. However, prediction generation needs online interactions and it should be considered for running online. Since off-line costs are not critical, PGP must be evaluated in terms of computational efficiency. While the party j realizes totally $n/2$ encryptions, rf_j homomorphic multiplications, rf_j homomorphic additions, and 2 decryptions, the cooperator party performs additional two decryptions and two homomorphic additions, where rf_j stands for number of rating and filled ratings in a's vector held by the party j. Since τ can be determined previously, the parties can encrypt trust values after comparing them with τ and store the encrypted trust values off-line. However, this brings $n^2/4$ storage requirement for each party. To benchmark cryptographic operations, CRYPTO++ (2009) can be referred.

Secondly, the proposed method bids parties to communicate for trust computation and prediction generation. In the STCP, there must be at least two communications consisting of significant sizes of data exchange in bi-directional way while trust propagation requires two mutual communications; one for informing which of the held trusts are null and the other for sharing the sub-aggregates for propagated trusts. During the PGP, there must be at least three communications. Recall that if two parties collaborate on APD with off-line generated trust values, two online communications are needed to provide CF services.

Thirdly, there are also storage overheads with respect to off-line model generation. During off-line phase, the parties temporarily need spaces to keep default ratings, user mean values, and c_{au}. However, trust values computed off-line requires $n^2/4$ spaces from each party in order to utilize the constructed CF prediction model online. Note that, depending on data entry traffic and recommended product profiles, trust model must be updated in a particular period.

3.6. Prediction Quality Analysis: Experiments

Using MLP, the proposal is empirically evaluated in terms of its accuracy and coverage. This data set consists of 100,000 ratings collected from 943 users on 1,682 movies. While ratings are integers from 1 (dislike) to 5 (like), each user has rated at least 20 movies. 80% and 20% of available ratings are randomly selected for training and testing, respectively. While training subset is used as input data for specified CF process, test ratings are queried for prediction. The returned prediction values are compared based on the accuracy metric MAE. Another metric used for evaluation is coverage given in Eq. (2.13). To reach more dependable results, each experiment is performed 100 times and overall averages are presented.

Hwang and Chen (2007) evaluate experiments in which the scheme determines *a*'s neighborhood selecting the best *k* similar users. However, rather than such determination, it is preferred to use threshold-based scheme in order to simplify prediction generation process. Herlocker et al. (1999) empirically demonstrate that such process can be performed either of both methods. To determine the optimum value of the threshold τ, various experiments were conducted using MLP. According to the outcomes, it is concluded that 0.7 produces satisfactory accuracy and coverage values. Thus, τ is set at 0.7 in the following trials.

In the first experiment, it is investigated how collaboration on APD affects accuracy and coverage of trust-based CF system. For this reason, an experiment is conducted comparing split and combined data without any privacy considerations. The number of users in input data is varied and MAE and coverage values are computed. The results are given in Table 3.1. Both accuracy and coverage gains show similar manner with respect to increasing number of users. Such gains are

initially higher; however, with joining more users into CF process, they decrease in deceleration. APD generally contributes more significant to accuracy rather than coverage according to results in Table 3.1. This experiment shows that APD contributes more to the prediction quality of CF system when amount of available rating profile is lower.

Table 3.1. Effects of APD on Accuracy

MAE				
Type	***n* = 125**	**250**	**500**	**943**
Split	0.8196	0.7935	0.7730	0.7631
Integrated	0.7803	0.7621	0.7498	0.7446
Gain (%)	4.80	3.96	3.00	2.42

Table 3.2. Effects of APD on Coverage

Coverage (%)				
Type	***n* = 125**	**250**	**500**	**943**
Split	91.51	95.91	98.01	98.97
Integrated	95.99	97.74	98.71	99.14
Gain (%)	4.90	1.91	0.71	0.17

After justifying how APD contributes prediction accuracy and coverage of CF system, how default ratings affect accuracy is analyzed. By setting n to 500, for different levels of filling, i.e. β_j and different types of default ratings, accuracy changes are observed on combined data. The obtained results are given in Fig. 3.1. According to outcomes given in the figure, it can be said that accuracy is inversely proportional to filling level for all types of default ratings. However, considering MAE value for split data for n = 500 in Table 3.1, which is 0.7730, accuracy for all types and levels in the proposed model is more accurate. To speak about specific types of default ratings, for $\beta_j \leq 50$, the best default rating type is overall mean. However, after such value, for overall mean-based default filling, accuracy significantly becomes worse.

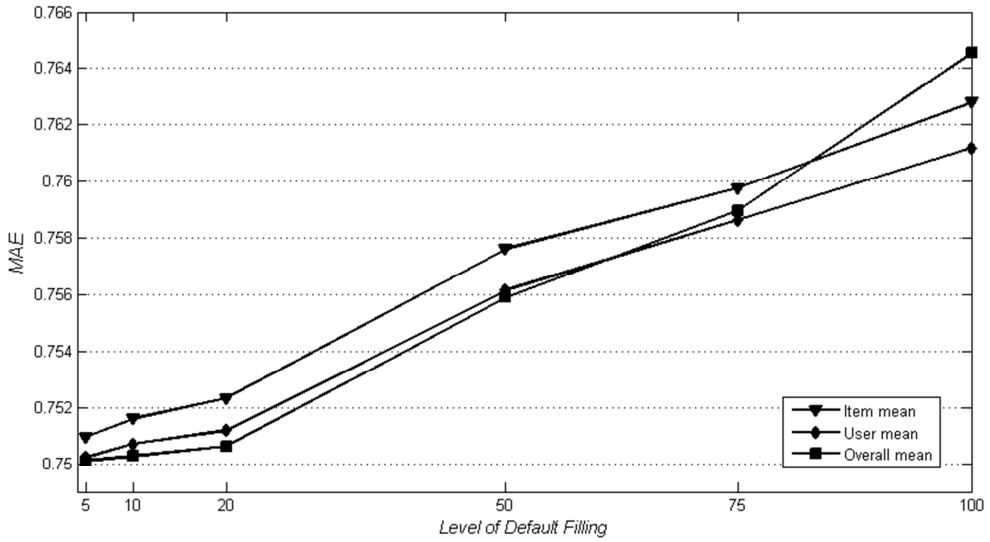

Figure 3.1. Accuracy vs. Level of Default Filling

There is a parallel relation in terms of accuracy between filling with item and user mean default ratings. However, user mean shows better accuracy as seen from Fig. 3.1. For the values of θ_d higher than 100, both schemes achieve very similar results. The outcomes become very closer to each other.

In the final experiment, the goal is to benchmark the accuracy values obtained by using the split data only and the combined data by the proposed method. For this purpose, trials are conducted for different level of filling with respect to varying number of users. Considering average number rated items per user is 106 in MLP, if β_j = 10 then E(θ_j)= 5 and E(|*fc*|)= 0.05×106 = 5.3, where E(x) is expected value of x and |*fc*| stands for the number of filled cells. Roughly speaking, it is expected about 5 of unrated cells would be filled with default ratings. Similarly, for β_j = 20 and 50, E(|*fc*|) values are 10.6 and 26.5, respectively. Since such listed E(|*fc*|) values can be considered decent values providing balance between privacy and data originality, for β_j being 10, 20, and 50 with the best filling scheme, i.e. overall mean, for such values trials are performed and the results are displayed in Fig. 3.2. According to such figure, it is obvious that for all n and focused β_j values, the proposed methods outperform the results on split data only. Especially for smaller number of users, the proposed scheme provides more quality referrals due to the insufficient amount of ratings in partitioned case. These outcomes show that the proposed scheme is preferable in order to overcome problems caused by split data.

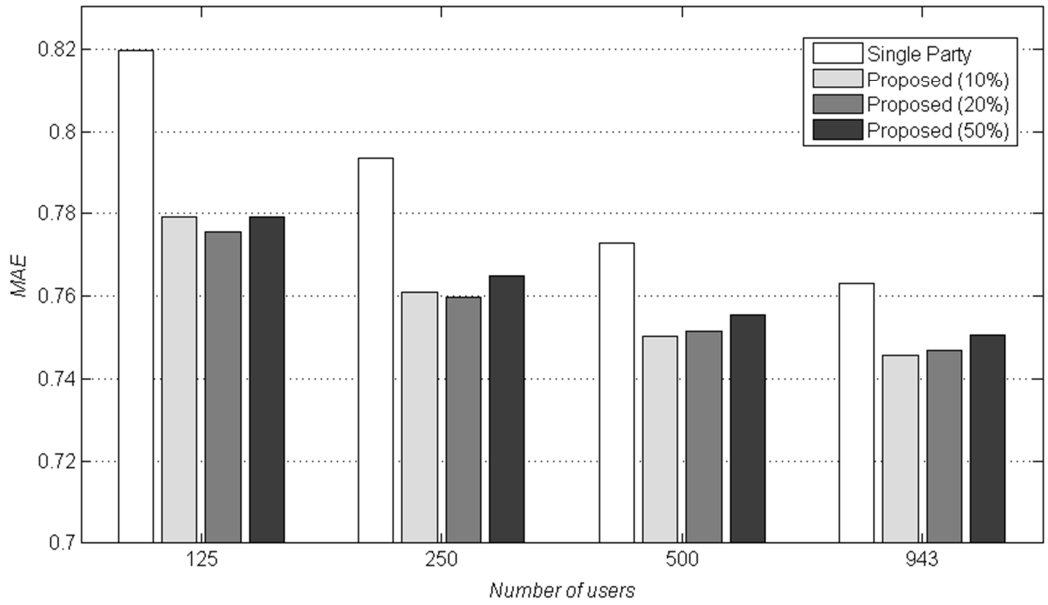

Figure 3.2. Single Party vs. the Proposed Method

3.7. Chapter Summary

In this chapter, in order to provide trust-based predictions on APD with privacy, a solution is presented. This solution makes it possible for two parties to provide predictions using their joint data without divulging their sensitive data to each other. The proposed scheme gives control of some parameters to the collaborating parties. The solution is justified in terms of efficiency, privacy-preservation, and accuracy through theoretical and experimental analyses. The experimental analyses demonstrate that the solutions produce satisfactory results in prediction quality especially in situations where available data are insufficient.

4. PRIVACY-PRESERVING NAÏVE BAYESIAN CLASSIFIER-BASED COLLABORATIVE FILTERING ON ARBITRARILY PARTITIONED DATA

In order to eliminate privacy, financial, and legal concerns of those companies having inadequate data and want to provide recommendations on combined data, a privacy-preserving scheme is proposed to estimate naïve Bayesian classifier-based predictions on arbitrarily partitioned data between two parties. This method is intended to help online vendors provide binary ratings-based predictions on partitioned data without violating their corporate privacy requirements. It is shown that the proposed scheme is secure and able to offer recommendations efficiently. Conducted real data-based experiments demonstrate that collaboration is vital for better services; and accuracy losses due to privacy measures can be suppressed by the gains due to collaboration. Thus, the proposed method is preferable for estimating accurate predictions efficiently on partitioned data while preserving data holders' privacy over the scheme on split data only.

4.1. Introduction

Users' preferences about different products can be represented either using numeric ratings or binary votes. In some applications, it is preferable to know whether a user likes an item or not, rather than knowing how much she likes it. If a customer likes an item, the related rating is represented as *like* (1). Similarly, if she dislikes it, the vote is then *dislike* (0). If users' preferences about various products are represented by binary ratings, NBC can be used to predict whether a will like the target item q or not. Hence, a scheme is proposed to show how to find NBC-based predictions on APD with confidentiality.

NBC is a supervised learning method based on the probabilistic model of Bayes theorem. With many attributes, it is very hard and time consuming to calculate the Bayes probabilities. However, naïve assumption facilitates that by stating these attributes are conditionally independent. Moreover, NBC is robust against isolated noise points and irrelevant attributes (Amatriain et al., 2011). NBC has been approved of successful solution for many real world applications from text classification (Youn and Jeong, 2009) to medicine (Geenen et al., 2011).

Due to its popularity, it takes attention for recommendation system infrastructure, as well. Due to the ease of availability over numerical ratings, binary ratings-based CF mechanisms, like NBC-based scheme, take special attention of recommender systems research community. In case of binary ratings, traditional numerical ratings-based CF schemes cannot be used to provide recommendations from such binary ones. Miyahara and Pazzani (2000) apply NBC to CF to offer binary rating-based referrals. They show that NBC-based scheme provide comparable results with correlation-based schemes. Su and Khoshgoftaar (2006) point out that although NBC-based scheme performs worse in terms of accuracy, it offers better scalability comparing to correlation-based methods. In this work, NBC-based scheme is scrutinized to examine the feasibility of providing CF services on APD. Additionally, binary rating-based CF promises relatively higher availability of input preference data either explicitly or implicitly.

Privacy-preserving schemes proposed for providing recommendations should achieve privacy, accuracy, and efficiency. Thus, the proposed scheme should meet the principal and auxiliary privacy constraints, be able to provide predictions with decent accuracy (accuracy losses due to privacy should be compensated by gains due to collaboration), and not cause too much additional costs (online costs do not prevent the scheme from estimating many predictions in restricted time intervals). Additional computation, communication, and memory overheads are inevitable due to privacy. They should be small enough to still estimate predictions efficiently. Online time requirements are much more rigid than off-line ones. Supplementary online costs should be minimized as much as possible. Therefore, the problem that is desired to be solved can be briefly defined, as follows: *How to provide NBC-based recommendations with decent accuracy on APD efficiently while achieving the collaborating parties' corporate privacy?*

4.2. NBC-based Collaborative Filtering

When users express their opinions about the products they bought before or showed interest using binary ratings, CF process becomes a classification. In other words, it is predicted whether a will like q or not. NBC is a popular and widely used classification algorithm. It is simple, easy to understand, and achieves

comparable classification accuracy. Due to naïve assumption, its performance becomes better. It is utilized in various applications for classifying unlabeled samples (Lee, 2006). NBC-based CF scheme works, as follows (Miyahara and Pazzani, 2000):

1. Let **D** be training set of users and their related ratings for various items. Each user is represented by an *m*-dimensional rating vector, $V = (v_1, v_2, \ldots, v_m)$, depicting ratings for *m* products including null ratings.
2. The algorithm operates on two classes, *like* and *dislike*. Given an active user *a*, the classifier predicts that *a*'s rating for *q* belongs to the class having the higher probability.
3. The naïve assumption states that features or ratings are independent, given the class label. Thus, the probability of the target item *q* for *a* belonging to $class_j$ (C_j), where *j* is *like* or *dislike*, given *a*'s *m* rating values, can be written, as follows:

$$p(C_j \mid v_1, v_2, ...v_m) \quad \alpha \quad p(C_j) \prod_{h=1}^{m} p(v_h \mid C_j) \tag{4.1}$$

4. In Eq. (4.1), prior, $p(C_j)$, and conditional probabilities, $p(v_h|C_j)$, can be estimated from training data, where v_h represents the feature value or rating of item *h* for *a*. To assign a target item *q* to a class, the probability of each class is computed; and the example is assigned to the class with the higher probability.

NBC also takes attention of P2D2M community. Vaidya et al. (2008b) examine the similar problem for the case of VPD; and offer NBC scheme on VPD with privacy. Keshavamurthy et al. (2010) propose secure multi-party computation-based approach with trusted third party to compute the aggregate class instances for VPD using the probabilistic model of classifier such as naïve Bayes technique. In another study (Skarkala et al., 2011), privacy-preserving tree augmented NBC is offered for statistical databases that are horizontally partitioned. In their method, privacy is ensured by multi-candidate election scheme.

NBC-based CF algorithms are examined considering privacy-related issues. Kaleli and Polat (2007b) present solutions for achieving private referrals

on NBC using randomized response techniques. Their solution makes it possible for servers to collect private data without greatly compromising users' privacy. Their scheme is based on a central server. In another study (Kaleli and Polat, 2009), the authors further investigate how to improve privacy-preserving NBC-based CF systems' online performance by grouping users in various clusters. Their proposal makes it sufficient to realize computation on a few subsets of input data rather than dealing with full of available data. Bilge and Polat (2010) offer a preprocessing scheme to promote efficiency and accuracy of NBC-based CF scheme with privacy. On the contrary of all these schemes, which are based on central data, this proposal is based on partitioned data between two parties, where data partitioning is arbitrary. Although there is an NBC-based CF solution on partitioned data between two parties without greatly exposing their privacy (Kaleli and Polat, 2007a), the authors consider HPD and VPD. Their method helps data holders produce binary ratings-based predictions on HPD or VPD without deeply jeopardizing their confidentiality. Unlike their approach, a scheme is going to be proposed when data is partitioned arbitrarily. Kaleli and Polat (2010) also suggest an NBC-based CF solution for P2P environments. While in their proposal the transactions are realized between peers on individual data, in the proposed scheme, data holders collaborate on a set of users' data.

4.3. NBC-based Predictions on Arbitrarily Partitioned Data with Privacy

When users' ratings are held by a single company, it is easy to estimate NBC-based predictions without considering privacy concerns. On the other hand, when users' ratings including the active users are arbitrarily partitioned between two parties, it is challenging to offer NBC-based referrals efficiently without jeopardizing data owners' confidentiality. Due to privacy concerns, online performance and accuracy become worse. In order to improve online efficiency, collaborating parties prefer performing as much computations as possible off-line. The companies can construct the NBC-based prediction model off-line using the proposed secure protocols satisfying predefined constraints and implications. After constructing such model, they then serve their customers demanding CF services online. Thus, the proposal is explained in terms of off-line and online phases in the following.

4.3.1. Off-line Phase: Model Generation

The model consists of two arguments, the likelihood $P(V \mid C_j)$ and the priori $P(C_j)$ probabilities. Thus, the parties *A* and *B* need to compute these probabilities off-line while preserving their confidentiality. In this section, how the parties can estimate the required probabilities in a secure manner using the proposed protocols is explained.

Estimating the Likelihood Probabilities: Any active user *a* can ask a prediction for any item. Hence, the target item *q* can be one of *m* items. Moreover, *a* might rate any individual item. Thus, the parties need to consider every possible rating (1 or 0) for any item. Suppose that *H* and *Q* represent the rating sets for an item *h* and a target item *q*, respectively. Also, let H_A and Q_A; and H_B and Q_B be rating sets for items *h* and *q* held by *A* and *B*, respectively. Due to arbitrary partitioning, $H = (H_A \parallel H_B)$ and $Q = (Q_A \parallel Q_B)$, while assuming that the parties hold disjoint sets of ratings. Considering arbitrary partitioning, for any rating v_h, the likelihood can be formulated, as follows:

$$P(v_h \mid C_j) = \frac{\eta}{\delta} = \frac{\eta_{AA} + \eta_{AB} + \eta_{BA} + \eta_{BB}}{\delta_{AA} + \delta_{AB} + \delta_{BA} + \delta_{BB}} \tag{4.2}$$

in which η_{AA} represents number of users whose ratings for *h* and *q* (notice that the rating for *q* represents the class label) are held by *A* and their ratings for item *h* satisfy the condition, η_{AB} represents number of users whose ratings for *h* and *q* are held by *A* and *B*, respectively; and their ratings for item *h* satisfy the condition, and so on. Similarly, δ_{AA} represents number of users whose class label is C_j and rated both *h* and *q*; and such ratings are held by *A*, δ_{AB} represents number of users whose class label is C_j and rated both *h* and *q*; and the ratings for *h* and *q* are held by *A* and *B*, respectively, and so on. Due to arbitrary partitioning between two parties *A* and *B*, four possible cases are considered in order to estimate numerator and denominator. Also note that since users' preferences are represented using binary ratings, the parties need to estimate $P(v_h \mid C_j)$ probabilities for four possible cases (v_h = 1 or 0 and C_j = 1 or 0).

