

Privacy-preserving SOM-based recommendations on horizontally distributed data

Cihan Kaleli, Huseyin Polat*

Computer Engineering Department, Anadolu University, Eskisehir 26470, Turkey

ARTICLE INFO

Article history:

Received 13 December 2011

Received in revised form 31 January 2012

Accepted 19 February 2012

Available online 9 March 2012

Keywords:

Privacy

Distributed data

Clustering

Recommendation

Performance

ABSTRACT

To produce predictions with decent accuracy, collaborative filtering algorithms need sufficient data. Due to the nature of online shopping and increasing amount of online vendors, different customers' preferences about the same products can be distributed among various companies, even competing vendors. Therefore, those companies holding inadequate number of users' data might decide to combine their data in such a way to present accurate predictions with acceptable online performance. However, they do not want to divulge their data, because such data are considered confidential and valuable. Furthermore, it is not legal disclosing users' preferences; nevertheless, if privacy is protected, they can collaborate to produce correct predictions.

We propose a privacy-preserving scheme to provide recommendations on horizontally partitioned data among multiple parties. In order to improve online performance, the parties cluster their distributed data off-line without greatly jeopardizing their secrecy. They then estimate predictions using k -nearest neighbor approach while preserving their privacy. We demonstrate that the proposed method preserves data owners' privacy and is able to suggest predictions resourcefully. By performing several experiments using real data sets, we analyze our scheme in terms of accuracy. Our empirical outcomes show that it is still possible to estimate truthful predictions competently while maintaining data owners' confidentiality based on horizontally distributed data.

© 2012 Elsevier B.V. All rights reserved.

1. Introduction

Rapid improvements in the Internet technology help people purchase several kinds of products through the Internet facilities. Due to its attractiveness, many online vendors have been founded to promote online shopping. To facilitate their customers choose the right products, e-commerce sites employ Collaborative Filtering (CF) schemes because selecting appropriate products to purchase becomes a challenging problem as number of choices increases [1]. In addition to recommending various products like books, movies, music CDs, and so on, CF systems are also used to suggest web pages.

The basic steps in CF process are, as follows [10,17]: After collecting users' likings about various items, an $n \times m$ user-item matrix (D) is created, where n and m represent number of users and items, respectively. CF schemes then estimate similarities between users in their database and an active user (a) who is looking for a prediction for a target item (q). Next, they determine neighbors of the active user a (the best k similar users) according to the similarity weights. Finally, a weighted average of their ratings on the target item q is calculated.

One of the main purposes of CF systems is to offer truthful and reliable referrals. To produce precise and dependable predictions, such systems should collect ratings from enough number of users. When online vendors own limited number of users' data, it becomes a challenge to form reliable and large enough neighborhoods; that might cause low quality CF services. Additionally, inadequate number of users' data lead to cold start problem, where e-commerce sites can recommend predictions for limited number of items. That might cause to lose customers due to the lack of accuracy in the recommendations received [6]. Therefore, holding sufficient number of users' ratings is imperative for the overall success of CF systems.

Some companies, especially recently established ones, might not have enough users' data for recommendation purposes. Moreover, customers may prefer different online vendors for shopping. In other words, different users purchase the same products from different companies and they can request referrals from corresponding vendors. Consequently, ratings of the same items collected from many users for CF purposes might be horizontally partitioned among multiple vendors. For example, some clients purchase books from Amazon.com and some prefer Barnes & Noble.com, while others get them from Borders, and so on. These book sellers' databases may include ratings for the same books recorded from disjoint sets of customers, and these can be jointly used for better referrals. Notice that this does not mean that online

* Corresponding author. Tel.: +90 222 321 35 50; fax: +90 222 323 95 01.

E-mail addresses: ckaleli@anadolu.edu.tr (C. Kaleli), polath@anadolu.edu.tr (H. Polat).

Due to the increasing popularity of privacy in e-commerce applications, PPCF is receiving increasing attention. Canny [8,9] proposes PPCF schemes in which users control all of their own private data using some cryptographic approaches. Polat and Du [35] utilize Randomized Response Techniques (RRTs) to perturb users' data while still producing binary ratings-based referrals with decent accuracy. The users either send their true ratings or the exact opposite of their ratings with a probability. In another study, Polat and Du [36] propose a PPCF scheme based on inconsistently masked data. Each user variably disguises their private data using different methods. Their scheme is still able to make it to offer predictions from inconsistently disguised data. Zhang et al. [44] introduce a two-way communication privacy-preserving scheme for CF in which users perturb their ratings for each item based on the server's guidance instead of using an item-invariant perturbation. Parameswaran and Blough [34] propose a framework for obfuscating sensitive information in such a way that it protects individual secrecy and also preserves the information content required for CF. Kaleli and Polat [19] propose a method for producing private referrals using Naïve Bayesian Classifier (NBC)-based CF. They propose to use RRT for preserving users' confidentiality. According to their empirical results, their method is able to produce predictions with decent accuracy.