Due to the availability of data, *A* can compute pure components, η_{AA} and δ_{AA}, by itself without the help of *B*. Similarly, *B* can calculate η_{BB} and δ_{BB} by itself without the help of *A*. However, there are crosswise components like η_{AB}, η_{BA}, δ_{AB},

and δ_{BA}, which must be computed collaboratively using the integrated data. Considering self-computable and cooperative-computable components, to calculate η and δ values securely, Numerator of Likelihood Computation Protocol (NLCP) and Denominator of Likelihood Computation Protocol (DLCP) are proposed respectively in which DLCP is the variant of NLCP. The parties estimate η securely using the NLCP, as follows:

Numerator of Likelihood Computation Protocol

For each possible four cases (v_h = 1 or 0 & C_j = 1 or 0; notice that v_h and C_j can take two different values)

For each item pairs ((h, q) | h = 1, 2, …, m; q = 2, 3, …, m; $h < q$)

i. If $v_h = 0$, then A inverts the vector H_A (converts 1s into 0s and 0s into 1s).

ii. If $C_j = 0$, then B inverts the vector Q_B (converts 1s into 0s and 0s into 1s).

iii. A and B fill empty cells with zero in H_A and Q_B, respectively.

iv. A encrypts each value in H_A using an HE scheme with its public key KA; and obtains ξ_{KA} (H_A) so that it prevents B from learning such values.

v. A then generates a random vector, R_A, which is necessary for self-blinding.

vi. Using self-blinding property of HE, A computes ξ_{KA} (H_A, R_A) = ξ_{KA} (H_A) × $(R_A)^N$.

vii. A finally sends ξ_{KA} (H_A, R_A) to B. Due to encryption and self-blinding, B cannot derive useful information from such encrypted values.

viii. Using HE property, B performs ξ_{KA} (H_A, R_A)^{Q_B}; and finds ξ_{KA} (H_A × Q_B).

ix. B performs element wise homomorphic addition of ξ_{KA} (H_A × Q_B); and finds ξ_{KA} (η_{AB}).

x. B sends it to A, who can decrypt it using its related private key, which is known by it only; and obtains η_{AB}.

xi. The parties switch their roles and perform the same steps to find η_{BA}.

xii. A and B finds $\eta_A = \eta_{AA} + \eta_{AB}$ and $\eta_B = \eta_{BB} + \eta_{BA}$, respectively; and exchange them.

xiii. Each part finally computes $\eta = \eta_A + \eta_B$, which is required for the numerator of Eq. (4.2).

In the NLCP, the goal is to count the corresponding cross-wise elements, which are required to compute the numerator of Eq. (4.2). Notice that Eq. (4.2) estimates $P(v_h| C_j)$ values. In the first and the second steps, if v_h and/or C_j values are equal to zero, then the related vectors are inverted. In the third step, the unrated cells are simply ignored by assigning them zero values. In the following steps up to the Step x, using HE with self-blinding property, the parties achieve secure multiplication of plaintexts over distinguished ciphertexts. In the remaining steps, they perform complementary computations and exchanges. They finally complete the protocol by having the numerator, i.e η, of $P(v_h \mid C_j)$ in Eq. (4.2).

To estimate likelihood probabilities using Eq. (4.2), in addition to the numerator, the parties need to compute the denominator. For the denominator part, the parties need to determine those users whose class label is known (rated q) and they also rated the related item h. As stated before, A and B can estimate δ_{AA} and δ_{BB} by themselves without needing each other. However, they need to collaborate to estimate δ_{AB} and δ_{BA} values. The DLCP is offered to compute δ values without violating A and B's privacy, as follows:

Denominator of Likelihood Computation Protocol

For each possible four cases (v_h = 1 or 0 & C_j = 1 or 0)

For each item pairs $((h, q) \mid h = 1, 2, \ldots, m; q = 2, 3, \ldots, m; h < q)$

i. If $v_h = 0$, then A inverts the vector H_A (converts 1s into 0s and 0s into 1s).

ii. B replaces each rating in vector Q_B by 1.

iii. A and B fill empty cells with zero in H_A and Q_B, respectively.

iv. A encrypts each value in H_A using an HE scheme with its public key KA; and obtains $\xi_{KA}(H_A)$ so that it prevents B from learning such values.

v. A then generates a random vector, Z_A, which is necessary for self-blinding.

vi. Using self-blinding property of HE, A computes $\xi_{KA}(H_A, Z_A) = \xi_{KA}(H_A) \times (Z_A)^N$.

vii. A finally sends $\xi_{KA}(H_A, Z_A)$ to B. Due to encryption and self-blinding, B cannot derive useful information from such encrypted values.

viii. Using HE property, B performs $\xi_{KA}(H_A, Z_A)^{Q_B}$; and finds $\xi_{KA}(H_A \times Q_B)$.

ix. B performs element wise homomorphic addition of $\xi_{KA}(H_A \times Q_B)$; and finds $\xi_{KA}(\delta_{AB})$.

x. B sends it to A, who can decrypt it using its related private key because it is known by it only; and obtains δ_{AB}.

xi. The parties switch their roles and perform the same steps to find δ_{BA}.

xii. A and B finds $\delta_A = \delta_{AA} + \delta_{AB}$ and $\delta_B = \delta_{BB} + \delta_{BA}$, respectively; and exchange them.

xiv. Each part finally computes $\delta = \delta_A + \delta_B$, which is required for the denominator of Eq. (4.2).

The goal of the DLCP is to count the satisfying values for the denominator of $P(v_h \mid C_j)$ in Eq. (4.2). The protocol follows the similar steps and ideas like the NLCP. It is actually based on the same mathematical and cryptographic property. However, the DLCP counts the number of commonly rated items providing that the first value v_h equals the specific value of either 1 or 0. In the first step, the value of the v_h is considered. If its value is zero, then we invert the vector. In the second step, the rated cells are taken consideration for the following computations. The remaining steps flow similarly like in the NLCP. At the end of the protocol, the required value for the denominator, i.e δ, of $P(v_h \mid C_j)$ is obtained. Once they completed the abovementioned protocols, the parties have the values for numerator and denominator parts (η and δ values, respectively). They can now estimate all necessary likelihood probabilities, $P(v_h \mid C_j)$, using Eq. (4.2). In order to complete the model, in addition to estimating $P(v_h \mid C_j)$ values, A and B must compute priori probabilities, $P(C_j)$ values.

Priori computation: Priori computation includes determining the probability of having 1 or 0 for the target item q. The class probabilities for both classes should be estimated based on q's ratings. In other words, priori

computation is performed by just determining probability of selecting a rating belonging to class C_j from a set of ratings for an item vector because the target item can be one of m items. It can be calculated for ratings arbitrarily partitioned between A and B, as follows:

$$P(C_j) = \frac{|C_{j,Q}|}{|Q|} = \frac{|C_{j,Q_A}| + |C_{j,Q_B}|}{|Q_A| + |Q_B|}. \quad (4.3)$$

Since the target item can be any item, the parties need to find out priori probabilities for all items off-line. To do so, they need to exchange the partial aggregates in Eq. (4.3). However, exchanging such values directly infers how many users liked or disliked an item. To prevent the parties from deriving such information and enhance privacy, we propose to mask target item's rating vector using randomized perturbation techniques. The parties first perturb the target item's rating vector and then estimate partial aggregates in Eq. (4.3) based on masked data. They finally can exchange such aggregates and find priori probabilities. To determine priori values, Privately Priori Estimation Protocol (PPEP) is proposed which is given as follows:

Privately Priori Estimation Protocol

I. A uniformly randomly selects half of the items (m_A items).

II. For each item $i = 1, 2, \ldots, m_A$

a. A determines the rating density (d_i) and the number of unrated cells (NR_i).

b. A uniformly randomly chooses θ_i over the range $(0, d_i]$. Notice that number of filled cells is associated the density.

c. A then uniformly randomly selects $m_{Ai} = (\theta_i \times \mathrm{NR_i})/100$ number of unrated cells to fill with default or fake ratings.

d. A finally fills them with pre-determined default of fake ratings, where B does not know the default or fake ratings.

e. A now finds $|C_{j,Q_A}|$ values for both classes and $|Q_A|$ values; and sends them to B.

f. B determines $|C_{j,Q_B}|$ values for both classes and $|Q_B|$ values; and computes $P(C_j)$ values for both classes using Eq. (3)

g. B saves such priori probabilities for m_A items

III. For the remaining m_B items ($m = m_A + m_B$), the parties switch their roles and perform the same steps to find the priori probabilities for m_B items

The PPEP aims to fill randomly selected subset of the unrated cells in each item vector in order to hide actual numbers of *like* and *dislike* values. Sometimes it might be more damaging for revealing such values. Thus, each party joins such filling process for randomly chosen half of items, as pointed in the step I. In the step II, each item is taken into process; and m_{Ai} pieces of the unrated cells are filled with default or fake ratings. Notice that the amount of cells and which cells to be filled are determined based on density. Then, *A* computes the local aggregates over such filled data and sends them to *B*. After obtaining final priories for such half of items, *B* stores them. In the step III, priories are computed for the remaining subset of items in the same manner with swapping duties. At the end of the protocol, each party ends up complementary subset of all estimated priori values.

In order to disguise each item vector in a decent way, the following issues are presented that should be considered by the parties:

1. *Density-based determination of amount of filling*: They must consider the density of item rating vectors in order to determine how many unrated cells should be filled with predetermined default or fake ratings.
2. Number of cells to be filled with 1 or 0 can be determined, as follows:
    a. The parties can fill selected unrated cells in such a way so that like/dislike ratio is preserved as much as possible to approximate priori probability to original value.
    b. The parties determine default ratings and fill the cells with the corresponding ones.
3. Default or fake ratings can be determined, as follows:
    a. Randomly
    b. Non-personalized ratings on available data
        i. Item mode
        ii. User mode
        iii. Overall mode

c. Personalized ratings on available data estimated using NBC-based scheme

At the end of off-line phase, the parties now own the model, which can be used to offer NBC-based recommendations online.

4.3.2. Online Phase: Recommendation Estimation

Online phase is triggered by a who asks a prediction (p_{aq}) for q from MP. One of the parties acts as an MP interacting with a. Assume that A acts as an MP. Also note that a's ratings are arbitrarily partitioned between A and B. MP first needs to estimate probabilities for both classes using Eq. (4.1). It then assigns the item into the class with the higher probability. Such probabilities can be estimated in a distributive manner, as follows:

$$p(C_j \mid v_1, v_2, ...v_m) \quad \alpha \quad p(C_j) \prod_{h=1}^{M_A} p(v_h \mid C_j) \prod_{h=1}^{M_B} p(v_h \mid C_j), \tag{4.4}$$

where M_A and M_B show the number of a's ratings held by A and B, respectively.

Proposed *Online Recommendation Estimation Protocol* (*OREP*) works, as follows:

1. A informs B about q.
2. If q is one of the m_A items, A computes $\prod_{h=1}^{M_A} p(v_h \mid C_j)$ and B calculates $p(C_j) \prod_{h=1}^{M_B} p(v_h \mid C_j)$ values for both classes based on a's data held by A and B, respectively.
3. If q is one of the m_B items, A computes $p(C_j) \prod_{h=1}^{M_A} p(v_h \mid C_j)$ and B calculates $\prod_{h=1}^{M_B} p(v_h \mid C_j)$ values for both classes based on a's data held by A and B, respectively.
4. B sends such aggregates for both classes to A.
5. A computes the final probabilities for both classes using Eq. (4.4) and compares them to determine the q's class label for a.
6. It finally assign the item to the class with the higher probability and returns the class label to a.

4.4. Privacy Analysis

The proposed scheme is examined in terms of privacy based on the lemma and its proof below. Recall that each party is *semi-honest*; thus, tries to guess information as much as they can while obeying the rules of the game. Remember also that a chain is only as strong as its weakest link. Adopted analogy assumes that the chain preserves privacy while the weakest link is the maximum inference probability.

Lemma: The proposal satisfies the auxiliary privacy constraint at lower bound of 1out of $\left(\begin{matrix}|C_{v_h,H_A}|\\ \eta_{AB}\end{matrix}\right)$.

Proof: There are some information exchanges during the proposed protocols conducted in off-line and online phases. In the NLCP, the parties share encrypted vectors of items with each other, where each party obtains one self-crosswise component; and final numerator and denominator values are obtained by both companies.

i. *Encrypted data exchange*: Paillier (1999) shows that the HE scheme is semantically secure; and it also supports self-blinding property, which diversifies cipher-texts of the same plaintexts by addition of particular random values.

ii. *Self-crosswise components* (η_{AB} or η_{BA}): This can be examined in terms of A. If A knows η_{AB}, for A, the probability of guessing those users who rated q as C_j held by B is 1 out of of $\left(\begin{matrix}|C_{v_h,H_A}|\\ \eta_{AB}\end{matrix}\right)$. Similarly, if A knows δ_{AB}, it can guess those users who rated q as C_j held by B with probability of 1 out of $\left(\begin{matrix}|H_A|\\ \delta_{AB}\end{matrix}\right)$. In general, due to the characteristics of the rating data, the first combination value is expected to be smaller than the latter one. Therefore, the probability $1/\left(\begin{matrix}|C_{v_h,H_A}|\\ \eta_{AB}\end{matrix}\right)$ is larger inference probability value. Since the protocol is symmetric, this inference possibility also applies for B.

iii. Final numerator and denominator values: Considering the guessing process in terms of A and recalling Eq. (4.2), it is much harder to derive information from exchanged aggregates required for numerator and denominator. Since the self-computable values (known by B only) are added to collaborative-computable values, A cannot infer useful information from δ_B values. Again, due to the symmetric property, B cannot learn useful information from δ_A.

For priori computation, after disguising process, each party shares the amount of liked and disliked ratings of each item. Since θ is uniformly randomly chosen from the interval of $(0, \rho_q]$, its expected value is $\rho_q/2$. It should be remarked that the density (ρ_q) is unknown to A; and first of all, A must guess it correctly. After guessing it correctly, the problem is determining what the value of θ is because it is randomly selected. The most rational probability of guessing its value is 1 out of E[θ]. After A guess ρ_q and θ, it can find the number of filled cells. Even if it figures out such information, due to data sparsity, it cannot learn the actual filled cells (those users who really rated q and held by B) from knowing the number of rated and filled cells. However, it can guess it with probability of 1 out of $\binom{n_u}{q_B}$ in which $n_u = n$ – number of users rated q and held by A; and q_B shows the number of users who rated q and held by B. Notice that q_B can be estimated after guessing the density and θ.

At the final stage (in the online prediction computation), the helping party returns two aggregate values for each class. Primarily, it is not clear to MP or A that how many of the conditional probabilities are used for determining these aggregated values. Note that such aggregates are multiplication of some and yet unknown number of conditional probabilities. Also note that MP can act as an active user in multiple scenarios to derive conditional probabilities. To eliminate such attack, during each transaction, the parties can fill some of a's rating vector with default ratings or remove a few ratings from her rating vector. Moreover, they update their model periodically.

Corollary: Since there is no direct exchange of the actual ratings and the rated items, principal privacy constraints are satisfied and their guarantee directly

depends on the satisfaction of auxiliary privacy constraint. The proposal satisfies the principal privacy constraint at the lower bound of 1 out of $\left(\frac{\left| C_{v_h, H_A} \right|}{\eta_{AB}} \right)$ because it satisfies the auxiliary privacy constraint at the same lower bound.

4.5. Supplementary Costs Analysis

Collaboration over partitioned data and privacy measures bring supplementary costs. In this section, the overheads are highlighted in terms of computation, communication, and storage costs. Compared to online costs, off-line costs are not that critical for overall performance. Hence, additional online costs should be as small as possible for overall performance of the proposed scheme.

Supplementary storage costs can be explained, as follows: Additional costs are expected due to saving the model. Each party stores likelihood probabilities and half of priori probabilities. Due to storing likelihood probabilities, each party needs four $m \times m$ matrices. Similarly, each party needs one $1 \times m/2$ matrices for storing priori probabilities. Hence, additional storage costs due to the proposed scheme are in the order of $O(m^2)$.

The proposed scheme also causes some additional communication costs. Since online costs are more important, extra online communication costs are first explained. In a traditional CF system, there are two communications only. In the proposed scheme, on the other hand, there are four communications due to data exchange between collaborating parties. Hence, online communication costs increases two times. During off-line phase, the parties perform three protocols. Due to such protocols, number of off-line communications is in the order of $O(m)$, assuming that they exchange $1 \times m$ vectors. Although additional off-line communication costs are significant, they do not affect online performance and they are acceptable as long as they are done off-line.

Additional online computation costs are also important for overall performance. Compared to central server-based scheme, proposed scheme does not introduce extra online computation costs. In other words, the scheme performs the same number of multiplications, additions, and comparisons with a traditional NBC-based scheme on integrated data held by a single company. Unlike online phase, off-line phase causes significant extra computation costs. Additional

computation costs performed off-line can be explained, as follows: Additional costs due to encryptions, decryptions, multiplications, and exponentiations dominate off-line computation costs. Compared to such operations, additional costs due to random number generation, filling, addition, and so on are negligible. Number of encryptions, decryptions, multiplications, and exponentiations conducted off-line are in the order of $O(mn)$, $O(m^2)$, $O(mn)$, and $O(m^2n)$, respectively. In order to find out the running times of cryptographic functions, benchmarks are available at the CRYPTO++ toolkit (CRYPTO++, 2009). Off-line phase should be performed periodically to update the model by inserting new data. The periods of model updating depends on data flow, applications, and system administration. To speed up update process, modified elements only should be considered in each update. In a traditional user- and NBC-based CF (Miyahara and Pazzani, 2000), online prediction can be processed in the order of $O(mn)$. Hence, additional costs introduced during off-line phase can be considered acceptable.

4.6. Prediction Quality Analysis: Experiments

In this section, the proposed scheme is evaluated empirically to observe prediction quality using real data sets. Two different data sets, namely MLP and MLM are used, which have been already used in previous chapters. Recall that in both data sets, ratings are integer value from dislike to like as a range of [1, 5]. In order to perform NBC-based CF tasks on these data sets, the integer ratings are performed to binary ones with respect to threshold value of 4. If any rating value is greater than or equal to 4, then it is assigned to class 1, or 0 otherwise. The generic experimentation method is to divide already selected data into training and testing components in the percentages of 80 and 20, respectively. During data division process, ratings are randomly categorized for training or testing sets.

In order to benchmark the prediction qualities, two different evaluation metrics are utilized: classification accuracy (CA) and F-measure (F1). While CA is derived by obtaining the percentage of correctly predicted ratings, to calculate F1, precision and recall values must be evaluated. F1 is the harmonic mean of precision and recall, as follows:

$$F1 = 2\frac{Precision \times Recall}{Precision + Recall} \tag{4.5}$$

In the first experiment, trials are conducted to verify the hypothesis that APD contributes the prediction quality of the data owners. It is desired to show how CA and F1 values improve due to collaboration. Trials are performed using MLP for comparing split and combined data while varying *n* from 125 to 943. To reach reliable outcomes, the experiments are repeated for 100 times; and in each case, the train users are randomly selected from the input data. After computing CA and F1 values for split and integrated data, they are displayed in Table 4.1. How much gain is provided by APD with respect to split data is also computed. According to results presented in Table 4.1, first of all, the APD-contribution hypothesis is correct for all cases and both metrics; because all gain values are conspicuously larger. From these results, second deduction issue is that combined data provide better gain for both CA and F1 metrics for lower *n* values. Table 4.1 also shows that F1 gain is less than CA gain for all cases. Hence, it can be said that APD-based data combination contributes more to the parties having fewer amounts of data. Online vendors prefer providing recommendations on their integrated data to offer enhanced outcomes.

Table 4.1. Effects of Collaboration on Accuracy (MLP)

n		**125**	**250**	**500**	**943**
Split	CA	0.6069	0.6307	0.6511	0.6645
	F1	0.6697	0.6883	0.7047	0.7151
Integrated	CA	0.6356	0.6555	0.6696	0.6776
	F1	0.6907	0.7070	0.7179	0.7239
APD Gain (%)	CA	4.73	3.93	2.85	1.98
	F1	3.13	2.71	1.87	1.23
***t*-values**	CA	14.98	20.11	24.09	31.25
	F1	6.07	8.52	10.54	20.07

To justify the empirical results given in Table 4.1 in terms of the improvements, some *t*-tests are conducted and the results are also given in Table 4.1. According to *t*-values and *t*-table, it can be said that the improvements are

statistically significant for the confidence level of 99.9%. Notice that the *t*-values are inversely proportional to the APD gain percentages. While improvements in accuracy increase, statistical significance level reduces. The reason for this phenomenon can be explained with data scarcity. If the amount of input data becomes lesser, then fluctuations in accuracy should be expected to rise.

After ensuring about APD-based prediction quality gain, accuracy losses due to privacy-preservation must be investigated, too. In online phase, there are no operations affecting prediction quality. Considering off-line phase, since likelihood computation is just based on encryption, which does not perform modifications on the content of data or say plaintext, it does not affect the output accuracy. However, during priori probability computation, rather than encryption-based operations, this scheme proposes randomization-based operations, which perturb the original data. Thus, some losses in accuracy can be expected while hypothesizing that these changes do not crucially affect the prediction quality. To check whether this hypothesis holds or not, experimental trials are carried out using MLP data set. In these trials, it is anticipated to compare the outcomes produced on the originally combined data and that on disguised data with respect to varying number of users. The results are given in Table 4.2.

Table 4.2. Effects of Privacy-preservation Measures on Prediction Quality (MLP)

n		**125**	**250**	**500**	**943**
Original	CA	0.635594	0.655465	0.669643	0.677621
	F1	0.690713	0.706966	0.717855	0.723857
Disguised	CA	0.635804	0.655533	0.669716	0.677657
	F1	0.690849	0.706986	0.717878	0.723880
Variation (%)	CA	0.033	0.010	0.011	0.005
	F1	0.020	0.003	0.003	0.003

According to outcomes given in Table 4.2, it can be observed that random disguise procedure's effects on accuracy are so small, even that can be neglected. In order to differentiate the outcomes, the results are displayed in higher precision to the six-digit, the percentage of variation at most 0.03%. Proposed disguising policy bounds this variation since random values are added while like/dislike ratio

is preserved. Considering all findings in this section, prediction qualities are improved due to APD while privacy-preservation of proposed scheme does not affect the prediction quality.

Experiments are also performed using MLM data set to show how collaboration and the proposed privacy-preserving schemes affect accuracy. The outcomes are summarized in Table 4.3. Improvements due to collaboration (*APD gain*) are derived by comparing the results of split and combined cases while *ultimate gain* values were determined by considering split data only and collaboration over proposed method. These results on MLM verify the first hypothesis. In other words, APD contributes prediction quality in the same manner as MLP.

Table 4.3. Effects of Collaboration and Privacy-preservation on Prediction Quality (MLM)

n		**125**	**250**	**500**	**1,000**	**2,000**
Split	CA	0.622681	0.646837	0.664281	0.677421	0.685142
	F1	0.695649	0.711724	0.725984	0.736373	0.742484
Integrated	CA	0.646436	0.665044	0.675931	0.68288	0.686343
	F1	0.713083	0.725903	0.735649	0.741808	0.745324
Proposed	CA	0.646543	0.665115	0.675959	0.682899	0.686349
	F1	0.713157	0.725938	0.735670	0.741821	0.745327
APD Gain (%)	CA	3.8150	2.8148	1.7538	0.8059	0.1754
	F1	2.5062	1.9922	1.3312	0.7382	0.3825
Ultimate Gain (%)	CA	3.8321	2.8257	1.7580	0.8087	0.1762
	F1	2.5167	1.9971	1.3342	0.7399	0.3829

Rather than looking variation between combined and disguised cases as in Table 4.2, ultimate gains promised by proposed method are put forward in Table 4.3. Note that such values are in percentages and there are negligible differences, starting from 10^{-4} in ratio, between APD gains and ultimate gains among different numbers of users. According to results depicted in Table 4.3, it can be concluded that proposed privacy-preserving scheme contributes CA and F1 values ultimately over single party. Using *t*-test analysis, it is also checked whether such ultimate gain is statistically significant or not. *t*-values are computed for ultimate gains of

both CA and F1 metrics as shown in Table 4.4. According to empirical outcomes in Table 4.4, it can be said that obtained ultimate gains are statistically significant at the confidence level of 99.9%.

Obtained empirical outcomes show that collaboration between two parties definitely improves accuracy in terms of both CA and F1 measure. Moreover, recommendations estimated from integrated data can be considered more reliable than the ones computed from split data only. Due to accuracy improvements, online vendors, even the competing ones, want to collaborate. If they provide more accurate and dependable predictions, then more customers prefer their sites to trade. When they have privacy, financial, and legal concerns, they can utilize the proposed scheme to eliminate such concerns. Due to proposed randomization-based data masking, accuracy losses are expected. However, as seen from obtained experimental outcomes, such losses are very small and can be neglected. Accuracy gains due to collaboration definitely compensate the accuracy losses due to privacy-preserving measures. In conclusion, online vendors can use the proposed scheme to provide accurate referrals while preserving their confidentiality.

Table 4.4. Statistical Significances of Ultimate Gains (MLM)

n		**125**	**250**	**500**	**1,000**	**2,000**
CA	Ultimate Gain (%)	3.8321	2.8257	1.7580	0.8087	0.1762
	t-value	16.01	17.41	15.55	11.67	3.95
F1	Ultimate Gain (%)	2.5167	1.9971	1.3342	0.7399	0.3829
	t-value	7.43	8.59	8.48	7.34	5.18

4.7. Chapter Summary

In this study, a privacy-preserving naïve Bayesian classifier-based collaborative filtering scheme is proposed for two parties ending up with arbitrarily partitioned binary data. Such scheme can be a solution for e-commerce sites suffering from insufficient data for recommendation services. It is both theoretically and empirically proved that proposed scheme provides practical balance between the conflicting goals of privacy, efficiency, and accuracy. Not only it promotes to the accuracy of collaborative filtering service providers, but also it ensures their

privacy. Moreover, it demands decent amount of online computation, communication, and storage overheads.

5. PRIVACY-PRESERVING HYBRID COLLABORATIVE FILTERING ON CROSS PARTITIONED DATA

In this chapter, it is investigated how to offer hybrid CF-based referrals with decent accuracy on CPD between two e-commerce sites while maintaining their privacy. The proposed schemes should prevent data holders from learning true ratings and rated items held by each other while still allowing them to provide accurate CF services efficiently. Real data experiments are performed to evaluate the proposals in terms of accuracy. The results show that the proposed methods are able to provide precise predictions. Moreover, the proposed methods are evaluated in terms of privacy and supplementary costs. It is also demonstrated that they are secure and online overhead costs due to privacy concerns are insignificant.

5.1. Introduction

According to the data utilization in online phase, CF algorithms can be categorized into two groups: memory- and model-based algorithms. Each group of algorithms has their own advantages and disadvantages (Su and Khoshgoftaar, 2009). Since memory-based algorithms operate on entire data online, for large data sets, online performance degrades and scalability problems occur. Accuracy and coverage might get worse when data are sparse because they depend on the existence of ratings for co-rated items. However, they are easily implemented and new user and/or item can be easily and incrementally added. Moreover, accuracy is better compared to model-based ones. On the other hand, model-based methods generate a model off-line. Thus, their online efficiency is better. Furthermore, methods in this type better address sparsity, scalability, and coverage problems. Some computations can be done off-line to improve online performance while predictions should be generated using memory-based methods. Hybrid schemes are proposed to gain the advantages of both memory- and model-based algorithms while decreasing the effects of disadvantages mentioned above (Su and Khoshgoftaar, 2009). Pennock et al. (2000) experimentally show that the hybrid scheme they proposed outperforms both types of schemes in accuracy while ensuring scalability. Their proposed method is in the hybrid manner, which

realizes some computations off-line to improve online performance while generates predictions using memory-based methods.

In this chapter, it is focused on how to produce high quality referrals on hybrid CF approaches efficiently from CPD while ensuring corporate privacy. The proposed scheme first estimates a model off-line and then predictions are provided based on the model using a memory-based approach. Since accuracy, privacy, and online efficiency are conflicting goals, the proposed methods should provide balance among them. Intuitively, the key requirement is to obtain preferable accuracy gain due to data integration despite accuracy losses due to privacy measures. Supplementary costs due to privacy concerns should be small and still make it possible to offer loads of referrals to many users efficiently. In other words, the proposed schemes must run and respond to each query in acceptable time with little online overhead costs.

Partitioned data-based CF with privacy is becoming popular with increasing popularity of e-commerce. Polat and Du (2008) show how to offer top-*N* recommendations based on HPD or VPD without deeply violating data owners' privacy. This study is different from their work. They consider HPD or VPD, while here CPD is considered. CF algorithm focused in this study executes on numeric ratings, while theirs operate on binary ratings. Moreover, they offer sorted list of referrals, while the proposed schemes produce predictions for single items. Polat and Du (2005b) discuss how to provide predictions for single items based on VPD between two parties while preserving their privacy. The authors consider VPD, while this study investigates how to offer predictions with privacy based on CPD. Furthermore, they consider all users in the database as neighbors and utilize entire users' data for prediction computations, while it is proposed privacy-preserving schemes to select *a*'s neighbors given a set of *n* users. Since they utilize all users' data, they are able to conduct some computations off-line. Although prediction computations are performed on selected users' data, the parties can still be able to conduct some computations like selecting the best neighbors off-line to improve online performance. Yakut and Polat (2010) investigate how to produce SVD-based private recommendations on HPD or VPD between two data owners. The authors utilize SVD-based CF scheme to produce

referrals based on partitioned data between two parties only while protecting their privacy. Unlike this study, they also consider VPD or HPD. Kaleli and Polat (2007a) investigate how to achieve NBC-based CF tasks on distributed data with privacy. The authors utilize binary ratings and NBC-based CF algorithm to generate referrals, where the scheme determines whether *a* will like *q* or not, while this scheme determines how much *a* will like or dislike *q*. Unlike their schemes, CPD is investigated in this proposal.

5.2. Hybrid Collaborative Filtering

CF systems first calculate the similarities between *a* and each user in *D* using a similarity measure. Users *a* and *u* can be thought as two vectors in an *m* dimensional item-space. The similarity between them (w_{au}) can be calculated by computing the cosine of the angle between these two vectors, as follows (Sarwar et al., 2001):

$$w_{au} = \cos(\vec{r_a}, \vec{r_u}) = \frac{\vec{r_a} \cdot \vec{r_u}}{\left\|\vec{r_a}\right\| \left\|\vec{r_u}\right\|}, \tag{5.1}$$

where the · represents the dot-product operation, $\left\|\vec{r_x}\right\|$ represents the Euclidean length of the vector $\vec{r_x}$, which is the square root of the dot product of the vector with itself, and $\vec{r_a}$ and $\vec{r_u}$ are users *a*'s and *u*'s ratings vectors, respectively. To select a subset of the users as neighbors of *a*, Herlocker et al. (1999) propose the following scheme, referred to as the best-*N*: select the best *N* correlates for a given *N* as neighbors. The prediction for *a* on item *q* (p_{aq}) can be computed, as follows (Herlocker et al., 1999), where $v_{norm_{uq}} = v_{uq} - \overline{v_u}$, n_a shows the number of *a*'s neighbors who rated *q*, $\overline{v_u}$ is the average of user *u*'s ratings, and w_{au} is the similarity weight between users *a* and u:

$$p_{aq} = \overline{v_a} + \frac{\sum_{u=1}^{n_a} v_{norm_{uq}} \times w_{au}}{\sum_{u=1}^{n_a} w_{au}}, \tag{5.2}$$

5.3. Hybrid Collaborative Filtering on Cross Partitioned Data with Privacy

Before giving the details of the proposed scheme, the problem that should be solved can be briefly defined, as follows: *When data collected for CF purposes are cross partitioned between two parties, how do such companies offer recommendations using hybrid CF algorithms based on CPD without invading their corporate privacy to each other?* The proposed solution consists of off-line model construction and online prediction estimation. Off-line computation functions include mean, deviation from mean normalization, and cosine similarity for determining similar items. Similarly, during online phase, cosine similarity is utilized for estimating user-user similarities. In addition, active users' neighbors are determined. Finally, using a prediction algorithm, a referral is computed. The proposed solutions are certainly expected not to conflict predefined principal and auxiliary privacy constraints. While searching for a solution to the above-mentioned problem, there are major challenges that should be considered. The first one is providing high quality predictions while preserving confidentiality. Second, the proposed scheme must ensure online efficiency. Finally, besides protecting individual votes, rated items should also be guarded.

5.3.1. Off-line Process

The first task in off-line phase is data normalization using deviation from mean method (Herlocker et al., 1999). To do that, the parties need the mean ($\overline{v_u}$) of each user u's ratings for all u = 1, 2,…, n. Recall that Eq. (2.5), the mean is an example of algebraic measures, which can be computed by applying an algebraic operation to two or more distributive measures. An important property of distributive measures like sum, count, and so on, is that they can be computed by partitioning the entire set into smaller subsets, computing the measure for each subset, and merging their results. Therefore, the parties are able to compute $\overline{v_u}$ for all u = 1, 2, …, n, as follows in a distributive manner as in Eq. (2.6).

The parties first compute the corresponding aggregate data (sum and count). They then exchange them and finally compute the final outcomes. However, they want to achieve such tasks without deeply jeopardizing their privacy. Note that since a can ask predictions from either party, each party should be able to return a

prediction to *a*, based on the distributed computation with the other party. Also note that both parties should exchange data during data normalization. Otherwise, they might not join the distributed CF process. Therefore, the following privacy-preserving scheme is proposed in which fake or default ratings are added to data holders' databases to calculate the $\overline{v_u}$ values in a secure manner:

1. Each party *j* selectively or uniformly randomly chooses a γ_j value over the range (1, 100].
2. Each party *j* then uniformly randomly generates a random value δ_j over the range (1, γ_j].
3. The companies selectively or uniformly randomly choose some of their unrated items' cells to fill with fake ratings (v_f) based on δ_j. Note that $C_g^f = C_{f-g}^f$, where C_g^f shows the number of ways picking *g* unordered outcomes out of *f* possibilities. Therefore, if $\delta_j < (100\text{-}2d_j)/2$, then the party sets δ_j at the chosen value; otherwise, it sets it at 100- d_j - δ_j, where d_j shows the density, as percent, of the data that company *j* holds. The parties then selectively or uniformly randomly choose δ_j percent of their unrated items' cells to fill with v_f.
4. Each party might decide to use one of the following methods to find v_f in each sub-part of *D*. Note that *D* might be considered to be split into four sub-parts, where each party holds two of them:
    a. Generate v_f using the distribution of the ratings residing in that sub-part. After determining the distribution of the ratings and their characteristics in each sub-part, the parties generate v_f using that distribution. Users' preferences about many products usually show Gaussian distribution with mean (μ) and standard deviation (σ). Therefore, the parties compute the mean and the standard deviation of the ratings in each sub-part; and generate v_f using the corresponding values for each sub-part. Note that number of v_f depends on how many cells to be filled, which are determined before.
    b. Rather than using a distribution to generate v_f, determine non-personalized or default ratings (v_d) by computing local averages in

each sub-part. The parties can utilize one of the following methods to find v_d in each sub-part:

i. *Local overall mean*: Calculate the overall mean of the ratings in that sub-part as a non-personalized rating.

ii. *User (row) mean*: Compute the mean ratings of each user using their ratings in that sub-part as non-personalized ratings.

iii. *Item (column) mean*: Compute the mean ratings of each item using their corresponding ratings in that sub-part as v_d.

5. After deciding which method to use for determining v_f, they generate or find them; and fill chosen unrated items' cells with corresponding v_f. Besides original databases, the parties end up with filled matrices.

6. Since *a* can ask prediction from *A* or *B*, both parties should normalize their data using deviation from mean method. Moreover, both of them should give aggregate results of its data to the other party; and receive results. Therefore, they follow the following steps to normalize their data, where for simplicity, we explain the procedure in terms of *A* only:

   a. *A* finds aggregate values ($\sum_{j=1}^{M_{uA}} v_{uj}$ and M_{uA}) on filled database for all $u = 1, 2, \ldots, n$; and sends them to *B*.

   b. After receiving such values, *B* computes $\sum_{j=1}^{M_{iB}} v_{uj}$ and determines M_{uB} values based on its original database.

   c. *B* now can estimate $\overline{v_u}$ values for all users using the received aggregate data on filled database from *A* and the aggregate data on its original data. It then sends $\overline{v_u}$ values to *A*.

   d. *A* and *B* compute $v_{norm_{uj}} = v_{uj} - \overline{v_u}$ values of filled and original databases, respectively. Therefore, the parties obtain normalized values based on deviation from mean method. They then save such normalized databases, as shown in Fig. 5.1a.

7. The parties simultaneously follow the same steps, where they switch their roles. Therefore, they end up with normalized databases, as shown in Fig. 5.1b.

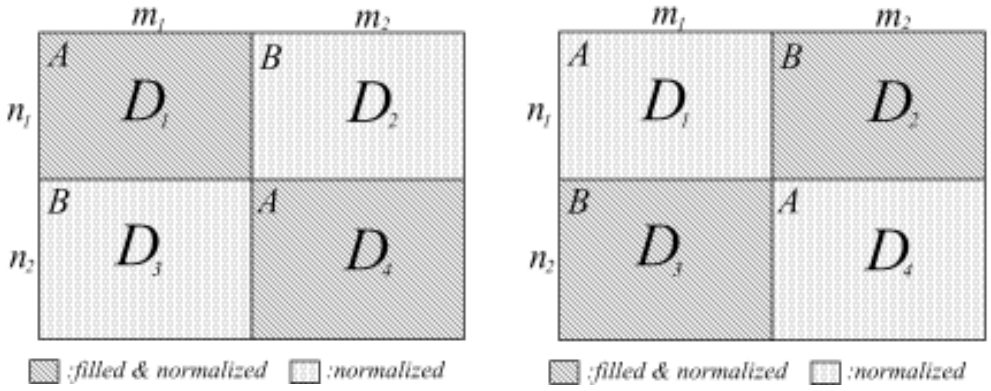

Figure 5.1. Masked Normalized Databases (a) $Model_1$ (b) $Model_2$

Data holders mask their data by filling some of the unrated item cells with non-personalized votes. It is possible to estimate sum values using some secure multi-party computation (SMC) protocols. However, it is hard to implement such cryptographic protocols compared to the proposed schemes. Such cryptographic solutions introduce additional computation costs so that efficiency becomes lower. Moreover, in addition to hiding true rating values, cooperating parties want to hide the rated and/or unrated items, as well. Filling some of the empty cells helps companies conceal their rated items. Finally, adding default ratings into user-item matrices increases amount of available ratings, which helps provide more dependable predictions and improve coverage.

After data normalization, each party ends up with two normalized databases; one is based on filled while one is based on original ratings. Since the parties collect ratings over time, they update their data; and off-line process is repeated per particular time. To prevent data deriving about each other data through such continual computations, data holders remove some of the old ratings and insert the newer ones. To improve the overall performance of the algorithm proposed by Herlocker et al. (1999), it is proposed to utilize hybrid approaches. For this purpose, after data normalization, the parties then construct a model off-line, where the best similar items to each item are found by using cosine similarity measure. When *a* asks predictions from *A*, the parties work on the model

generated from the database (MD_1) shown in Fig. 5.1a, which is called $Model_1$. If *a* asks predictions from *B*, the parties work on the model constructed from the database (MD_2) shown in Fig. 5.1b, which is called $Model_2$. Then, predictions are estimated based on such model using the proposed memory-based algorithm. Two items *i* and *j* can be thought of as two vectors in an *n* dimensional item-space (Sarwar et al., 2001). As in Eq. (5.1), the similarity between two items (w_{ij}), can be similarly calculated by computing the cosine of the angle between two vectors. The best similar items to each item are then determined based on such similarity measures. After choosing such similar items to each item off-line, the model is constructed. It is then utilized for prediction computations. Eq. (5.1) can be simplified, as follows:

$$w_{ij} = \cos(\vec{r_i}, \vec{r_j}) = \sum v'_{ik} \times v'_{jk}, \tag{5.3}$$

where $v'_{ik} = v_{ik} \Big/ \sqrt{\sum_{k=1}^{n} v_{ik}^2}$ and $v'_{jk} = v_{jk} \Big/ \sqrt{\sum_{k=1}^{n} v_{jk}^2}$.