where k_1, k_2, \dots, k_C show the number similar users held by the first, second, ..., and the C th party, respectively, who rated q . The MP, which is asked by a for recommendation, needs aggregate data from other $C - 1$ companies to estimate recommendations. If the MP sends a 's data to other parties, they can easily compute the required data. However, since a 's ratings are valuable and will be added to the MP's database; and her data will be used for prediction generation in the following queries, it does not send them to collaborating companies. Thus, such companies should compute $\sum_{u=1}^k z_{uj} v_{duq}$ and

$\sum_{u=1}^k z_{uj}$ aggregate values for all $j = 1, 2, \dots, m-1$, where m shows number of items; and send them to the MP without greatly jeopardizing their privacy. Therefore, the parties follow the following Private Distributed k -nn CF Protocol (PDKNN) to offer predictions:

1. The MP determines a 's cluster (c_a) using Eq. (1) after receiving required data from a . Note that in Eq. (1), x represents a 's ratings vector. The MP assigns a to the closest cluster. It sends c_a and q to other parties.
2. Each party including the MP computes $\sum_{u=1}^k z_{uj} v_{duq}$ and $\sum_{u=1}^k z_{uj}$ aggregate values based on those users' data that are in c_a . They then send them to the MP.
3. The MP then is able to estimate P using Eq. (8) after collecting required aggregate data from other parties.
4. It finally estimates p_{aq} and returns it to a .

The parties can succeed recommendation process based on HDD. However, PDKNN has the following shortcomings:

- a. Since the MP has the required partial results for the target cluster (c_a) to estimate p_{aq} , it can use them for producing referrals for those active users who will be in that cluster and ask prediction for q .
- b. The MP can collect aggregate data values for fake active users over a time in order to derive information about other parties' databases.

To overcome the aforementioned shortcomings, the parties follow the following steps to compute aggregate values in the step 2 of PDKNN protocol, where we call the new protocol as the IPDKNN (Improved PDKNN):

1. Each party j uniformly randomly selects a random number (β_j) over the range $(0, \gamma]$. They then uniformly randomly choose β_j percent of the users who did not rate q , where the probability of selecting any user is proportional to the number ratings she has due to accuracy concerns. In other words, the chance of selecting a user with more ratings is bigger than the chance of selecting a user with fewer ratings.

2. Each party then fills selected users' cells for q with non-personalized ratings (v_d). Since v_d values are estimated based on available ratings, when selecting users, giving higher probability to those users with more ratings makes sense. The parties generate v_d values using the distribution of users' ratings, which can be considered as a Gaussian distribution with mean (μ) and standard deviation (σ).
3. Before calculating $\sum_{u=1}^k z_{uj} v_{duq}$ and $\sum_{u=1}^k z_{uj}$ aggregate values for all $m-1$ items, each party uniformly randomly selects some of its z_{uj} values, removes their values, and replaces with zero. For this purpose, each data holder j uniformly randomly selects a random number (α_j) over the range $(0, \delta]$. They then uniformly randomly choose α_j percent of their z_{uj} values, remove their values, and replace with zero.
4. Each party then estimates the required aggregate values based on its modified database. They finally send them to the MP.

In order to enhance the understanding of our proposed scheme, we presented our method in Fig. 1. Notice that Fig. 1 shows the exchange data between the MP and the helping companies and the MP and a . Note that x represents a 's ratings vector. The figure also demonstrates the computations performed by each involving party including the MP and a .

When the parties follow the aforementioned protocols, they preserve their privacy against each other and such protocols force them to collaborate whenever a asks a prediction from one of them. They can produce accurate and dependable predictions. In one hand, the parties increase the amount of data involved in aggregate data computation by inserting v_d values in some q 's empty cells. On the other hand, the amount of data involved in such computations is reduced due to removed z_{uj} values. In each query, data owners choose different β and α so that unpredictable randomness is added to their databases. Each party will compute different partial results for a cluster in different recommendation processes. Therefore, they collaborate with each other to answer queries until they have enough data to offer accurate and dependable recommendations by themselves.