As seen from Eq. (5.5), to select the best similar items, normalized ratings are needed. To determine v'_{ik} and v'_{jk} values, the parties need to compute $\sum_{k=1}^{n} v_{ik}^2$ and $\sum_{k=1}^{n} v_{jk}^2$ aggregates, where

$$\sum_{k=1}^{n} v_{ik}^2 = \sum_{k=1}^{n_1} v_{ik}^2 + \sum_{k=1}^{n_2} v_{ik}^2, \tag{5.4}$$

where n_1 and n_2 show the number of users held by *A* and *B*, respectively. Similar equation can be written for $\sum_{k=1}^{n} v_{jk}^2$. *A* and *B* need to compute sub-aggregates, $\sum_{k=1}^{n_1} v_{ik}^2$ and $\sum_{k=1}^{n_2} v_{ik}^2$, respectively; and exchange them. Since their databases are already masked or filled during data normalization, they compute such aggregates for all items and exchange them. They then determine v'_{ik} and v'_{jk} values. Due to filled databases, they cannot derive information about each other's ratings. Now, they can estimate w_{ij} values, as follows:

I. For similarities between those items in the left or the right halves, the parties can compute w_{ij} values, as follows:

a. *A* finds $\xi_{K_{A_1}}(\sum_{k=1}^{n_1} v'_{ik} v'_{jk}) = \xi_{K_{A_1}}(A_i)$ values for the first item and sends them to *B*, where ξ shows HE function while K_{A_1} is *A*'s public key.

b. *B* similarly finds $\xi_{K_{A_1}}(\sum_{k=1}^{n_2} v'_{ik} v'_{jk}) = \xi_{K_{A_1}}(B_i)$ values for the first item.

c. *B* uses HE and gets $\xi_{K_{A_1}}(A_i + B_i)$ values. Note that there are m_1 - 1 such values.

d. *B* permutes them using a permutation function f_B and sends them back to *A*. After receiving them, *A* decrypts them and determines those satisfying τ, where τ is a threshold value.

e. It then sends those values satisfying τ to *B*. Since *B* knows the f_B, it determines which items whose similarities with the first item satisfy τ.

f. *B* informs *A*. Now, *A* and *B* know the items that are among the best similar items to the first one.

g. For the other items, the parties follow the similar steps. However, since some of the similarities are already computed, *A* sends bogus data for known similarities and uses different keys to encrypt the sub-aggregates. *B* similarly adds bogus data for already computed similarities.

h. For the items in the right half, the parties follow the same steps; however, they switch their roles.

II. For similarities between those items in the different halves, the parties can compute w_{ij} values, as follows:

a. For the upper half, the parties perform the followings:

i. *A* permutes m_1 items and hides each item column vector into X - 1 random vectors. In other words, it creates X - 1 random vectors for each item vector and hides each true vector into them. It then sends the first group of items vectors to *B*.

ii. *B* computes the scalar product between each received vector and each column vector in its database. It then finds

$\xi_{KB}(\sum_X)$ values for all *X*, where $\sum$ shows the scalar product of two vectors, K_B represents *B*'s public key.

iii. *A* uses OT to get the required results or sub-aggregates for that item. *A* follows the same steps for all items.

iv. *A* now has $\xi_{KB}(w_{1m_1+1}), \xi_{KB}(w_{1m_1+2}), \ldots, \xi_{KB}(w_{1m})$ and $\xi_{KB}(w_{2m_1+1}), \xi_{KB}(w_{2m_1+2}), \ldots, \xi_{KB}(w_{2m}), \ldots, \xi_{KB}(w_{m_1m_1+1}), \xi_{KB}(w_{m_1m_1+2}), \ldots, \xi_{KB}(w_{m_1m})$.

b. For the lower half, the parties perform the followings:

i. *A* permutes m_2 items. It hides each item column vector into *Y* - 1 random vectors. In other words, it creates *Y* - 1 random vectors for each item vector and hides each true vector into them. It then sends the first group of items vectors to *B*.

ii. *B* computes the scalar product between each received vector and each column vector in its database. It then finds $\xi_{KB}(\sum_Y)$ values using HE.

iii. *A* uses OT to get the required results or sub-aggregates for that item. *A* follows the same steps for all items.

iv. *A* now has the similar values as in the upper half. *A* now uses HE to get the similarity values.

v. It then creates random numbers $r_{A_{ij}}$ and finds $\xi_B(r_{A_{ij}})$. It uses HE to get $\xi_B(w_{ij} + r_{A_{ij}})$ values; and sends them to *B*.

vi. *B* decrypts them and sends them back to *A*. It now can find w_{ij} values by getting rid of random numbers. It then determines those satisfying τ and informs *B*.

Note that *A* cannot derive data if $n > m$, because there are *n* unknowns but *m* equations. It cannot find a unique solution from given equations. If $n < m$, then the parties should follow the following solution: *B* adds random numbers to disguised similarity values by $A(w_{ij} + r_{A_{ij}} + r_{B_{ij}})$ generated from the range [$-\tau \times c\%, \tau \times c\%$], where *c* is a security parameter. *A* then correspondingly adjusts τ.

5.3.2. Online Process

There are three major steps in online phase: Calculating w_{au} values between a and each user u for all u = 1, 2, …, n, determining a's neighbors, and computing p_{aq} using a's neighbors' ratings for q and the corresponding w_{au} values. It determines a's ratings mean vote ($\overline{v_a}$). It then normalizes her ratings using deviation from mean approach. Finally, it sends her normalized ratings and q to the other party. For simplicity, it is assumed that A acts as the MP. In other words, the parties work on $Model_1$. When a asks prediction from B, the parties follow the same steps by switching their roles, where they work on $Model_2$. The parties answer the query securely, as follows:

Computing w_{au} values: Note that w_{au} values can be computed using the cosine measure in a secure manner. Eq. (5.1) can be simplified, as follows:

$$w_{au} = \cos(\vec{r_a}, \vec{r_u}) = \sum_k v'_{ak} \times v'_{uk} \ , \tag{5.5}$$

where k shows the commonly rated items by both a and u; and $v'_{uk} = v_{uk} \Big/ \sqrt{\sum_{k=1}^{M_u} v_{uk}^2}$ and $v'_{ak} = v_{ak} \Big/ \sqrt{\sum_{k=1}^{M_a} v_{ak}^2}$. Note that M_a and M_u show the number of rated items by a and user u, respectively. Suppose that a asks prediction from A. Therefore, it acts as the MP. To find w_{au} values, the parties need to exchange aggregate data. Due to hybrid distribution, ratings of q are held by both parties. Therefore, each party j computes $\sum_{k \in k_j} v'_{ak} \times v'_{uk}$ values for all users u = 1, 2, …, n; sends the aggregate data, of those users whose ratings for q are held by the other party, to the other party. For example, suppose that the first n_1 users' ratings of q are held by B and the last n_2 users' ratings of q are held by A. In other words, q is one of the last m_2 items in Fig. 1.1. Then, A sends the corresponding similarity weight values or aggregate data for the first n_1 users to B, while B sends the corresponding similarity weight values for the last n_2 users to A. They find similarities using cosine measure in a secure manner, as follows:

a. A finds v'_{ak} values because it knows a's ratings, and sends them to B.

b. To determine v'_{uk} values, on the other hand, they need to compute $\sum_{k=1}^{M_u} v_{uk}^2$ aggregates, where

$$\sum_{k=1}^{M_u} v_{uk}^2 = \sum_{k=1}^{M_A} v_{uk}^2 + \sum_{k=1}^{M_B} v_{uk}^2 \,, \tag{5.6}$$

where M_A and M_B show user u's rated items held by A and B, respectively. The parties need to exchange such sub-aggregates to determine v'_{uk} values. They compute such values based on the filled matrices. Note that they masked their original databases using fake ratings during data normalization off-line. Therefore, they cannot derive information about each other's data while exchanging such sub-aggregates. Each party j then computes $\sum_{k=1}^{M_j} v_{uk}^2$ values for all u = 1, 2, …, n. They then exchange them and find v'_{uk} values.

c. As explained before, since scalar dot product is a distributive measure, the parties can compute w_{au} values using cosine measure, as follows:

$$w_{au} = \sum_{k \in k_A} v'_{ak} \times v'_{uk} + \sum_{k \in k_B} v'_{ak} \times v'_{uk} \,. \tag{5.7}$$

Each party can act as a in multiple scenarios to derive information about the other party's ratings. Therefore, we propose the following privacy-preserving scheme to securely calculate w_{au} values. The parties basically insert some fake normalized ratings into a's normalized ratings vector corresponding part like they do when normalizing their ratings. Considering the same example mentioned above, A fills some of a's unrated items' cells among the first m_1 items, while B fills some of a's unrated items' cells among the last m_2 items, as follows:

a. Each party j selectively or uniformly randomly chooses α_j over the range (1, 100].

b. Each party j uniformly randomly generates a random value β_j over the range (1, α_j].

c. They selectively or uniformly randomly choose β_j percent of a's unrated items' cells from the corresponding part to fill with fake normalized ratings (v_{fn}).

d. Each party might decide to use one of the following methods to determine v_{fn}:

i. Generate v_{fn} values using the distribution of a's normalized ratings.

ii. The parties can use default normalized ratings (v_{dn}), which are based on non-personalized ratings as v_{dn}: Use item (column), user (row), or local overall mean approach to determine them. They compute the mean normalized ratings of each item using their corresponding values held by that party.

e. Each party finally fills selectively or uniformly randomly chosen *a*'s corresponding unrated items' cells with v_{fn}.

Note that the parties perform those steps whenever *a* asks a prediction because each party can act as *a* to derive data. However, the parties can compute the item mean values off-line because they already have the required data to find them. That improves online performance. After adding v_{fn} into *a*'s vector, the parties can securely compute w_{au} values in a distributed manner. Each party sends the required aggregate data to the other party, which holds the ratings of *q*. The party then computes the final w_{au} values by adding aggregate data values.

Determining *a*'s neighbors: In order to determine *a*'s neighbors using the best-*N* approach, decreasingly sorted order of the w_{au} values are needed. Therefore, the parties follow the following steps:

a. It is assumed that *A* acts as the MP. The parties exchange the sub-aggregates needed to find w_{au} values. *A* and *B* then compute similarity weights between *a* and the n_1 and n_2 users, respectively.

b. B permutes n_2 similarity weights using a permutation function (f_B) and sends them to *A*.

c. A does not know the order of similarity weights due to permutation. However, for *A*, the probability of guessing the correct order of such weights is 1 out of $n_2!$. *A* then determines the best-*N* users as neighbors, where N_1 and N_2 users are selected from the users held by *A* and *B*, respectively, where $N = N_1 + N_2$. *A* then sends the chosen N_2 weights back to *B* who can determine which users are selected as neighbors because it knows f_B.

d. The parties now have *a*'s neighbors; however, they do not know which users are selected neighbors by the other party and which neighbors

already rated q. Those neighbors' ratings who rated q are used in prediction computation.

To further improve privacy or make it more difficult for the parties to learn the selected neighbors, the parties can perform the following: After determining the best N_1 and N_2 users among the users they hold, respectively, A and B uniformly randomly choose ψ_A and ψ_B percents of those users, who are member of the best-N users but not rated q, as neighbors, where ψ_j is an integer between 0 and d_j. Since such users do not rate q, the parties insert default values for q into such cells and use their data for prediction computation, as well.

Computing p_{aq}: p_{aq} can be estimated by using the algorithm proposed by Herlocker et al. (1999), as follows: Since A knows $\overline{v_a}$, it only needs the following value to compute p_{aq}:

$$P_{aq} = \frac{\sum_{u=1}^{n_a} v_{norm_{uq}} \times w_{au}}{\sum_{u=1}^{n_a} w_{au}}, \tag{5.8}$$

where $p_{aq} = \overline{v_a} + P_{aq}$ and $v_{norm_{uq}} = v_{uq} + \overline{v_u}$. Similarly, since P_{aq} can be computed in a distributive manner, it can be written, as follows:

$$P_{aq} = \frac{\sum_{u=1}^{s_A} v_{norm_{uq}} \times w_{au} + \sum_{u=1}^{s_B} v_{norm_{uq}} \times w_{au}}{\sum_{u=1}^{s_A} w_{au} + \sum_{u=1}^{s_B} w_{au}}, \tag{5.9}$$

where s_A and s_B shows the number of a's neighbors who rated q held by A and B, respectively; and $n_a = s_A + s_B$. It can be simply written that $P_{aq} = X_A + X_B / Y_A + Y_B$. Note that A does not know $v_{norm_{iq}}$, w_{au}, s_B values owned by B, and which users selected as a's neighbors held by B. Similarly, B does not know $v_{norm_{iq}}$, w_{au}, s_A values that A holds, and which users selected as a's neighbors held by A. After computing X_B and Y_B values, B sends them to A, which calculates P_{aq}. It then can compute p_{aq} by de-normalizing P_{aq}; and sends it to a.

5.4. Privacy Analysis

The principal privacy constraint states that the proposed schemes should hide actual ratings and rated items residing in data owners' databases against each other. Likewise, auxiliary privacy constraint affirms that no exchanged intermediate computation value allows parties inferring information conflicts the principal privacy constraint. Collaborating vendors are semi-trusted ones. Furthermore, they can also act as a in multiple scenarios to learn the other party's private data. In the proposed scheme, publicly known values are user and item IDs, each party's public key, partial aggregate results like sum and count, and the final prediction because each party can act as a. Similarly, confidential data include both true rating values and rated items.

The proposed solution consists of two major functions, off-line model generation and online prediction estimation. The output of off-line phase is a model. The parties share the neighbors of each item without exchanging actual weights. On the other hand, the output of online phase, called p_{aq}, might be known by both parties because any party can act as an active user. Off-line phase includes protocols like mean, similarity, and neighborhood formation. The output of mean protocol is user mean ratings. Similarity weights between any two pairs of items are obtained after similarity protocol. Finally, the best items are determined for each item during neighborhood formation. Online phase contains three protocols: estimating user-user similarities whose output is similarity weights between a and each user in D, forming a's neighborhood whose output is the best N similar users to a, and finally estimating p_{aq} whose output is the prediction returned to a. In the following, we demonstrate that the parties are not able to derive any data during both off-line and online phases. More specifically, it is shown that the proposed schemes preserve data owners' privacy during normalization and model construction conducted off-line; and computing w_{au} values, determining a's neighbors, and calculating p_{aq}, which are performed online.

During normalization, the parties estimate user mean values by exchanging partial aggregate results computed on filled databases. Thus, they are not able to derive information about each other's private data due to default ratings-based randomness and exchanged aggregate data. Even if they know the number of

ratings during computing $\overline{v_u}$ values, they cannot learn true rating values. Moreover, they do not know the v_d values, because such ratings are estimated from data residing in each party's database. However, the parties can guess the rated or unrated items in each other's databases because they exchange the number of ratings involved in mean or normalization protocol. For example, for the party A, guessing the correct γ_i is 1 out of 100. After guessing it, the probability of guessing the correct δ_B values is 1 out of γ_B. Since A knows n_B, it can guess the number of filled unrated items' cells with probability 1 out of (100 $\times$ γ_B). Then, it can guess the rated items by a single user with the probability of 1 out of (100 $\times$ $\gamma_B \times C_{B_r}^{m_B}$, where B_r represents the number of rated items among the items B owns by a user and m_B shows the number of items held by B. The probability for B can be determined similarly. During similarity computations and neighborhood formations, privacy is achieved via permutation, randomization, HE, and OT protocol. The probability of guessing the correct value from perturbed ones is 1 out of (m - 1)!. Similarly, after permutation, column vectors are hidden into random vectors. The probability of guessing the true column vector from disguised ones is 1 out of X (or Y). The value of X or Y can be determined based on how much privacy and off-line performance the parties want. Due to HE and OT protocol, which happen to be secure (Even et al., 1985; Naor and Pinkas, 1999), the parties cannot derive information about each other data. At the end, the parties share the best similar items to each item without revealing similarity weights.

In online computations, the parties can act as a in multiple scenarios to learn data residing in the other party's database. In such attacks, the party acting as an a changes only one rating in its ratings vector to learn the true votes and rated and/or unrated items in the other party's database during w_{au} computations. However, to defend themselves against such types of attacks, the parties perturb a's ratings vector, as explained previously. Since a's vector is masked like the parties disguise their databases using default ratings, privacy analysis can be similarly done. During similarity estimation, the parties exchange partial aggregates computed based on filled data. Without knowing the number of ratings involved in such calculations, it is not feasible to determine true ratings from such

aggregates. Proposed neighborhood formation protocol for a is also secure. The parties are not able to derive data while determining a's neighbors, because they do not exchange anything. First of all, they do not know how many and which users are selected by the other party as a's neighbors. Second, they do not know which neighbors have already rated q, because only those neighbors' data who already rated q are used in prediction estimation. Due to the similar reasons, proposed p_{aq} estimation protocol also does not violate main confidentiality constraint. Since one of the parties sends aggregate values to the MP during P_{aq} computation; and the MP does not know the values involved in P_{aq} computation and which users selected as neighbors held by the other party, it cannot derive information about the other party's data while computing P_{aq}.

5.5. Supplementary Costs Analysis

The proposed schemes consist of both off-line and online phases. Unlike online costs, off-line costs are not critical for overall performance. Due to privacy measures, additional costs are inevitable. However, such extra online costs should be small enough to provide predictions efficiently. In the following, the proposed schemes are analyzed in terms of supplementary storage, computation, and communication (number of communications and amount of transmitted data) costs.

Off-line phase consists of two major tasks: data normalization and model construction. In data normalization, the parties fill their sets with fake ratings and estimates user mean votes. Since fake ratings are non-personalized ratings, it takes constant time to determine them for each item or user. Thus, due to determining fake ratings, computation costs are in the order of $O(m)$ or $O(n)$ for item or user default ratings, respectively. Other computations like randomly selecting some parameters, empty cells, calculating sum and count values, and filling empty cells for normalization are negligible. In off-line process, the parties end up with two matrices. Due to them, supplementary storage costs are in the order of $O(nm)$. Due to fake or default ratings, there is no additional storage cost, because they can be derived. However, each party can determine v_f values off-line and saves them to improve online performance. In this case, due to v_f values, additional storage costs are in the order of $O(m)$ or $O(n)$ for item or user default ratings, respectively. To

normalize data, the parties communicate with each other. For each matrix, number of communications is two only. Thus, total number of communications is four only for normalization.

Model construction includes estimating similarities between items and determining those items satisfying a pre-determined threshold. Given an $n \times m$ matrix, without privacy concerns, computation costs for calculating similarities between m items are in the order of $O(m^2n)$. Those items satisfying the threshold are selected as the best similar items. With privacy concerns, the proposed model construction scheme includes HE and OT protocol. Other than encryption, decryption, HE, and OT, computations like permutation, creating random vectors, addition, and so on are negligible. Additional off-line computation costs during model creation due to privacy concerns are, as follows: The amounts of encryptions and decryptions are in the order of $O(m^2)$. Similarly, number of HEs is in the order of $O(m^2)$. To determine the running times of cryptographic algorithms, benchmarks for the CRYPTO++ toolkit from http://www.cryptopp.com/ can be used (CRYPTO++, 2009). An experiment is performed for testing the computational time spent on estimating the scalar product of two vectors using HE. Two randomly selected vectors with varying length from MovieLens Million (MLM) data set are used, which is described in the following. The vector length is varied from 1 to 1,000 and the corresponding computation time for each length value is displayed in Fig. 5.2. As expected, with increasing vector length, computation time augments. The computation times are not shown when there is no encryption, because even if the vector lengths are 1,000, the time to find their scalar product is less than 1 microsecond. As seen from the figure, encryption significantly affects the computation times needed to estimate scalar product of two vectors. Since HE is utilized off-line, such costs are not that critical for the overall performance.

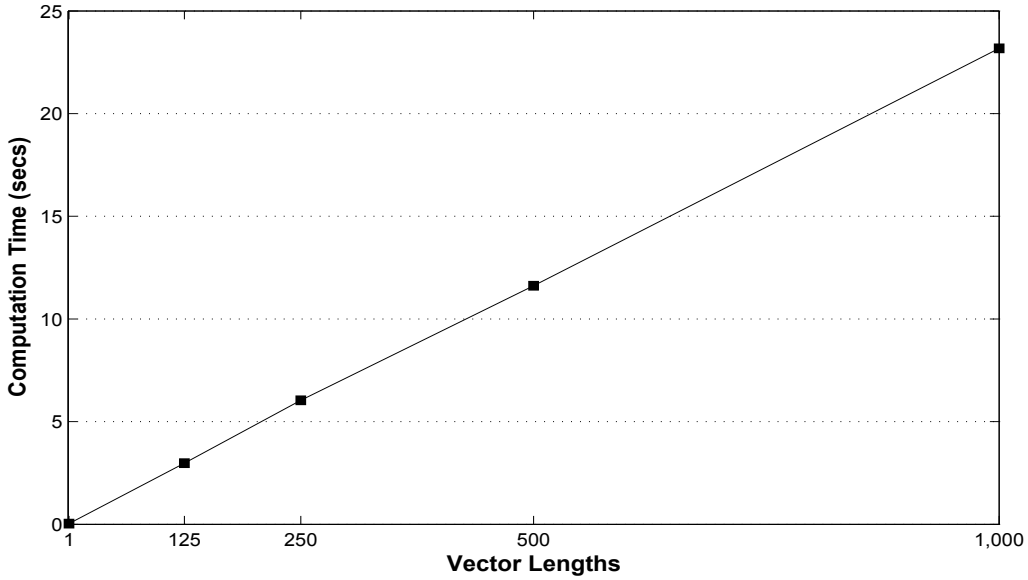

Figure 5.2. Effects of Encryption on Computation Time Spent on Scalar Product

The parties utilize $m_1 + m_2$ number of OT. Thus, number of OT employed is in the order of $O(m)$. For each OT, one of the parties (the sender) performs $2(X + Y)$ double exponentiations, while the other party (the receiver) conducts six exponentiations (Naor and Pinkas, 1999). An efficient OT protocol proposed by Naor and Pinkas (1999) could be achieved with polylogarithmic (in n) communication complexity. During model construction, number of communications between parties is $5(m+1)$ except communications due to OT.

In a prediction process, a sends a message containing her ratings and a query to the CF system, which returns a result. Thus, number of communications is two only. In the proposed scheme, since the parties communicate with each other online, the number of communications is six in total. Due to proposed schemes, additional amounts of data to be transferred are, as follows: First, a's ratings are sent to both parties rather than one of them. Second, on average, A and B exchange $n/2$ aggregate values. Finally, one of the parties sends two aggregate values to the MP.

It is expected that privacy concerns cause extra computation costs. Additional costs due to selection of α_j and β_j, determining a's unrated items, and filling her vector with v_{fn} values online can be considered negligible. However, when computing w_{au} values, number of multiplications increases due to inserted v_f values in each company's database and v_{fn} values into a's vector. Due to inserted

v_{fn} values, on average, total number of additional multiplications is (1/2) × α_j/2× (1/100) × m'_a /2 × d_j, where d_j shows density rate of the database held by company j and m'_a represents number of a's unrated items. Similarly, due to inserted v_f values, on average, total number of additional multiplications is (γ_i/2) × (1/100) × m'_u × d_j, where m'_u represents number of unrated items in D. Note that d_j is very low; and α_j and γ_i values are constants less than 100. Note also that proposed schemes do not cause any auxiliary computation costs while determining a's neighbors and p_{aq} computation. Thus, extra online computation costs due to privacy concerns are acceptable; and still make it possible to generate predictions efficiently.

5.6. Prediction Quality Analysis: Experiments

After examining the proposed schemes in terms of privacy and supplementary costs, now the proposed schemes are going to be evaluated in terms of accuracy and online performance, as well. To achieve such goal, a variety of experiments are performed using two well-known real data sets. It is shown that the proposed schemes are secure and theoretically does not introduce significant additional costs. There is also need for demonstrating that the proposed schemes are able to offer accurate predictions efficiently while privacy measures are in place.

When data holders own inadequate data, they are limited to generate recommendations for some items. If they agree to collaborate to offer predictions on their distributed data when privacy-preserving measures are introduced, they are more likely to provide referrals to more items. Therefore, experiments are performed to demonstrate how coverage changes through collaboration with varying n values. It is also hypothesized that the parties are able to generate more truthful recommendations if they decide to collaborate. To verify this, experiments are conducted based on split data only and integrated data. It is also planned to show how collaboration between two e-commerce sites affects the quality of the CF services. Hybrid approaches enhance online performance because some works are done off-line. A hybrid approach is utilized, where the best similar items to each item are selected. Since a smaller number of items are involved in prediction computations, online performance is expected to improve.

Moreover, selecting some of the items for generating predictions affects accuracy, as well. Trials are conducted to demonstrate the effects of hybrid approach on online performance and accuracy.

In order to add randomness into data held by each party and active users' data, the parties might use different non-personalized values. In one hand, density of the data increases due to filled default values; that might make accuracy better. On the other hand, such values might not represent users' true preferences; that may make accuracy worse. To understand the effects of various default values and determine the best choice, which gives the most accurate results, a variety of trials are run. As explained previously, privacy and accuracy are two clashing goals. Due to privacy protection measures applied by the vendors, accuracy is expected to be worse. Along the proposal, some privacy parameters like γ_j and α_j are used. They are among the factors that might have an effect on accuracy. To show how the quality of the predictions changes with varying γ_j and α_j values, different experiments are performed. Finally, after determining the optimum values of each parameter like N, γ_j, and α_j, a final set of experiments are performed to evaluate the overall performance of the proposed schemes.

Various experiments are performed using well-known real data sets Jester and MLM. The results on these data sets can be generalized. Jester (Gupta et al., 1999) is a web-based joke recommendation system. MLM was collected by GroupLens at the University of Minnesota (GroupLens). MLM contains discrete votes while Jester has continuous ratings. The ratings range from -10 to 10 and 1 to 5 in Jester and MLM, respectively. Although Jester has 100 jokes, MLM has 3,592 movies. On the other hand, MLM and Jester have 7,463 and 73,421 users, respectively. In Jester, almost 44% of the ratings are available. Each user in MLM, on the other hand, has rated at least 20 movies. Jester is much denser than MLM. Two well-known accuracy metrics, MAE and NMAE, are utilized as in Chapter 2 while coverage is also determined using Eq. (2.13). And finally, in order to show how online performance varies with hybrid approach, T is defined, in seconds, as online time required for generating predictions.

Given the whole data sets, it is first determined those users who have rated more than 100 and 60 items from MLM and Jester, respectively. Such selected

users are then randomly divided into training and test sets. Training and test users are randomly chosen from training and test sets, respectively. Number of users and/or items selected for training and testing for each set of experiments might be diverse because different experiments might need different requirements. After selecting test users, five rated items are randomly selected as test items from each test user ratings vector. Such rated items' ratings are withheld, replaced their entries with null; and predicted their ratings using the proposed privacy-preserving scheme. Due to randomness, each experiment is run for 100 times to reliably catch the effects of uncertainty. The recommendations are compared provided with privacy concerns with the true user-specified ratings. After calculating MAE, NMAE, and T values, overall averages are displayed.

The details and the outcomes of the experiments performed based on two real data sets are explained in the following.

Experiment 1-Effects of Collaboration on Coverage and Accuracy: Collaboration between two e-commerce sites having limited amount of data definitely affects coverage. Insufficient data might result very low coverage. With increasing quantity of data, coverage is expected to improve. Therefore, at first, experiments are conducted to confirm the effects of partnership on coverage and to demonstrate how coverage changes with varying amount of data. For these experiments, users are uniformly randomly selected from given data sets. Both data sets are utilized while varying n from 100 to 2,000, where such users are randomly chosen from given data sets. It is assumed that predictions can be generated if q_t and r_c are at least one and two, respectively, where q_t shows the number of users who have already rated q; and r_c represents the number of commonly rated items between a and the user who has rated q. For MLM, the outcomes are displayed in Fig. 5.3. Since Jester is denser than MLM, when n is 10 or bigger, coverage for integrated data is always 100%. However, when n is 10 and 20, coverage for split data are 89% and 95%, respectively. Although Jester is dense, when n is small (less than or equal to 20), the parties are not able to offer recommendations for all items using their split data only. Through collaboration, however, they can generate predictions for all items.

The experiment results confirm premise that coverage improves when vendors offer referrals on CPD. Since MLM is sparse data set, as expected, coverage significantly recovers through collaboration. Also note that it develops with increasing *n* values. Due to collaboration and increasing *n* values, amount of ratings available for CF increases; that makes coverage better. Therefore, CF on CPD helps online vendors provide recommendations for more items, even if they have dense data sets.

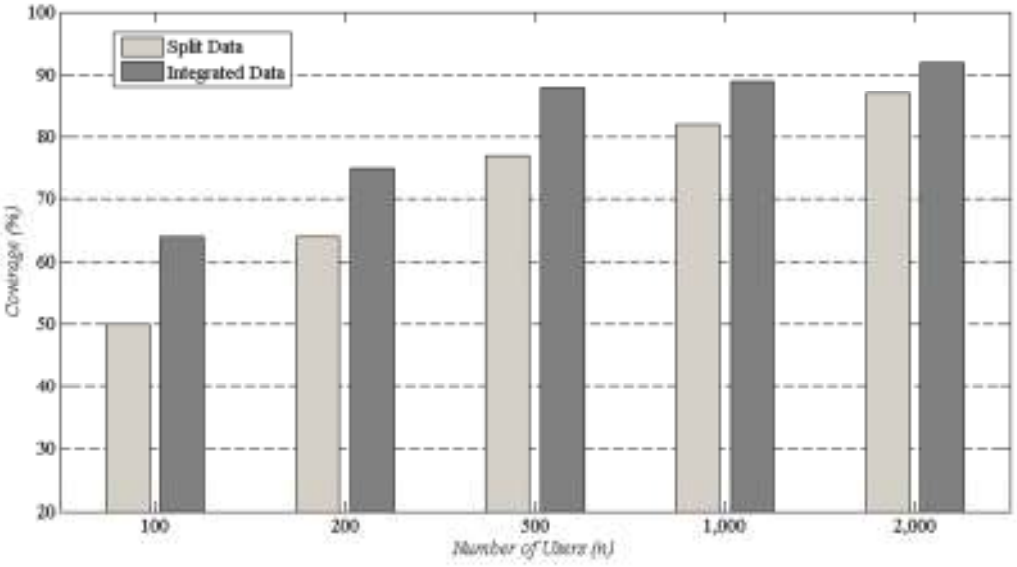

Figure 5.3. Coverage with Varying *n* Values (MLM)

To give an idea about how integrating split data affects the quality of the referrals, experiments are performed while varying amount of data that each party holds. Such experiments are set up using both data sets, where *n* varying from 100 to 1,000. Note that data are split between two parties, as shown in Fig. 1.1. 500 users are used for testing while computing predictions for five rated items for each test user. Predictions are first found using the data held by each party only. After calculating MAE and NMAE values for each party, then such values are averaged. Split data is finally integrated and predictions are computed for the same test set. After computing overall outcomes, the results are computed in order to show how collaboration between vendors affects accuracy. Since MAE and NMAE values show similar trends, MAEs are shown in Fig. 5.4 and 5.5 for both data sets.

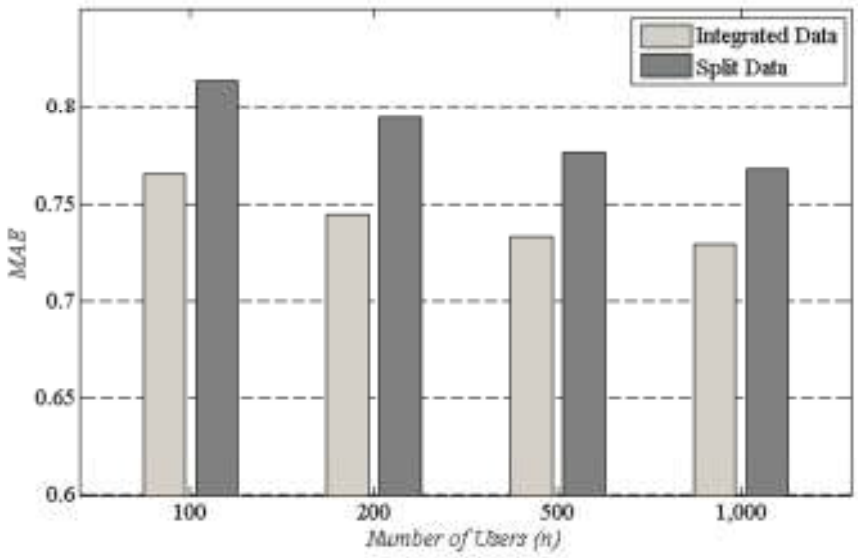

Figure 5.4. MAEs with Varying *n* Values (MLM)

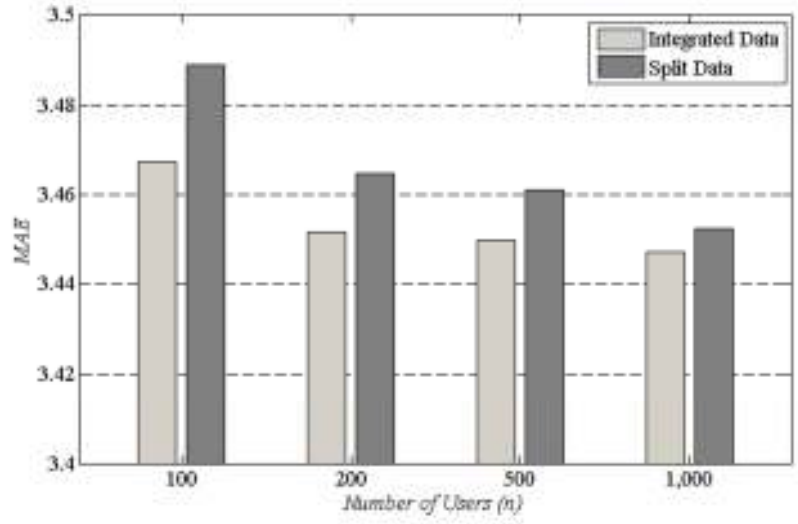

Figure 5.5. MAEs with Varying *n* Values (Jester)

As seen from Fig. 5.4 and 5.5, the quality of the predictions improves with collaboration between parties for both data sets. For MLM data set, which is sparse, MAE values significantly enhances when data owners integrate their split data. However, for Jester, such improvements are very small. This phenomenon can be explained the density of Jester. Since it is very dense compared to MLM, each party is able to offer accurate and dependable recommendations using their own data. As expected, with increasing *n* values, accuracy improves, as well. When there are 100 users, MAE values are 0.8139 and 0.7659 for split and integrated data, respectively for MLM. Thus, accuracy improves by 4.80%. For the same cases, NMAE values are 0.2034 and 0.1914, respectively. Similarly,

MAE enhances from 0.7684 to 0.7290 through collaboration for MLM when *n* is 1,000. For Jester, on the other hand, improvements are less than 1% due to data integration. When data collected for CF purposes are sparse, combining CPD between two parties definitely makes accuracy better. *t*-test analysis is performed with significance level being 0.05 to determine whether the improvements due to collaboration are statistically significant or not. For MLM with *n* being 200, the value of *t* is 27.06, which is greater than the value of *t* for 0.05 in the *t*-table. Although the improvements are small for Jester compared to MLM, the enhancements are still statistically significant. Similarly, for Jester with *n* being 100, the value of *t* is 2.198, which is still greater than the value of *t* for 0.05 in the *t*-table.

Experiment 2-Effects of Hybrid Approach: Trials are performed to demonstrate the effects of hybrid method on both performance and accuracy. For these experiments, *n* is fixed and set it at 500 while varying *N* from 3,592 to 125 and from 100 to 25 for MLM and Jester, respectively. Again, there are 500 test users and five rated items for each active user as test items. In other words, 2,500 recommendations are generated. First of all, it is demonstrated that how the quality of the recommendations and online performance change by applying the hybrid approach for Jester data set. Although MAE and NMAE values are computed, MAE and *T* values with varying *N* values for Jester data set demonstrated in Fig. 5.6 and 5.7, respectively. Similarly, trials are conducted to show how accuracy and performance vary with different *N* values for MLM. However, since MAE and NMAE show similar tendencies, NMAE are only displayed together with *T* values for MLM in Fig. 5.8 and 5.9.

MAE values slightly become worse with decreasing *N* values for Jester, as seen from Fig. 5.6. However, such accuracy losses are insignificant because when *N* values are 100 and 25, corresponding MAE values are 3.4501 and 3.4525, respectively. The loss in MAE values due to varying *N* from 100 to 25 is 0.0024 only. Unlike MAE, *T* values improve with decreasing *N* values according to Fig. 5.7, as expected. In order to generate 2,500 predictions using Jester, 4.42 seconds are spent online when *N* is 100. On the other hand, when *N* is 25, 3.85 seconds are needed to offer the same number of recommendations. If *N* is varied from 100 to

50, performance gain is 0.33 seconds while accuracy loss is 0.0019 only. For Jester, 50 can be determined as the optimum value of *N* based on accuracy and online performance. Without sacrificing on accuracy, online performance improves by using the best *N* items for prediction generation. Since there are 100 jokes only in Jester, improvements due to selecting the best items are limited.

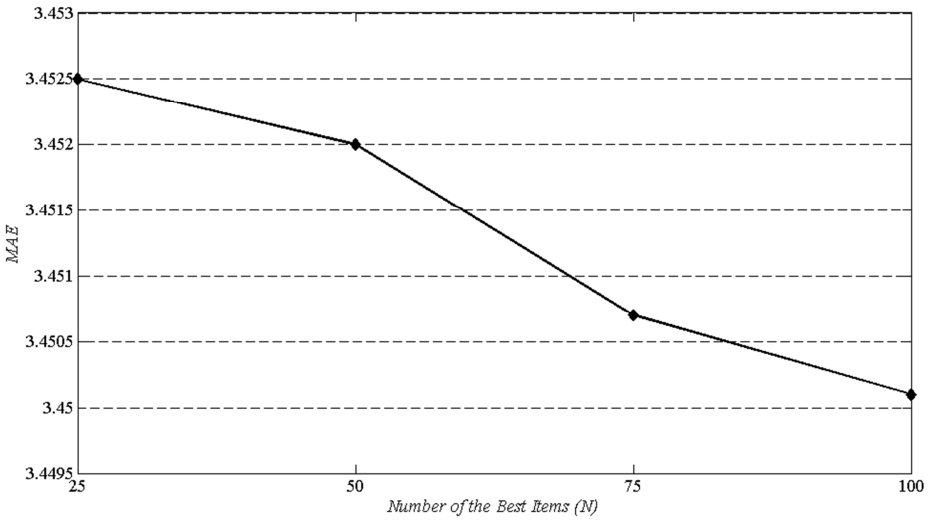

Figure 5.6. Accuracy with Varying *N* Values (Jester)

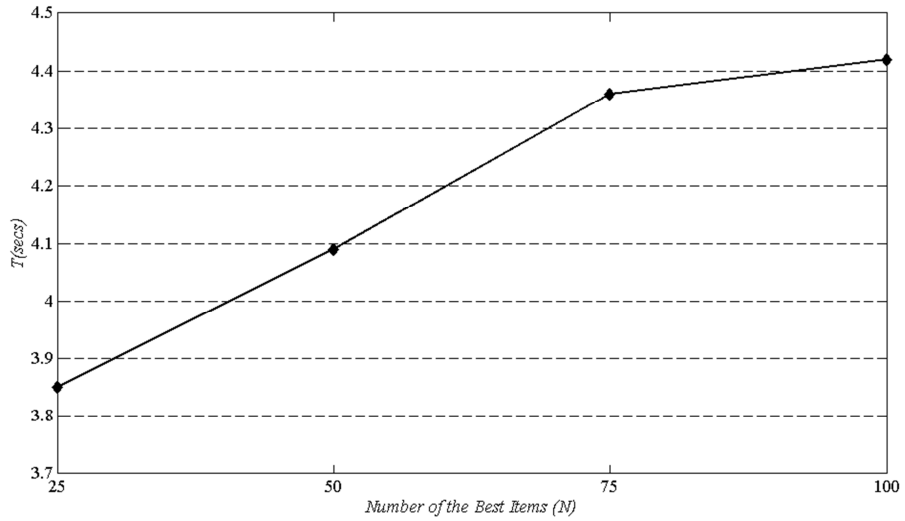

Figure 5.7. Performance with Varying *N* Values (Jester)

As seen from Fig. 5.8, NMAE values become worse with decreasing *N* values. Although NMAE values significantly degrade while varying *N* from 1,000 to 500 and so on, accuracy almost becomes stable while varying *N* from 3,592 to 1,000. The same quality can be achieved using the best 1,000 items instead of using the entire items' ratings.