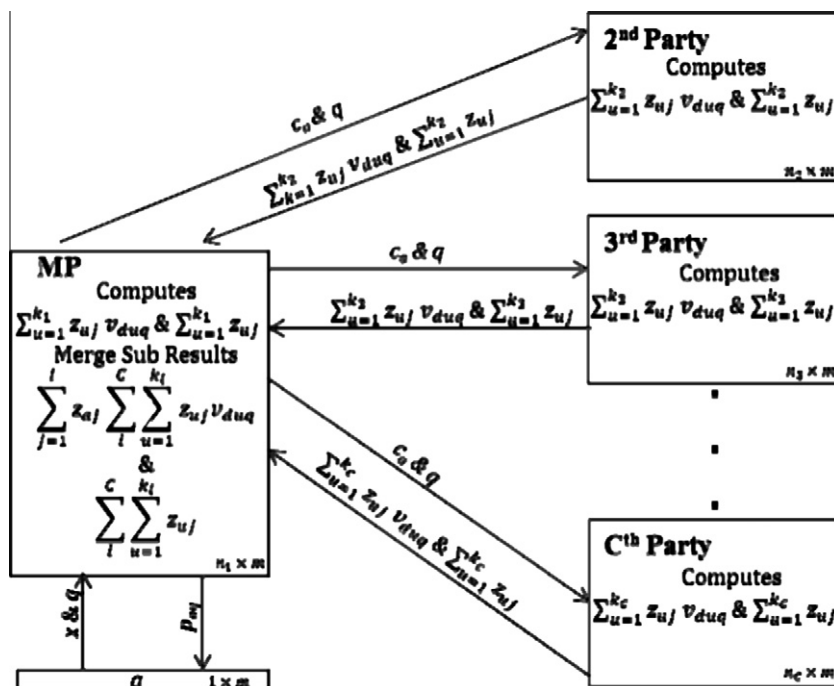


Fig. 1. An overview of private distributed k -nn collaborative filtering protocol.

To improve online performance, the parties compute normalized values off-line and store them. Due to their storage, extra storage cost is in the order of $O(nm)$. Since each party uses the cluster centers, they save them in C matrices with size $c \times m$. Accordingly, additional storage cost is in the order of $O(m)$ because C and c are constants. Although our scheme seems to cause further storage costs; however, the parties should save such information after calculating off-line to improve online performance even if they offer referrals by themselves.

Providing recommendations is an online process. Performing CF tasks efficiently is imperative. For this reason, our proposed scheme must not introduce significant extra computation costs that might harm the efficiency of CF schemes. Due to clustering, which is conducted off-line, additional costs are not critical. During online phase, data owner insert some default votes, which increases the amount of computations, while they remove some of the z-scores, which decreases the amount of computations. Generally speaking, applying these two different randomness processes surpass their effects on the amount of computations. Furthermore, data used in CF are distributed in our scheme and online computations are done simultaneously. Traditional k -nn-based scheme's online computation time (without the use of clustering) is in the order of $O(nm)$, while our proposed scheme improves online computation costs by $C \times c$ times without considering communication costs due to clustering and simultaneous computations.

Our scheme introduces extra online communication costs. In traditional CF schemes, an active user sends a message (her ratings vector and a query) to a server that returns a prediction. Hence, number of communications is two only. In our distributed scheme, a sends the same message to the MP as in traditional systems. However, the MP sends c_a and q to $C - 1$ companies so that they return partial results (two vectors containing $m - 1$ aggregate values). In other words, number of communications is $2C$ in our scheme or in the order of $O(C)$, where C is a constant. Therefore, communication costs increase by C times. Supplementary costs can be considered negligible because auxiliary parties simultaneously communicate with the MP.

7. Accuracy and overall performance analysis

To test our scheme in terms of accuracy and investigate its overall performance, we perform various experiments on real data sets. Accuracy shows how precise our privacy-preserving scheme-based recommendations are. We conduct trials for testing how the proposed scheme affects the quality of the predictions.

7.1. Data sets and evaluation criteria

We used two well-known real data sets, Jester and MovieLens (ML), constructed for CF purposes. Jester data set contains ratings for jokes [13]. ML data set was collected by GroupLens at the University of Minnesota (www.cs.umn.edu/research/GroupLens). We describe the data sets in Table 1, where we present various properties of both data sets.

To measure the quality of the referrals, we used Mean Absolute Error (MAE) because it is the most well-known statistical accuracy metric. The lower the MAE is, the more accurate our results are. Thus, MAE value should be minimized. MAE measures how close the predictions with privacy concerns to the true ratings. If p_1, p_2, \dots, p_d are actual user rating values, and p'_1, p'_2, \dots, p'_d are predicted values with privacy concerns, then $\{\xi_1, \xi_2, \dots, \xi_d\} = \{p'_1 - p_1, p'_2 - p_2, \dots, p'_d - p_d\}$ represents errors. Therefore, the MAE can be computed, as follows: $MAE = \frac{\sum_{i=1}^d |\xi_i|}{d}$, where d shows the total number of predictions. Since collaboration among multiple parties increases the amount of data involved in prediction process, the parties are able to generate predictions for more items

and they might overcome cold start problem. Thus, to show how collaboration affects the number of items for which predictions could be provided, we utilized coverage metric, which is the percentage of items for which a CF algorithm can provide referrals. Coverage can be calculated, as follows: $Coverage = v_{res}/v_{test}$, where v_{res} and v_{test} stand for the number of predictions returned and the number of test ratings. Finally, we applied statistical t -tests in order to show that our results are statistically significant and they are not occurred by chance. We first compute a t value. Then, we find a p -value from t -distribution table. If the p -value chosen for some significance level (usually 0.10, 0.05, or 0.01) is less than the calculated t value, then it is concluded that the improvements are statistically significant and they are not happened by chance.

7.2. Methodology

Given the entire data sets, we first determined those users who rated at least 50 items from both data sets. We then uniformly randomly divided such users into two disjoint sets, training and test sets. We finally randomly selected 1000 and 500 users for training and testing from train and test sets, respectively. For each test user, we uniformly randomly chose five rated items. After withholding their true ratings, we replaced their entries with null; and tried to provide referrals for them using the train users' data. We assumed that data are distributed among C companies, where C might be 1, 2, 3, 5, 7, or 10. Hence, each data owner owns about n/C number of train users. Since we use SOM clustering to determine k nearest neighbors, we clustered each train set using SOM clustering algorithm. We ran our trials using MATLAB 7.6.0 on a computer, which is Intel Core2Duo, 2.0 GHz with 2 GB RAM. To carry out SOM clustering, we used the toolbox in MATLAB. Since Roh et al. [38] determined the optimum cluster number as three, we set radius of lattice to $3/2$; and network topology to hexagonal lattice, which is default topology in the MATLAB toolbox.

7.3. Experiment results

Effects of collaboration on coverage and accuracy: We first performed trials to demonstrate how coverage changes with varying n and C values. In other words, we first tested how collaboration affects overall performance. With increasing available data, coverage is expected to increase. We hypothesize that the parties are able to provide predictions for more items if they integrate their split data through collaboration. To verify this hypothesis, we performed experiments while changing n from 250 to 1000 and C from 1 to 10. We assumed that if there is at least one rating for q ; and at least two commonly rated items between a and those users who rated q , the CF system can provide referrals for q . We found coverage values for data owners based on data sets they own only (split data) and combined data (collaboration) for both data sets. Since Jester is a dense data set (the density of the set that we use is about 72%), coverage is 100% even if n is 250 and C is 10. However, since ML is a very sparse set (the density is about 4%), coverage is significantly affected by varying available data. In Fig. 2, we presented the average coverage values for ML with varying n and C values.

As expected, coverage significantly improves with increasing n values for sparse data set ML. If n increases, amount of ratings involving in recommendation process also increases; that makes

Table 1
Data sets.

Name	Item	Size ($n \times m$)	Total votes	Density (%)	Range	Type
Jester	Joke	$24,983 \times 100$	1,810,455	72.47	$[-10, 10]$	Continuous
ML	Movie	6040×3900	1 million	4.22	$[1, 5]$	Discrete

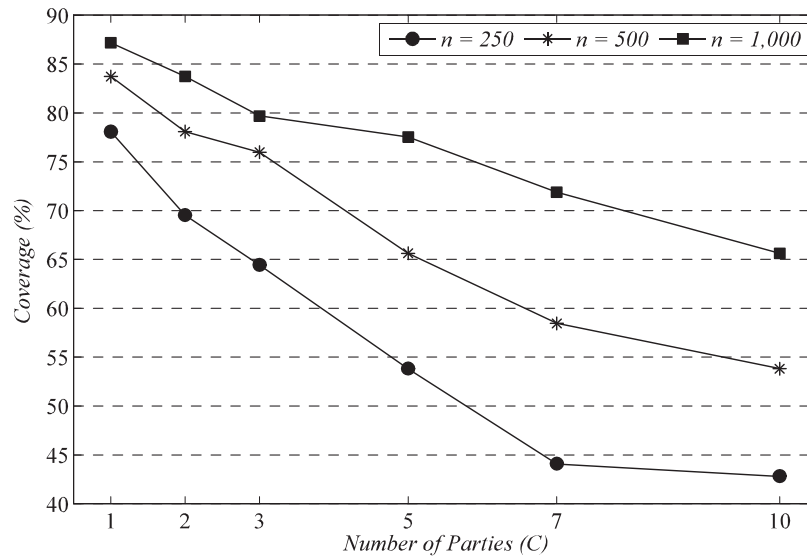


Fig. 2. Coverage with varying n and C values.

coverage better. As seen from Fig. 2, due to integrating split data, coverage enhances. When 250 users' data horizontally distributed among 10 parties, coverage is about 42%. If they integrate their data through collaboration, coverage increases to 78%. Note that when C is 1, data are held by a single party. In other words, when $C = 1$, it means that the parties decide to collaborate. For sparse data sets, collaboration among various parties definitely improves coverage.