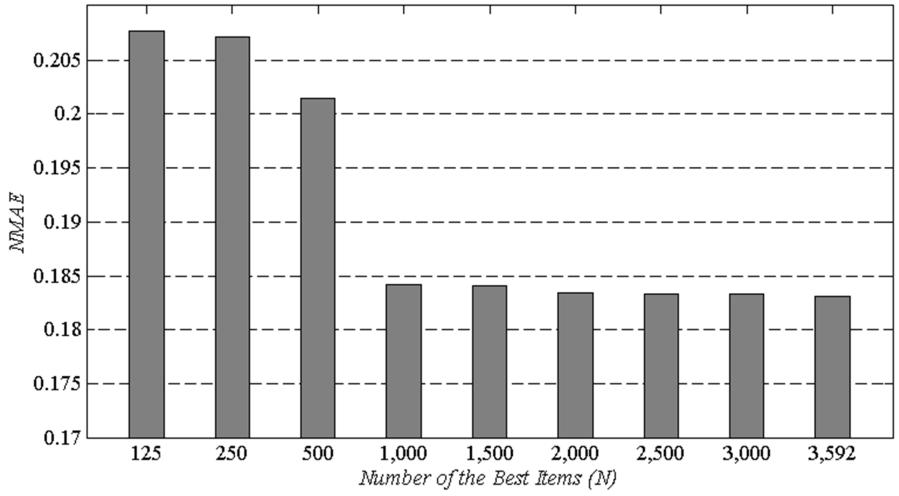

Figure 5.8. Accuracy with Varying *N* Values (MLM)

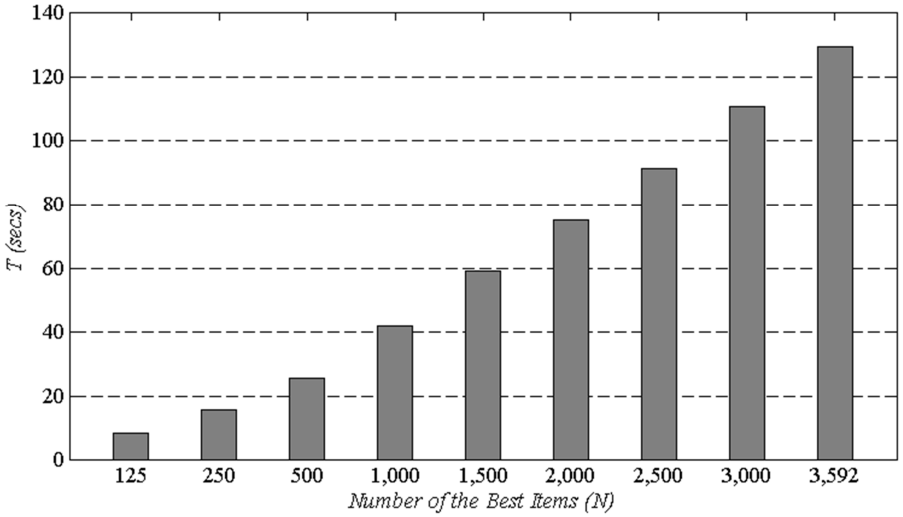

Figure 5.9. Performance with Varying *N* Values (MLM)

As expected, online performance improves with decreasing *N* values because amount of data involved in recommendation computations decreases according to Fig. 5.9. Since *T* enhances significantly when *N* is 1,000 compared to *N* being 3,592 and almost the same accuracy is achieved, 1,000 can be considered as the optimum value of *N* for MLM. For values of *N* less than 1,000, accuracy losses are expected due to the sparsity of MLM. It becomes a challenge to find enough commonly rated cells with decreasing *N* values. That leads to inaccurate results.

As seen from Fig. 5.6, the outcomes are very close to each other for varying *N* values. This phenomenon can be explained the density and the small number of items in Jester. As seen from Fig. 5.8, the outcomes become worse

with decreasing N values for MLM. Although NMAE values significantly degrade while varying N from 1,000 to 500 and so on, accuracy almost becomes stable while varying N from 3,592 to 1,000. The same quality can be achieved using the best 1,000 items instead of using the entire items' ratings. As expected, online performance improves with decreasing N values because amount of data involved in recommendation computations decreases. Since T enhances significantly when N is 1,000 compared to N being 3,592 and almost the same accuracy is achieved, 1,000 can be considered as the optimum value of N for MLM. For values of N less than 1,000, accuracy losses are expected due to the sparsity of MLM. It becomes a challenge to find enough commonly rated cells with decreasing N values. That leads to inaccurate results.

Experiment 3-Effects of Different Non-personalized Values: In order to generate fake values, the scheme proposes to use various methods to determine non-personalized votes, which are utilized to fill sparse data sets and a's ratings vector. Experiments are first performed for evaluating the effects of default values when they are inserted into train sets. Similar tests are applied to give an idea about how the results of the proposed scheme changes when a's ratings vector is filled with different non-personalized values. Various trials are conducted using both data sets. For both data sets, there are 500 users for training and testing, respectively, where γ_j and α_j values are set to 50. Since β_j and δ_j values and unrated items' cells are chosen randomly, such experiments are run for 100 times. NMAE values for MLM and Jester are displayed with various methods of determining non-personalized ratings in Fig. 5.10 and 5.11, respectively.

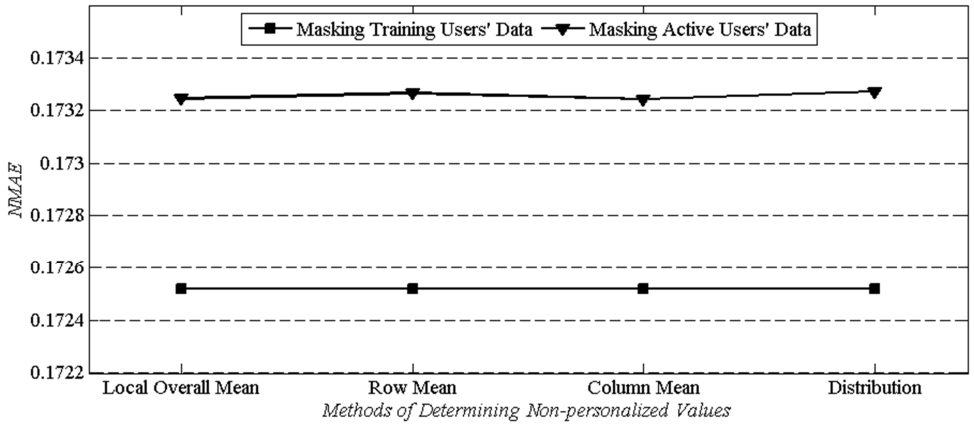

Figure 5.10. NMAE vs. Methods of Determining Non-personalized Values (Jester)

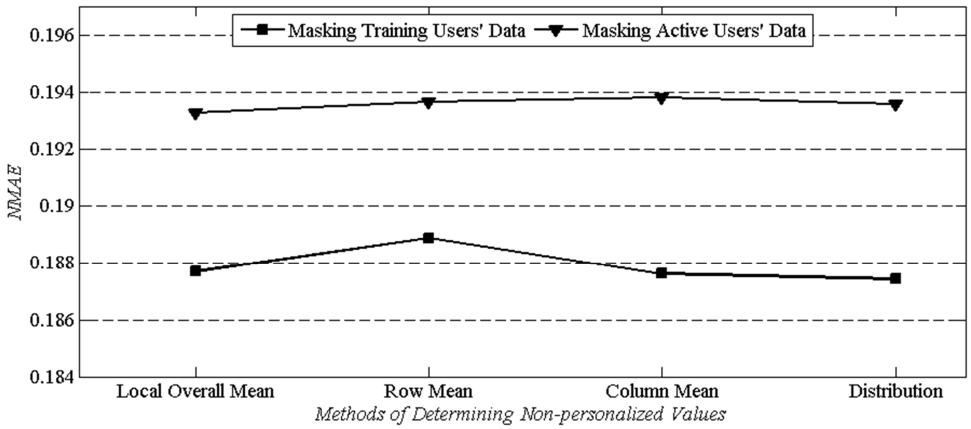

Figure 5.11. NMAE vs. Methods of Determining Non-personalized Values (MLM)

As seen from Fig. 5.10, although all four methods give very similar results for Jester, column mean approach achieves the best results for both disguising train and *a*'s data. Without any disguising, NMAE is 0.1727 for Jester for 500 train users. When train data is disguised with non-personalized ratings generated from using column mean approach, NMAE is about 0.1732. Since Jester is a dense set, due to train data disguising with non-personalized votes, accuracy losses can be considered insignificant. Similarly, when disguising active user's data with non-personalized normalized ratings generated from using column mean approach, NMAE is about 0.1725. In this case, accuracy slightly improves. However, such improvement is trivial.

For MLM, utilizing users' data distribution to produce non-personalized ratings to disguise train data achieves the best results compared to other methods as seen from Fig. 5.11. Without any data disguising, NMAE value is 0.1831 for MLM for 500 train users. When train data is disguised with non-personalized ratings generated from using data distribution approach, NMAE is about 0.1874. Although accuracy becomes worse with data perturbation, losses in NMAE values due to data masking are very small. Local overall average method gives the best results for masking active user's data for MLM. When *a*'s data is perturbed with non-personalized values generated from using local overall average method, NMAE is about 0.1932. Since MLM is a sparse data set compared to Jester, accuracy losses due to data masking are larger. However, such losses are very small and still make it possible to offer accurate predictions.

Experiment 4-Effects of Varying γ_j and α_j Values: The parties fill some of the blank cells in their databases based on γ_j values. In order to show how this affects the results, experiments are performed using both data sets while varying γ_j values. There are 500 users for training and testing, respectively, where the method that gives the best results to generate non-personalized values to fill unrated items' cells is selected. As determined in the previous experiments, generating such values based on data distribution and column mean methods for masking train users' data for MLM and Jester, respectively achieve the best results. Therefore, they are used for data masking. Trials are run for 100 times. Note that the parties hide active users' data by filling some of their empty cells with default values based on α_j values. After assessing how γ_j values affect the outcomes, experiments are performed to evaluate proposed schemes with varying α_j values because such α_j is another factor that might affect accuracy. Experiments are conducted using both data sets while varying α_j values. There are also 500 users as train and test users, respectively, where column mean and local overall average methods are used to determine default values for filling some of a's unrated items' cells for Jester and MLM, respectively because they achieve better results. Experiments are run for 100 times. After computing overall averages, the outcomes displayed for MLM and Jester in Fig. 5.12 and 5.13, respectively.

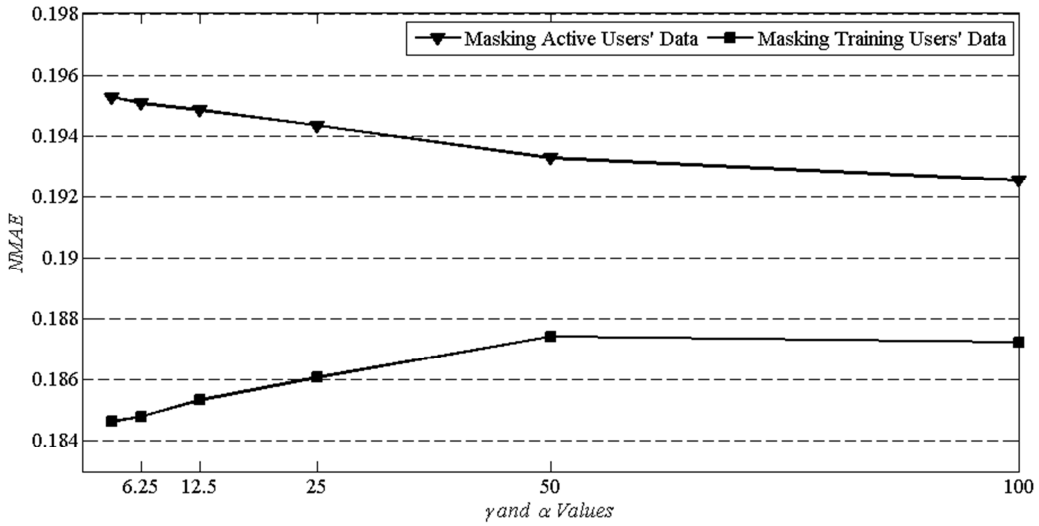

Figure 5.12. NMAE with Varying γ and α Values (MLM)

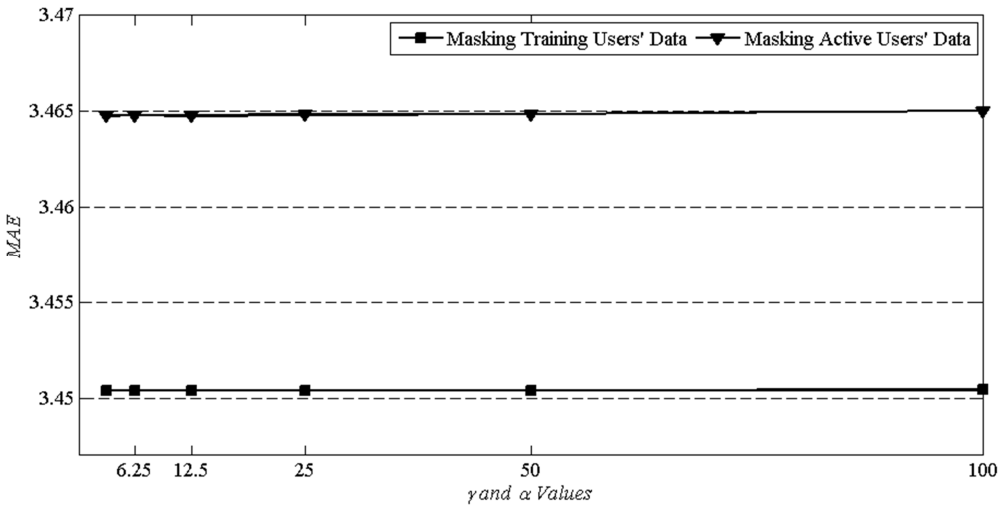

Figure 5.13. MAE with Varying γ and α Values (Jester)

In Fig. 5.12, it is shown how accuracy changes with varying γ and α values for MLM data set. When train users' data are perturbed, NMAE values slightly become worse with increasing γ values from 3.125 to 100. The best results are obtained when γ is 3.125. Although preciseness becomes worse due to data masking, accuracy losses are small, as seen from Fig. 5.12. In case of perturbing active users' data, accuracy slightly enhances with increasing α values. Unlike masking train users' data, α being 100 gives the best results. Unlike MLM, the results are very similar all γ and α values for Jester, as seen from Fig. 5.13. This phenomenon can be explained due the density of Jester. In order to show the minor differences due to varying γ and α values, MAE values are displayed for Jester. Although the outcomes are very similar, the best results are obtained when γ is 12.5 and α is 3.125. For both data sets, the results are very promising for all values of γ and α. The parties can decide their values based on how much accuracy and privacy they want.

Experiment 5-Overall Performance of the Proposed Schemes: After evaluating the effects of various factors separately, finally trials are conducted to assess the overall performance of the proposed schemes. In other words, it is intended to give an idea about the joint effects of privacy and accuracy parameters with varying n values. The obtained results are also compared with the ones based on split and integrated data without privacy concerns. Both data sets are used, where 500 users are utilized as test users. The values of γ_j and α_j are set at their

optimum values that determined in the previous experiments. Those methods are used to determine non-personalized values that give the best results to mask private data. *N* is fixed at its optimum values. Such experiments are run for 100 times. After computing overall averages, MAE values are displayed for both data sets in Fig. 5.14 and 5.15.

With increasing *n* values, the quality of recommendations improves for MLM data set. Although MAE values for *n* values less than 200 are worse when privacy is protected, PPCF on CPD schemes achieve better results for larger *n* values. As seen from Fig. 5.14, the results for split data are the worst due to the insufficient amount of ratings. On the contrary, the outcomes for integrated data are the best, as expected. Due to collaboration, accuracy is expected to become better. However, if data holders offer predictions on their integrated data while preserving their privacy, preciseness becomes worse. For *n* values larger than or equal to 200, the parties are able to offer predictions with decent accuracy using the proposed PPCF on CPD schemes when they own sparse data. Those companies having insufficient data are able to provide more accurate results when they collaborate while preserving their privacy. The improvements are statistically significant because for *n* being 500, the value of *t* is 15.46, which is still greater than the value of *t* for 0.01 in the *t*-table.