To show how accuracy changes due to collaboration among multiple parties, we conducted experiments using both data sets. We wanted to compare results when collaborating and when the parties do not collaborate. We used 500 users for testing and used 250, 500, and 1000 users for training, where we varied C from 1 to 10. Notice again that when $C = 1$, it means that data owners collaborate and provide predictions on integrated data. If $C = 2, 3$, and so on, then it means that data are partitioned between two, three parties, and so on, respectively. With increasing C values from 2 to 10, the parties provide predictions on their split data only. Number of users held by each party decreases with increasing C values. We clustered train data using SOM clustering. The optimum value of

c was determined by Roh et al. [38], where they found three as the best one. Thus, we clustered train users into three clusters, which happen to give the best results. We first estimated predictions for test items for all test users using the data held by each party only. Then, predictions were estimated for the same items using the integrated data. We computed the MAE values for both cases; and displayed them in Fig. 3 for Jester and ML data sets.

As seen from Fig. 3, when data owners decide to collaborate, they achieve the best results (the outcomes for $C = 1$). The MAE values improve with decreasing C values. In other words, if data owners provide predictions on their integrated data via collaboration, they offer more accurate recommendations. Similarly, accuracy enhances with increasing n values, as expected. When data are distributed among various parties, each party uses its data to provide predictions. Since available data decrease, accuracy becomes worse. If they decide to collaborate, they are able to use more data for referral generation. That makes accuracy better. When n is 1000 and data are distributed among 10 parties, the MAE is about 0.83 for ML, while it is about 0.75 if they collaborate. Thus, integrating split data definitely enhances the quality of the

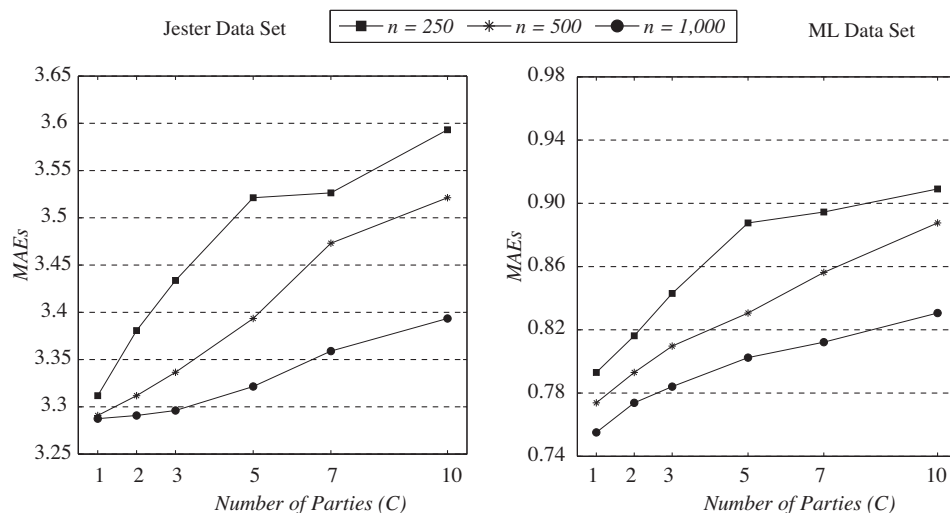


Fig. 3. MAEs with varying n and C values.

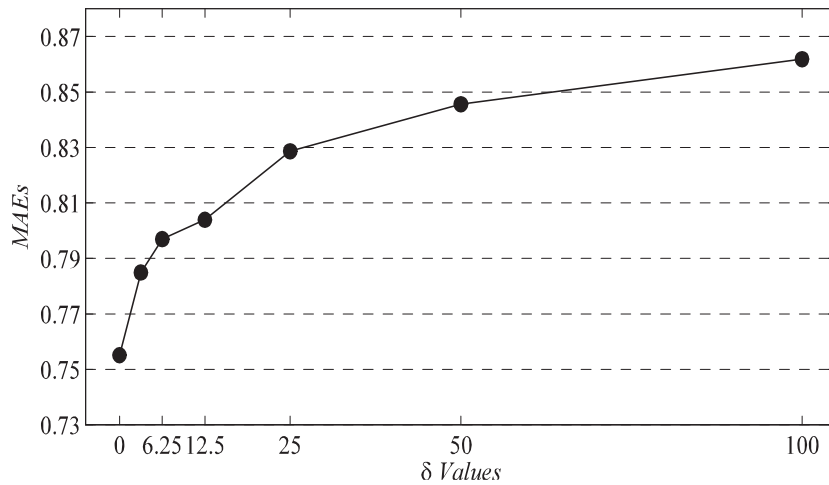


Fig. 4b. MAEs with varying δ values (ML data set).

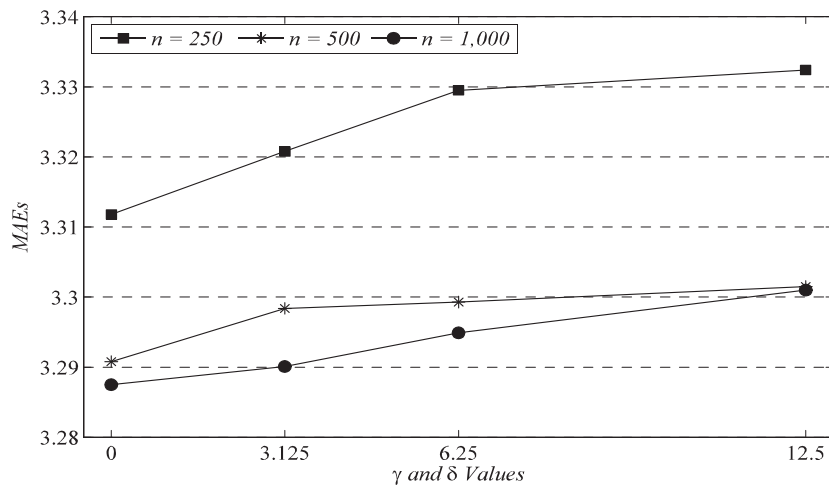


Fig. 5a. Joint effects of varying γ and δ values on MAEs (Jester data set).

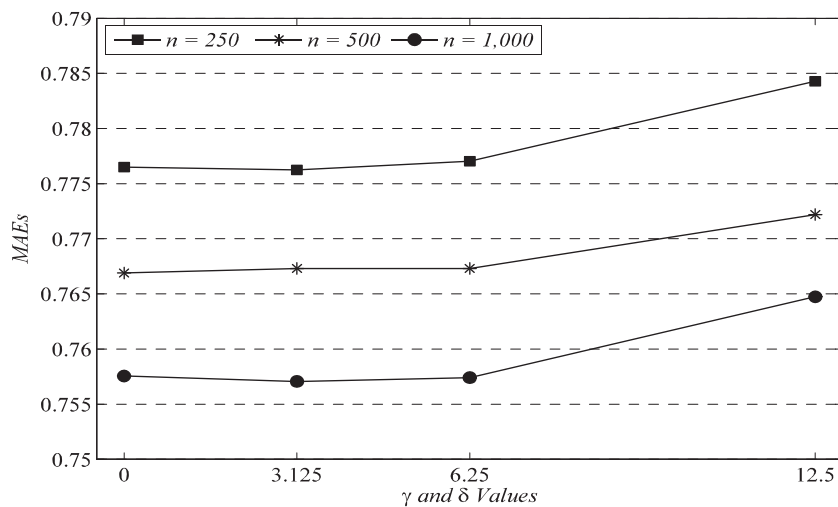


Fig. 5b. Joint effects of varying γ and δ values on MAEs (ML data set).

mendations with privacy concerns. As shown previously, due to collaboration among various parties even competing companies, improvements in accuracy are statistically significant. On the other

hand, privacy-preserving measures cause losses in accuracy. Such losses should not surpass the gains due to alliance. Compared to the enhancements, accuracy losses are smaller.

To show whether privacy-preserving distributed scheme develops accuracy significantly or not, we performed *t*-tests. We compared the results on split data with the ones on our proposed scheme. For ML data set, for example, the improvements are statistically significant for $n = 1000$, $\gamma = \delta = 3.125$, and $C = 10$ because the value of t is 5.63, which is still greater than the value of t for significance level = 0.01 in the *t*-table. In the same case, for Jester, the value of t is 7.62, which is still greater than the value of t in the *t*-table. Even if C is 5, the improvements are still statistically significant for both data sets. In our *t*-test experiments, we used 99 as the degree of freedom. As the *t*-tests show, our distributed data-based scheme with privacy improves accuracy. When data holders offer predictions on their split data only, accuracy diminishes due to the insufficient amount of ratings. However, if they collaborate, accuracy enhances even if they apply privacy-preserving measures because of increasing amount of available ratings.