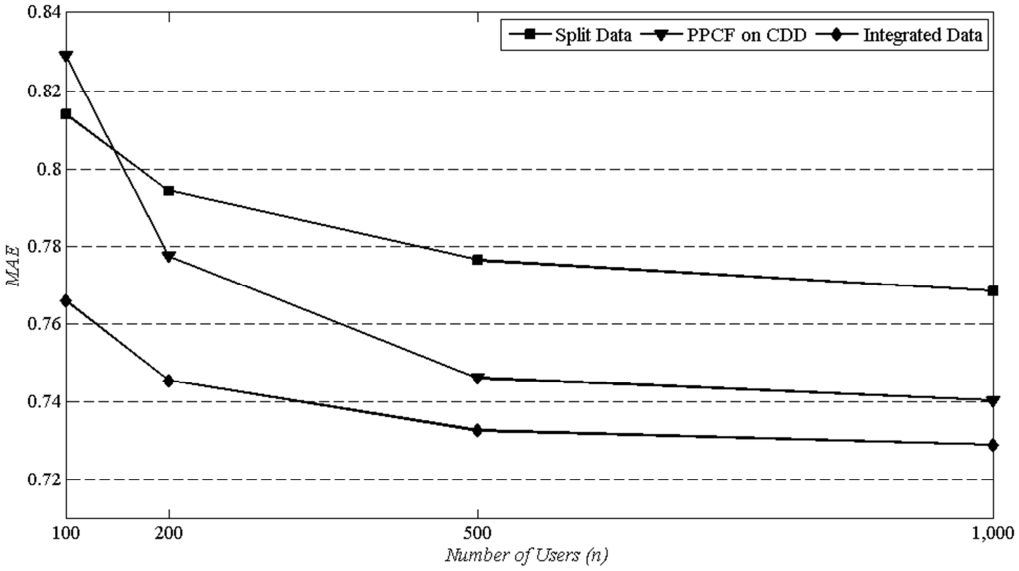

Figure 5.14. Overall Performance of PPCF on CPD with Varying *n* Values (MLM)

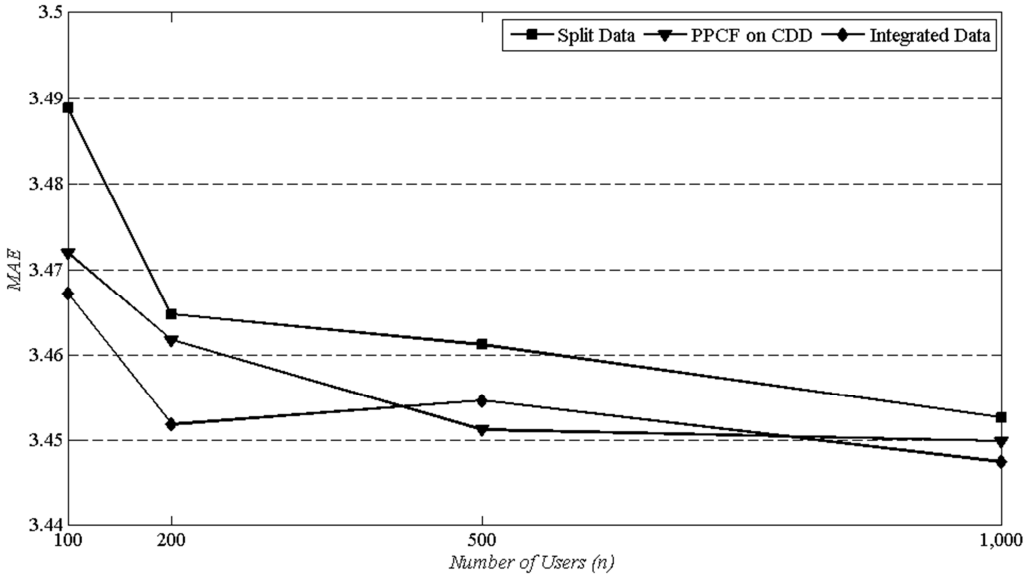

Figure 5.15. Overall Performance of PPCF on CPD with Varying *n* Values (Jester)

According to Fig. 5.15, the similar results are obtained for Jester, as well. MAE values become better with increasing *n* values for all cases for Jester data set. Improvements in accuracy become stable after 500 users. As shown in Fig. 5.15, collaboration between parties improves accuracy. On the other hand, due to privacy protection measures, the quality of the predictions slightly becomes worse. Compared to the results on split data only, the quality of the referrals on integrated data without privacy concerns and the outcomes of proposed privacy-preserving scheme are better. Therefore, the parties are able to offer more accurate referrals based on CPD while preserving their privacy than predictions on split data alone. To determine how significant such accuracy gains, *t*-test is performed. When *n* is 100, the value of *t* is 1.94, which is still greater than the value of *t* for 0.1 in the *t*-table.

The goal of this study is to improve the quality of the recommendations generated based on a hybrid scheme by using more ratings via distributed configuration while expecting smaller losses in accuracy due to privacy preservation. This study mainly focuses on the relative accuracy rather than the absolute accuracy. Thus, the experimental results should be examined with the light of this fact. Experimental results show that proposed privacy-preserving scheme has achieved desired goal.

In the literature, there are various studies comparing different CF approaches. Some examples of such works are, as follows: Zhang et al. (2008) compare user-based, item-based, and hybrid CF algorithms experimentally. In their experimental configuration, they obtain MAEs of 0.7980, 0.7896, and 0.7791 for user-based, item-based, and hybrid techniques, respectively. According to netflixprize.com, the winner algorithm has root mean square error (RMSE) of 0.8567. The scheme is the improved version of algorithm, combined from the matrix factorization and neighborhood methods, proposed by Koren (2008). Goldberg et al. (2001) find NMAE of 0.187 for Jester. In the proposed scheme, for example, when there are 200 users in MLM, the MAE for split data is about 0.7943, while it is 0.7455 for combined data. For the same case, if privacy concerns are taken into account, the MAE is about 0.7760, as seen from Fig. 5.14. Similarly, for proposed scheme the obtained RMSEs are 1.0120, 0.9421, and 0.9627 for split data, integrated data, and privacy-preserving scheme, respectively. As seen from Fig. 5.10, for Jester, NMAE is about 0.173, while proposed scheme's NMAEs of about 0.193 and 0.187 for masking *a*'s data and train data, respectively for MLM. Thus, this scheme is able to offer accurate predictions with privacy. Generally speaking, as seen from given *t*-test results, accuracy gains due to collaboration are significant. Although privacy concerns cause accuracy losses, accuracy gains due to collaboration outweigh such losses. Thus, the parties are able to produce more accurate predictions on integrated data without deeply jeopardizing their privacy than the predictions provided on split data only.

5.7. Chapter Summary

Distributed data-based computations while protecting data owners' privacy are increasingly becoming popular. Some privacy-preserving schemes are presented to provide predictions with decent accuracy on CPD between two companies. At first, one of the hybrid data configurations is introduced between two online vendors, which is the combination of vertical and horizontal partitioning. Similar partitioning called arbitrary partitioning is defined by Jagannathan and Wright (2005). Such partitioning can be considered as a combination of numerous horizontal and vertical partitioning models. Proposed data partitioning model is simpler and special version of arbitrary partitioning, where CPD consists of just a

vertical partitioning of only two different horizontal partitioned databases, or vice versa. The proposed methods are scrutinized in terms of privacy, too. Such schemes prevent the data holders from deriving information about each other's databases. The vendors are not able to learn the exact ratings and the rated items held by each other. Due to privacy protection measures, additional costs are inevitable. Although off-line costs are not that critical for the success of CF systems, the proposed schemes are analyzed in terms of both supplementary off-line and online costs. Such methods cause negligible extra costs; that makes them offer predictions efficiently. To evaluate the proposed schemes in terms of accuracy, a variety of experiments are performed using well-known real data sets. Obtained results show that they still make it possible to produce precise predictions. The results also confirm that integrating data improves both accuracy and coverage. Through experimental results, the optimum values of privacy and accuracy parameters are determined. Their effects on accuracy are demonstrated. The parties can determine the values of such parameters based on how much accuracy and privacy they want. Each party can also variably mask their data. To sum up, the proposed schemes provide accurate predictions efficiently without greatly violating data owners' privacy.

6. CONCLUSIONS AND FUTURE WORK

In this dissertation, a range of privacy-preserving recommender solutions are proposed in novel problem framework caring about arbitrarily partitioned data in the context of P3CF. In this chapter, results attained are summarized to conclude overall text and future directions are highlighted to draw the attentions of PPDM researchers.

6.1. Results

In general perspective, this study investigates the problem of "how two parties provide CF services on arbitrarily partitioned data between them with guaranteeing corporate privacy" and some solutions are presented to respond the research problems listed in Section 1.5. To be satisfactory in terms of efficacy, the proposed solutions should provide pleasing accuracy and coverage, be efficient especially in terms of online response time, and ensure corporate privacy. First of all, it is empirically demonstrated that all the proposed schemes promises accuracy improvements due to contribution of collaborating parties' data even if there are accuracy losses arisen from privacy-preserving process. Accuracy improvements due to collaboration outweigh the losses due to privacy concerns. Such improvements are justified as statistically significant via *t*-test analysis, too. Additionally, the proposed schemes contribute to the coverage of the recommender systems. Secondly, the proposed schemes are theoretically analyzed in terms of supplementary off-line and online costs. Off-line costs are not that critical for the overall success of the CF systems and the off-line tasks can be done in decent time. The schemes bring out conceivable online overheads due to privacy concerns. About efficiency of off-line process, the proposed computations can be performed in plausible time. Finally, the proposed schemes are analyzed in terms of privacy and it is shown that the parties can offer predictions on arbitrarily partitioned data without jeopardizing corporate privacy.

First of all, this dissertation introduces the first P3CF solutions on APD in state-of-the-art with three original research works. Using offered solution in the first of such works (Yakut and Polat, 2012a), two e-commerce parties can provide item-based CF services on APD with corporate privacy. In the second work,

estimating trust-based referrals on APD is also investigated in the context of P3CF and a solution is proposed in this variant. The third study focuses on how to produce referrals on arbitrarily partitioned binary data and NBC-based solution is offered in this variant. As a specific case of APD, CPD is introduced and a solution is offered to produce referrals using hybrid CF algorithm on CPD (Yakut and Polat, 2012b). Certainly, this is the first P3CF solution on CPD.

To compare the proposed item- and trust-based schemes in terms of the prediction quality, experimental findings with respect to varying number of users on MLP are displayed in Table 6.1. Note that the values corresponding to the proposed trust-based CF method is taken for $\beta_j = 20$. According to Table 6.1, with respect to increasing number of users, accuracy gain increases for the proposed item-based algorithm while such gain decreases for the trust-based method. Considering accuracy values,

- For n being 125, 250 and 500; trust-based method outperforms item-based scheme while their accuracy values are the closest for $n = 500$.
- When $n = 943$, item-based scheme gives better prediction quality than trust-based one.

According to above observations, it can be said that the trust-based proposal can be preferred when there are few users while the item-based scheme promotes the prediction quality in case of availability of so many user rating profiles.

Table 6.1. Prediction Quality on Numerical APD (MLP)

n		**125**	**250**	**500**	**943**
Item-based CF	*Split*	0.8013	0.7901	0.7769	0.762
	Proposed	0.8030	0.7766	0.7564	0.738
	Gain (%)	-2.12	1.71	2.64	3.15
Trust-based CF	*Split*	0.8196	0.7935	0.7730	0.7631
	Proposed	0.7757	0.7596	0.7516	0.7470
	Gain (%)	5.36	4.27	2.78	2.12

To evaluate accuracy yields with respect to APD and CPD, empirical outcomes obtained for item-based CF on APD and hybrid CF on CPD with respect to varying number of users on MLM. The outcomes are displayed in Table 6.2. According to Table 6.2., gain tendencies are generally similar and accuracy

improves with increasing number of users. For the smallest *n* values, the proposed schemes fail to promote accuracy even it is valid for item-based CF for *n* = 250. For *n* being 250 or 500, CPD-based proposal outperforms APD-based one in terms of accuracy. However, the best accuracy results are observed via the item-based CF on APD for *n* = 1,000.

Table 6.2. Prediction Quality: APD vs. CPD (MLM)

	Item-based CF on APD				*Hybrid CF on CPD*			
n	**125**	**250**	**500**	**1,000**	**100**	**200**	**500**	**1,000**
Split	0.7919	0.7798	0.7625	0.7464	0.8139	0.7943	0.7765	0.7685
Proposed	0.7946	0.7883	0.7507	0.7221	0.8194	0.7761	0.7479	0.7421
Gain (%)	-0.34	-1.09	1.55	3.26	-0.67	2.30	3.68	3.43

6.2. Future Work

Via this study, arbitrarily partitioned data is placed in P3CF literature. As future research directions in this variant, the following issues can be considered. Data might be arbitrarily distributed among more than two parties. It should be studied how to offer accurate predictions efficiently when ratings are arbitrarily distributed among multiple parties while preserving their privacy. To offer a solution for such e-commerce companies, the proposed schemes can be extended to multi-party schemes. However, some modifications are needed; and such modifications and their consequences can be investigated as a future work.

One important issue that should be addressed is data overlapping. Along the study, it is assumed that ratings are shared in mutually exclusive manner. Even with this assumption, the focused problems are still challenging because the solutions should achieve privacy, accuracy, and performance at the same time. However, in real life scenarios, overlapping ratings are inevitable. It is a proper research task to scrutinize how to handle such overlapping and to show performance changes with different amounts of overlapping data. The effects of overlapping ratings on secrecy and accuracy should also be examined. Furthermore, new approaches can be invented for data masking so that the parties can avoid having cells with double ratings. In order to improve the overall performance, some aggregate values can be disclosed. It should be scrutinized

whether this can be possible or not; and if so, how this affects the overall performance.

The parties are assumed as semi-honest in the defined problem framework, but in the PPDM literature there are works operating with malicious participants. Semi-honest is quite realistic in many situations while malicious model-based studies concern preventing any malicious behavior done by participants of process by utilizing more expensive cryptographic techniques (Kantarcioglu and Kardes, 2008). It is in place future work to investigate P3CF on arbitrarily partitioned data in malicious model settings.

It is still an interesting topic to investigate how to provide predictions using pure model- or memory-based CF algorithms based on CPD with corporate privacy. It can be studied how to apply methods and protocols proposed in this study to such algorithms. Moreover, a study can be conducted looking for whether it is possible to offer referrals based on binary ratings, which are cross partitioned between two parties while preserving corporate privacy. Furthermore, there are some CF proposals (Mild and Reutterer, 2001; Lee et al., 2005) on market basket data consisting of transacted items only. It can be congruous research direction to investigate P3CF on arbitrarily partitioned market basket data.

Some future works can be also revealed about topics sideward to studies in this dissertation. It is assumed that user and item IDs are publicly known. More studies can be conducted to improve the proposed schemes in such a way that such IDs are protected, as well. It can also be studied how to tackle the cumbersome of encryptions performed during online phase. In order to enhance the online efficiency of the proposed schemes, parallel computing-based, application-oriented hardware, and some other techniques can be applied to these schemes. Such proposals and implementations can constitute a range of future studies, too.

Finally, considered partitioning configurations with additional probable partitioning configurations can be applied to realize various data mining tasks. For example, although privacy-preserving schemes to achieve clustering, back-propagation neural network learning, and decision tree tasks based on APD have been proposed (Jagannathan and Wright, 2005; Han and Ng, 2007; Prasad and

Rangan, 2007; Bansal et al., 2010), how to perform classification, association rule mining, and regression analysis on APD while preserving confidentiality is still an open question. For example, in this study, NBC-based CF on arbitrarily partitioned binary data is investigated; similarly, NBC can be performed on classification tasks using categorical data. One more research topic can be "how classification tasks can be realized on arbitrarily partitioned categorical data using naïve Bayesian classifier."

REFERENCES

Ackerman, M.S., Cranor, L.F. and Reagle, J. (1999), "Privacy in e-commerce: Examining user scenarios and privacy preferences," *Proceedings of the 1st ACM Conference on Electronic Commerce*, Denver, CO, USA, 1-8.

Agrawal, R. and Srikant, R. (2000), "Privacy-preserving data mining," *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, USA, 439-450.

Ahmad, W. and Khokhar, A. (2007), "An architecture for privacy-preserving collaborative filtering on Web portals," *Proceedings of the 3rd International Symposium on Information Assurance and Security*, Manchester, UK, 273-278.

Ahn, H.J. (2008), "A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem," *Information Sciences*, **178** (1), 37-51.

Aïmeur, E., Brässard, G., Fernandez, J.M. and Onana, F.S.M. (2008), "Alambic: A privacy-preserving recommender system for electronic commerce," *International Journal of Information Security*, **7** (5), 307-334.

Amatriain, X., Jaimes, A., Oliver, N. and Pujol, J.M. (2011), "Data mining methods for recommender systems," *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira and P. B. Kantor, Springer, New York, NY, USA.

Balabanovic, M. and Shoham, Y. (1997), "Fab: Content-based collaborative recommendation," *Communications of the ACM*, **40** (3), 66-72.

Bansal, A., Chen, T. and Zhong, S. (2010), "Privacy-preserving Back-propagation neural network learning over arbitrarily partitioned data," *Neural Computing & Applications*, **20** (1), 1-8.

Basu, A., Kikuchi, H. and Vaidya, J. (2011a), "Privacy-preserving weighted Slope One predictor for item-based collaborative filtering," *Proceedings of the International Workshop on Trust and Privacy in Distributed Information Processing*, Copenhagen, Denmark.

Basu, A., Vaidya, J., Dimitrakos, T. and Kikuchi, H. (2012a), "Feasibility of a privacy-preserving collaborative filtering scheme on the Google App

Engine -- a performance case-study," *Proceedings of the 27th ACM Symposium on Applied Computing*, Trento, Italy, 141-146.

Basu, A., Vaidya, J. and Kikuchi, H. (2012b), "Perturbation-based privacy-preserving Slope One predictors for collaborative filtering," *Proceedings of the 6th IFIP International Conference on Trust Management*, Surat, India.

Basu, A., Vaidya, J., Kikuchi, H. and Dimitrakos, T. (2011b), "Privacy-preserving collaborative filtering for the cloud," *Proceedings of the 2011 IEEE 3rd International Conference on Cloud Computing Technology and Science*, Athens, Greece, 223-230.

Berkovsky, S., Borisov, N., Eytani, Y., Kuflik, T. and Ricci, F. (2007), "Examining users' attitude towards privacy-preserving collaborative filtering," *Proceedings of the Workshop on Data Mining for User Modeling, in conjunction with UM*, Corfu, Greece, 16-22.

Berkovsky, S., Eytani, Y., Kuflik, T. and Ricci, F. (2005), "Privacy-enhanced collaborative filtering," *Proceedings of the User Modeling Workshop on Privacy-Enhanced Personalization*, Edinburgh, UK, 75-84.

Berkovsky, S. and Kuflik, T. (2006), "Hierarchical neighborhood topology for privacy enhanced collaborative filtering," *Proceedings of the CHI06 Workshop on Privacy-Enhanced Personalization*, Montreal, Canada, 6-13.

Bilge, A. and Polat, H. (2010), "Improving privacy-preserving NBC-based recommendations by preprocessing," *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, Toronto, Canada, 143-147.

Bilge, A. and Polat, H. (2011), "An improved profile-based CF scheme with privacy," *Proceedings of the 2011 5th IEEE International Conference on Semantic Computing*, Palo Alto, CA, USA, 133-140.

Bilge, A. and Polat, H. (2012), "An improved privacy-preserving DWT-based collaborative filtering scheme," *Expert Systems with Applications*, **39** (3), 3841-3854.

Bogdanova, G. and Georgieva, T. (2008), "Using error-correcting dependencies for collaborative filtering," *Data & Knowledge Engineering*, **66** (3), 402-413.

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

Brun, A. and Boyer, A. (2009), "Towards privacy compliant and anytime recommender systems," *Lecture Notes in Computer Science*, **5692,** 276-287.

Calandrino, J.A., Kilzer, A., Narayanan, A., Felten, E.W. and Shmatikov, V. (2011), "'You might also like:' Privacy risks of collaborative filtering," *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 231-246.

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

Canny, J. (2002b), "Collaborative filtering with privacy via factor analysis," *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 238-245.

Chang, H.-J., Hung, L.-P. and Ho, C.-L. (2007), "An anticipation model of potential customers' purchasing behavior based on clustering analysis and association rules analysis," *Expert Systems with Applications*, **32** (3), 753-764.

Chen, S. and Williams, M.-A. (2010), "Towards a comprehensive requirements architecture for privacy-aware social recommender systems," *Proceedings of the 7th Asia-Pacific Conference on Conceptual Modelling*, Brisbane, Australia, **110**, 33-42.

Cheng, Z. and Hurley, N. (2009), "Trading robustness for privacy in decentralized recommender systems," *Proceedings of the Twenty-First Conference on Innovative Applications of Artificial Intelligence*, Pasadena, CA, USA, 79-84.

Chirita, P.-A., Nejdl, W. and Zamfir, C. (2005), "Preventing shilling attacks in online recommender systems," *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, Bremen, Germany, 67-74.

Cissée, R. and Albayrak, S. (2007), "An agent-based approach for privacy-preserving recommender systems," *Proceedings of the 6th International joint Conference on Autonomous Agents and Multiagent Systems*, Honolulu, Hawaii, 1-8.

Cranor, L.F. (2004), "'I didn't buy it for myself', Privacy and e-commerce personalization," *Designing Personalized User Experiences in eCommerce*, C.-M. Karat, J. O. Blom and J. Karat, Norwell, MA, USA, Kluwer Academic Publishers, 57-73.

Crypto++. (2009). "5.6.0 Benchmarks." Retrieved 13/02/2012, from http://www.cryptopp.com/benchmarks.html.