7.4. Comparison of the proposed method with traditional clustering-based schemes

Our empirical outcomes emphasize that it is possible to produce accurate recommendations and increase producing accurate prediction capability of the parties having insufficient data by employing SOM-based CF algorithm while preserving data owners' privacy. Our results are comparable with the ones reported in the literature obtained by utilizing other recommendation algorithms. In order to compare our results with the ones conducted in the literature, we used the Normalized Mean Absolute Error (NMAE) metric. The NMAE can be computed, as follows: $NMAE = \frac{MAE}{v_{\max} - v_{\min}}$, where v_{\max} and v_{\min} represent the maximum and minimum votes, respectively. One of the well-known CF algorithms, referred to as Eigentaste, proposed by Goldberg et al. [12] can produce predictions with NMAE value of 0.187. The authors utilize recursive rectangular clustering algorithm in order to cluster users off-line. Herlocker et al. [15] propose a memory-based scheme whose accuracy in terms of NMAE is about 0.1920, where the authors utilize some techniques like normalization, significance weighting, and neighbor selection for such enhanced accuracy. Roh et al. [38] apply SOM clustering to CF, where they also utilize Case-Based Reasoning (CBR) for better performance. The authors' scheme, called SOM cluster-indexing CBR CF predictor, yields NMAE of 0.1524. They also propose SOM cluster induction and SOM cluster neural network CF predictor as comparative models, which achieve NMAE values of 0.1892 and 0.1557, respectively. Furthermore, Roh et al. [38] utilize another comparative model referred to as simple Pearson CF predictor, which yields NMAE of 0.1719. Xue et al. [41] suggest grouping users using *k*-means clustering for improved performance. Their empirical results show that their clustering-based scheme achieves NMAE of 0.2055. Honda and Ichihashi [16] propose a scheme for CF utilizing linear fuzzy clustering. Their robust fuzzy clustering-based method produces predictions with NMAE of 0.1877. In another study performed by Taek-Hun et al. [40], the authors propose to use *k*-means clustering for grouping data for improved performance. According to their empirical results, the highest accuracy, in terms of NMAE, that their scheme achieved is about 0.1892. Without privacy concerns, Roh et al. [38] compare performance of SOM-based CF scheme with comparable models, as explained above. We suggest a privacy-preserving scheme to provide predictions on distributed data without violating data owners' privacy, where users' data are clustered utilizing SOM clustering only. When we consider the confidentiality of data owners, our proposed method yields the best results in terms of NMAE values of 0.1885 and 0.1645 for ML and Jester, respectively. For ML, compared to the results presented in Ref. [38], our results show that accuracy decreases due to privacy, as we expected. However, our results are still promising.

8. Conclusions and future work

We presented a privacy-preserving scheme to provide recommendations based on horizontally distributed data among multiple parties using clustering-based collaborative filtering algorithm. Accuracy, performance, and privacy are major goals that recommender systems want to accomplish. Since they are conflicting goals, we provided a scheme finding equilibrium among them. To improve online performance, clustering is widely used. We also applied clustering in our proposed scheme. Data collected for recommendation purposes might be partitioned among multiple companies, even competing sites. Performing prediction services on integrated data is vital to offer accurate predictions. Since data are split, the parties might not offer accurate referrals. As our experiment results show, integrating split data significantly improves preciseness. Although privacy concerns make accuracy worse, accuracy losses are smaller than the accuracy gains due to collaboration. We showed that enhancements in accuracy due to our proposed scheme are statistically significant. Auxiliary costs due to privacy are also negligible. Our scheme still makes it possible to offer referrals efficiently.

We will investigate whether we can apply other clustering methods or not while providing distributed data-based recommendations with confidentiality. We considered horizontally distributed data in this study. In addition to horizontal partitioning, data can be arbitrarily partitioned. Note that we assumed that each party's database includes exactly the same items. However, arbitrary partitioning is more common in real life. We are planning to show how to extend our scheme to arbitrarily distributed data. We are also planning to investigate how to provide trust-based recommendations on distributed data with privacy.

Acknowledgment

This work was supported by the Grant 108E221 from TUBITAK.

References

- [1] H.J. Ahn, A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem, *Information Sciences* 178 (2008) 7–51.
- [2] S. Berkovsky, Y. Eytan, T. Kuflik, F. Ricci, Privacy-enhanced collaborative filtering, in: *Workshop on Privacy-Enhanced Personalization in Conjunction with the International Conference on User Modeling, UK, 2005*, pp. 75–83.
- [3] M. Berry, G. Linoff, *Mastering Data Mining*, John Wiley & Sons, New York, 2000.
- [4] S.S. Bhowmick, L. Gruenwald, M. Iwaihara, S. Chatvichienchai, PRIVATE-IYE: a framework for privacy-preserving data integration, in: *Proceedings of the 22nd International Conference on Data Engineering Workshops*, Atlanta, USA, 2006, pp. 91–99.
- [5] J. Bobadilla, F. Ortega, A. Hernando, J. Alcalá, Improving collaborative filtering recommender system results and performance using genetic algorithms, *Knowledge-Based Systems* 24 (2011) 1310–1316.
- [6] J. Bobadilla, F. Ortega, A. Hernando, J. Bernal, A collaborative filtering approach to mitigate the new user cold start problem, *Knowledge-Based Systems* 26 (2012) 225–238.
- [7] J. Bobadilla, F. Serradilla, J. Bernal, A new collaborative filtering metric that improves the behavior of recommender systems, *Knowledge-Based Systems* 23 (2010) 520–528.
- [8] J. Canny, Collaborative filtering with privacy, in: *Proceedings of IEEE Symposium on Security and Privacy*, CA, USA, 2002, pp. 45–57.
- [9] J. Canny, Collaborative filtering with privacy via factor analysis, in: *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, 2002, pp. 238–245.
- [10] C. Clifton, M. Kantarcioglu, A. Doan, G. Schadow, J. Vaidya, A. Elmagarmid, D. Suciu, Privacy-preserving data integration and sharing, in: *Proceedings of the 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, Paris, France, 2004, pp. 19–26.
- [11] G. Gan, C. Ma, J. Wu, *Data Clustering: Theory, Algorithms, and Applications*, SIAM, 2007.
- [12] K. Goldberg, T. Roeder, D. Gupta, C. Perkins, Eigentaste: a constant time collaborative filtering algorithm, *Information Retrieval* 4 (2001) 133–151.
- [13] D. Gupta, M. Digiovanni, H. Narita, K. Goldberg, Jester 2.0: evaluation of a new linear time collaborative filtering algorithm, in: *Proceedings of the 2nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, CA, USA, 1999, pp. 291–292.