Dokoohaki, N., Kaleli, C., Polat, H. and Matskin, M. (2010), "Achieving optimal privacy in trust-aware social recommender systems," *Proceedings of the 2nd International Conference on Social Informatics*, Laxenburg, Austria, 62-79.

Emekci, F., Sahin, O.D., Agrawal, D. and El Abbadi, A. (2007), "Privacy-preserving decision tree learning over multiple parties," *Data & Knowledge Engineering*, **63** (2), 348-361.

Erkin, Z., Veugen, T. and Lagendijk, R.L. (2011), "Generating private recommendations in a social trust network," *Proceedings of the International Conference on Computational Aspects of Scoial Networks*, Salamanca, Spain, 82-87.

Even, S., Goldreich, O. and Lempel, A. (1985), "A randomized protocol for signing contracts," *Communications of the ACM*, **28** (6), 637-647.

Fogel, J. and Nehmad, E. (2009), "Internet social network communities: Risk taking, trust, and privacy concerns," *Computers in Human Behavior*, **25** (1), 153-160.

Geenen, P.L., van der Gaag, L.C., Loeffen, W.L. A. and Elbers, A.R.W. (2011), "Constructing naïve Bayesian classifiers for veterinary medicine: A case

study in the clinical diagnosis of classical swine fever," *Research in Veterinary Science*, **91** (1), 64-70.

Goldberg, D., Nichols, D., Oki, B.M. and Terry, D. (1992), "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, **35** (12), 61-70.

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

Goldwasser, S. and Micali, S. (1984), "Probabilistic encryption," *Journal of Computer System Sciences*, **28** (2), 270-299.

GroupLens. "MovieLens Data Sets," Retrieved 20/04/2012, from http://www.grouplens.org/node/73.

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

Han, S. and Ng, W.K. (2007), "Multi–party privacy-preserving decision trees for arbitrarily partitioned data," *International Journal of Intelligent Control and Systems*, **12** (4), 351-358.

Herlocker, J.L., Konstan, J.A., Borchers, A. and Riedl, J.T. (1999), "An algorithmic framework for performing collaborative filtering," *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, USA, 230-237.

Hoens, T.R., Blanton, M. and Chawla, N.V. (2010a), "A private and reliable recommendation system for social networks," *Proceedings of the 2010 IEEE 2nd International Conference on Social Computing*, Minneapolis, MN, USA, 816-825.

Hoens, T.R., Blanton, M. and Chawla, N.V. (2010b), "Reliable medical recommendation systems with patient privacy," *Proceedings of the 1st*

ACM International Health Informatics Symposium, Arlington, VA, USA, 173-182.

Hsieh, C.L.A., Zhan, Z., Zeng, D. and Feiyue, W. (2008), "Preserving privacy in joining recommender systems," *Proceedings of the International Conference on Information Security and Assurance*, Busan, Korea, 561-566.

Huang, C.-Y., Shen, Y.-C., Chiang, I.-P. and Lin, C.-S. (2007), "Characterizing Web users' online information behavior," *Journal of the American Society for Information Science and Technology*, **58** (13), 1988-1997.

Hurley, N. and Zhang, M. (2011), "Novelty and diversity in top-*N* recommendation -- Analysis and evaluation," *ACM Transactions on Internet Technology*, **10** (4), 1-30.

Hwang, C.-S. and Chen, Y.-P. (2007), "Using trust in collaborative filtering recommendation," *Proceedings of the 20th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems*, Kyoto, Japan, 1052-1060.

Inan, A., Kaya, S.V., Saygin, Y., Savas, E., Hintoglu, A.A. and Levi, A. (2007), "Privacy-preserving clustering on horizontally partitioned data," *Data & Knowledge Engineering*, **63** (3), 646-666.

Jae-wook, A. and Amatriain, X. (2010), "Towards fully distributed and privacy-preserving recommendations via expert collaborative filtering and RESTful linked data," *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Toronto, Canada, 66-73.

Jagannathan, G. and Wright, R.N. (2005), "Privacy-preserving distributed *k*-means clustering over arbitrarily partitioned data," *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, Chicago, IL, USA, 593-599.

Ji, A.-T., Yeon, C., Kim, H.-N. and Jo, G.-S. (2007), "Distributed collaborative filtering for robust recommendations against shilling attacks," *Proceedings of the 20th Conference of the Canadian Society for Computational Studies*

of Intelligence on Advances in Artificial Intelligence, Montreal, Quebec, Canada, 14-25.

Judith, D. (2008). "Privacy," Retrieved 25/04/2012, from <http://plato.stanford.edu/archives/fall2008/entries/privacy/>.

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

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

Kaleli, C. and Polat, H. (2009), "Similar or dissimilar users? or both?," *Proceedings of the 2nd International Symposium on Electronic Commerce and Security*, Nanchang, China, 184-189.

Kaleli, C. and Polat, H. (2010), "P2P collaborative filtering with privacy," *Turkish Journal of Electrical Engineering and Computer Sciences*, **18**, 101-116.

Kaleli, C. and Polat, H. (2011), "Privacy-preserving trust-based recommendations on vertically distributed data," *Proceedings of the 5th IEEE International Conference on Semantic Computing*, Palo Alto, CA, USA, 376-379.

Kaleli, C. and Polat, H. (2012a), "Privacy-preserving SOM-based recommendations on horizontally distributed data," *Knowledge-based Systems*, DOI: 10.1016/j.knosys.2012.02.013.

Kaleli, C. and Polat, H. (2012b), "SOM-based recommendations with privacy on multi-party vertically distributed data," *Journal of Operational Research Society*, **63**, 826-838.

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

Kantarcioglu, M. and Kardes, O. (2008), "Privacy-preserving data mining in the malicious model," *International Journal of Information and Computer Security*, **2** (4), 353-375.

Kantarcioglu, M. and Vaidya, J. (2003), "Privacy-preserving naïve Bayes classifier for horizontally partitioned data," *Proceedings of the Workshop*

on Privacy Preserving Data Mining held in association with the 3rd IEEE International Conference on Data Mining, Melbourne, FL, USA.

Katzenbeisser, S. and Petkovic, M. (2008), "Privacy-preserving recommendation systems for consumer healthcare services," *Proceedings of the 2008 3rd International Conference on Availability, Reliability and Security*, Barcelona, Spain, 889-895.

Kaya, H. and Alpaslan, F.N. (2010), "Using social networks to solve data sparsity problem in one-class collaborative filtering," *Proceedings of the 2010 7th International Conference on Information Technology: New Generations*, Las Vegas, NV, USA, 249-252.

Kaya, S., Pedersen, T., Savas, E. and Saygin, Y. (2009), "Efficient privacy-preserving distributed clustering based on secret sharing," *Emerging Technologies in Knowledge Discovery and Data Mining*, **4819**, 280-291.

Keshavamurthy, B.N., Sharma, M. and Toshniwal, D. (2010), "Privacy preservation naïve Bayes classification for a vertically distribution scenario using trusted third party," *Proceedings of the 2010 International Conference on Advances in Recent Technologies in Communication and Computing*, Kerala, India, 404-407.

Kobsa, A. (2007), "Privacy-enhanced personalization," *Communications of the ACM*, **50** (8), 24-33.

Koren, Y. (2008), "Factorization meets the neighborhood: A multifaceted collaborative filtering model," *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, NV, USA, 426-434.

Lam, S.K.T., Frankowski, D. and Riedl, J.T. (2006), "Do you trust your recommendations? An exploration of security and privacy issues in recommender systems," *Proceedings of the 2006 International Conference on Emerging Trends in Information and Communication Security*, Freiburg, Germany, 14-29.

Lathia, N., Hailes, S. and Capra, L. (2007), "Private distributed collaborative filtering using estimated concordance measures," *Proceedings of the 2007 ACM Conference on Recommender Systems*, Minneapolis, MN, USA, 1-8.

Lee, C.-H. (2006), "A semi-naïve Bayesian learning method for utilizing unlabeled data," *Knowledge-Based Intelligent Information and Engineering Systems*, **4251**, 187-194.

Lee, J.-S., Jun, C.-H., Lee, J. and Kim, S. (2005), "Classification-based collaborative filtering using market basket data," *Expert Systems with Applications*, **29** (3), 700-704.

Lemire, D. and Maclachlan, A. (2005), "Slope One predictor for online rating-based collaborative filtering," *Proceedings of the SIAM International Conference on Data Mining*, New Port Bridge, CA, USA, 471-475.

Li, C., Ma, L. and Dong, K. (2009), "Collaborative filtering cold-start recommendation based on dynamic browsing tree model in e-commerce," *Proceedings of the 2009 International Conference on Web Information Systems and Mining*, Shangai, China, 620-624.

Li, D., Lv, Q., Shang, L. and Gu, N. (2011a), "YANA: An efficient privacy-preserving recommender system for online social communities," *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, Glasgow, Scotland, 2269-2272.

Li, D., Lv, Q., Xia, H., Shang, L., Lu, T. and Gu, N. (2011b), "Pistis: A privacy-preserving content recommender system for online social communities," *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Lyon, France, 79-86.

Li, L., Huang, L. and Yang, W. (2011c), "Privacy-preserving outlier detection over arbitrarily partitioned data," *Proceedings of the International Symposium on Information Engineering and Electronic Commerce,* Huangshi, China, 103-106.

Machanavajjhala, A., Korolova, A. and Sarma, A.D. (2011), "Personalized social recommendations: Accurate or private," *Proceedings of the VLDB Endowment*, **4** (7), 440-450.

Massa, P. and Avesani, P. (2007), "Trust-aware recommender systems," *Proceedings of the 2007 ACM Conference on Recommender Systems*, Minneapolis, MN, USA, 17-24.

Massa, P. and Bhattacharjee, B. (2004), "Using trust in recommender systems: An experimental analysis trust management," *Lecture Notes in Computer Science*, **2995,** 221-235.

Mild, A. and Reutterer, T. (2001), "Collaborative filtering methods for binary market basket data analysis," *Active Media Technology*, **2252**, 302-313.

Miller, B.N., Konstan, J.A. and Riedl, J.T. (2004), "PocketLens: Toward a personal recommender system," *ACM Transactions on Information Systems*, **22** (3), 437-476.

Miyahara, K. and Pazzani, M.J. (2000), "Collaborative filtering with the simple Bayesian classifier," *Proceedings of the 6th Pacific Rim International Conference on Artificial Intelligence*, Melbourne, Australia, 679-689.

Naccache, D. and Stern, J. (1998), "A new public key cryptosystem based on higher residues," *Proceedings of the 5th ACM Conference on Computer and Communications Security*, San Francisco, CA, USA, 59-66.

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

Naor, M. and Pinkas, B. (2001), "Efficient oblivious transfer protocols," *Proceedings of the Symposium on Discrete Algorithms*, Washington, DC, USA, 448-457.

Obviex. (2011). "How to calculate the size of encrypted data?" Retrieved 9/13/2011, from http://www.obviex.com/Articles/CiphertextSize.aspx.

OECD (2000). Organization for Economic Co-operation and Development

OECD (2005). Guidelines on the Protection of Privacy and Transborder Flows of Personal Data.

Oliveira, S.R.M. and Zaïane, O.R. (2007), "A privacy-preserving clustering approach toward secure and effective data analysis for business collaboration," *Computers & Security*, **26** (1), 81-93.

Paillier, P. (1999), "Public key cryptosystems based on composite degree residuosity classes," *Lecture Notes in Computer Science*, **1592**, 223-238.

Papagelis, M., Plexousakis, D. and Kutsuras, T. (2005), "Alleviating the sparsity problem of collaborative filtering using trust inferences," *Proceedings of*

the 3rd International Conference on Trust Management, Paris, France, 224-239.

Parameswaran, R. and Blough, D.M. (2007), "Privacy-preserving collaborative filtering using data obfuscation," *Proceedings of the IEEE International Conference on Granular Computing*, San Jose, CA, USA, 380-380.

Pedersen, T.B., Saygin, Y. and Savas, E. (2007), "Secret sharing vs. encryption-based techniques for privacy-preserving data mining," *Proceedings of the Joint UNECE/Eurostat Work Session on Statistical Disclosure Control*, Manchester, UK, 17-19.

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

Pohlig, S. and Hellman, M. (1978), "An improved algorithm for computing logarithms over GF(*p*) and its cryptographic significance (Corresp.)," *IEEE Transactions on Information Theory*, **24** (1), 106-110.

Polat, H. and Du, W. (2005a), "Privacy-preserving collaborative filtering," *International Journal of Electronic Commerce*, **9** (4), 9-35.

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

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

Polat, H. and Du, W. (2006), "Achieving private recommendations using randomized response techniques," *Lecture Notes in Computer Science*, **3918**, 637-646.

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

Polat, H. and Du, W. (2008), "Privacy-preserving top-*N* Recommendation on distributed data," *Journal of the American Society for Information Science and Technology*, **59** (7), 1093-1108.

Prasad, P.K. and Rangan, C.P. (2007), "Privacy-preserving BIRCH algorithm for clustering over arbitrarily partitioned databases," *Proceedings of the 3rd International Conference on Advanced Data Mining and Applications*, Harbin, China, 146-157.

Rabin, M.O. (1981). How to exchange secrets by oblivious transfer, Aiken Computation Laboratory, Harvard University Technical Report No. 81, Cambridge, MA, USA.

Ramakrishnan, N., Keller, B.J., Mirza, B.J., Grama, A.Y. and Karypis, G. (2001), "Privacy risks in recommender systems," *IEEE Internet Computing*, **5** (6), 54-62.

Ray, S. and Mahanti, A. (2009), "Strategies for effective shilling attacks against recommender systems," *Lecture Notes in Computer Science*, **5456**, 111-125.

Rozenberg, B. and Gudes, E. (2006), "Association rules mining in vertically partitioned databases," *Data & Knowledge Engineering*, **59** (2), 378-396.

Sarwar, B., Karypis, G., Konstan, J.A. and Reidl, J.T. (2001), "Item-based collaborative filtering recommendation algorithms," *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, Hong Kong, 285-295.

Sarwar, B.M., Karypis, G., Konstan, J.A. and Riedl, J.T. (2000), "Application of dimensionality reduction in recommender systems–a case study," *Proceedings of the ACM WebKDD, Web Mining for E-commerce Workshop*, Boston, MA, USA.

Shapira, B., Elovici, Y., Meshiach, A. and Kuflik, T. (2005), "PRAW- PRivAcy model for the Web: Research Articles," *Journal of the American Society for Information Science and Technology*, **56** (2), 159-172.

Shokri, R., Pedarsani, P., Theodorakopoulos, G. and Hubaux, J.-P. (2009), "Preserving privacy in collaborative filtering through distributed

aggregation of offline profiles," *Proceedings of the 3rd ACM Conference on Recommender Systems*, New York, NY, USA, 157-164.

Skarkala, M.E., Maragoudakis, M., Gritzalis, S. and Mitrou, L. (2011), "Privacy-preserving tree augmented naïve Bayesian multi-party implementation on horizontally partitioned databases," *Proceedings of the 8th International Conference on Trust, Privacy and Security in Digital Business*, Toulouse, France, 62-73.

Skillicorn, D.B. and McConnell, S.M. (2008), "Distributed prediction from vertically partitioned data," *Journal of Parallel and Distributed Computing*, **68** (1), 16-36.

Spiekermann, S., Grossklags, J. and Berendt, B. (2001), "E-privacy in 2nd generation e-commerce: Privacy preferences versus actual behavior," *Proceedings of the 3rd ACM Conference on Electronic Commerce*, Tampa, FL, USA, 38-47.

Su, X. and Khoshgoftaar, T.M. (2006), "Collaborative filtering for multi-class data using belief nets algorithms," *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, Washington, DC, USA, 497-504.

Su, X. and Khoshgoftaar, T.M. (2009), "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, **2009**, 2-21.

Tada, M., Kikuchi, H. and Puntheeranurak, S. (2010), "Privacy-preserving collaborative filtering protocol based on similarity between items," *Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, Perth, Australia, 573-578.

Upmanyu, M., Namboodiri, A., Srinathan, K. and Jawahar, C. (2010), "Efficient privacy-preserving *k*-means clustering," *Intelligence and Security Informatics*, **6122**, 154-166.

Vaidya, J., Clifton, C., Kantarcioglu, M. and Patterson, A.S. (2008a), "Privacy-preserving decision trees over vertically partitioned data," *ACM Transactions on Knowledge Discovery from Data*, **2** (3), 1-27.

Vaidya, J., Kantarcioglu, M. and Clifton, C. (2008b), "Privacy-preserving naïve Bayes classification," *The VLDB Journal*, **17** (4), 879-898.

Van den Poel, D. and Buckinx, W. (2005), "Predicting online-purchasing behaviour," *European Journal of Operational Research*, **166** (2), 557-575.

Verhaegh, W., van Duijnhoven, A., Tuyls, P. and Korst, J. (2004), "Privacy protection in memory-based collaborative filtering," *Ambient Intelligence*, **3295**, 61-71.

Vozalis, M.G. and Margaritis, K.G. (2007), "Using SVD and demographic data for the enhancement of generalized collaborative filtering," *Information Sciences*, **177** (15), 3017-3037.

Warren, S. and Brandeis, L. (1890), "The right to privacy," *Harward Law Review 193*, **IV**(5).

Westin, A. (1967), "*Privacy and Freedom*," Atheneum, Newyork, NY, USA.

Yakut, I. and Polat, H. (2007a), "Achieving private SVD-based recommendations on inconsistently masked data," *Proceedings of the International Conference on Security of Information and Networks*, Famagusta, Cyprus, 172-177.

Yakut, I. and Polat, H. (2007b), "Privacy-preserving Eigentaste-based collaborative filtering," *Proceedings of the International Workshop on Security*, Nara, Japan, 169-184.

Yakut, I. and Polat, H. (2010), "Privacy-preserving SVD-based collaborative filtering on partitioned data," *International Journal of Information Technology and Decision Making*, **9** (3), 473-502.

Yakut, I. and Polat, H. (2012a), "Arbitrarily distributed data-based recommendations with privacy," *Data & Knowledge Engineering*, **72**, 239-256.

Yakut, I. and Polat, H. (2012b), "Privacy-preserving hybrid collaborative filtering on cross distributed data," *Knowledge and Information Systems*, **30** (2), 405-433.

Yang, W. and Huang, S. (2008), "Data privacy protection in multi-party clustering," *Data & Knowledge Engineering*, **67** (1), 185-199.

Yao, A. C.-C. (1986), "How to generate and exchange secrets," *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, Atlanta, GA, USA, 162-167.

Yi, X. and Zhang, Y. (2007), "Privacy-preserving distributed association rule mining via semi-trusted mixer," *Data & Knowledge Engineering*, **63** (2), 550-567.

Yi, X. and Zhang, Y. (2009), "Privacy-preserving naïve Bayes classification on distributed data via semi-trusted mixers," *Information Systems*, **34** (3), 371-380.

Youn, E. and Jeong, M.K. (2009), "Class dependent feature scaling method using naive Bayes classifier for text datamining," *Pattern Recognition Letters*, **30** (5), 477-485.

Yunhong, H., Guoping, H., Liang, F. and Jingyong, T. (2010), "Privacy-preserving SVM classification on arbitrarily partitioned data," *Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing*, Shangai, China, 67-71.

Zhan, J., Wang, I.C., Hsieh, C.-L., Hsu, T.-S., Liau, C.-J. and Wang, D.-W. (2008), "Towards efficient privacy-preserving collaborative recommender systems," *Proceedings of the IEEE International Conference on Granular Computing*, Hangzhou, China, 778-783.

Zhang, J., Ghorbani, A.A. and Cohen, R. (2007), "A familiarity-based trust model for effective selection of sellers in multiagent e-commerce systems," *International Journal of Information Security*, **6** (5), 333-344.

Zhang, L., Xiao, B. and Guo, J. (2008), "A hybrid approach to collaborative filtering for overcoming data sparsity," *Proceedings of the 9th International Conference on Signal Processing,* Beijing, China, 1595-1599.