- [14] S. Haykin, *Neural Networks: A Comprehensive Foundation*, second ed., Prentice-Hall, Upper Saddle River, 1999.
- [15] J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, *ACM Transactions on Information Systems* (TOIS) 22 (2004) 5–53.
- [16] K. Honda, H. Ichihashi, Component-wise robust linear fuzzy clustering for collaborative filtering, *International Journal of Approximate Reasoning* 37 (2004) 127–144.
- [17] B. Jeong, J. Lee, H. Cho, Improving memory-based collaborative filtering via similarity updating and prediction modulation, *Information Sciences* 180 (2010) 602–612.
- [18] C. Kaleli, H. Polat, Providing naive Bayesian classifier-based private recommendations on partitioned data, in: J.N. Kok, J. Koronacki, R.L. DeMantaras, S. Matwin, D. Mladenic, A. Skowron (Eds.), *Knowledge Discovery in Databases: PKDD 2007*, pp. 515–522.
- [19] C. Kaleli, H. Polat, Providing private recommendations using naive Bayesian classifier, in: K.M. WegrzynWolska, P.S. Szczepaniak (Eds.), *Advances in Intelligent Web Mastering*, 2007, pp. 168–173.
- [20] C. Kaleli, H. Polat, SOM-based recommendations with privacy on multi-party vertically distributed data, *Journal of the Operational Research Society* (2011), <http://dx.doi.org/10.1057/jors.2011.76>.
- [21] M. Kantarcioglu, J.S. Vaidya, Privacy-preserving naive Bayes classifier for horizontally partitioned data, *IEEE ICDM Workshop on PPDm*, Melbourne, FL, USA, 2003, pp. 3–9.
- [22] M. Kantarcioglu, C. Clifton, Privacy-preserving distributed mining of association rules on horizontally partitioned data, *IEEE Transactions on Knowledge and Data Engineering* 16 (2004) 1026–1037.
- [23] S. Kaya, T. Pedersen, E. Savaş, Y. Saygıın, Efficient privacy-preserving distributed clustering based on secret sharing, in: T. Washio, Z.-H. Zhou, J. Huang, X. Hu, J. Li, C. Xie, J. He, D. Zou, K.-C. Li, M. Freire (Eds.), *Emerging Technologies in Knowledge Discovery and Data Mining*, 2007, pp. 280–291.
- [24] J. Kelleher, D. Bridge, An accurate and scalable collaborative recommender, *Artificial Intelligence Review* 21 (2004) 193–213.
- [25] T. Kohonen, *Self-Organizing Map*, Springer, Berlin, Heidelberg, New York, 1995.
- [26] L. Kun, H. Kargupta, J. Ryan, Random projection-based multiplicative data perturbation for privacy preserving distributed data mining, *IEEE Transactions on Knowledge and Data Engineering* 18 (2006) 92–106.
- [27] N. Lathia, S. Hailes, L. Capra, Private distributed collaborative filtering using estimated concordance measures, in: *Proceedings of the 2007 ACM Conference on Recommender Systems*, Minneapolis, MN, USA, 2007, pp. 1–8.
- [28] X. Lin, C. Clifton, M. Zhu, Privacy-preserving clustering with distributed EM mixture modeling, *Knowledge and Information Systems* 8 (2005) 68–81.
- [29] Y. Lindell, B. Pinkas, Secure multiparty computation for privacy-preserving data mining, *Journal of Privacy and Confidentiality* 1 (2009) 59–98.
- [30] X. Luo, Y. Xia, Q. Zhu, Incremental collaborative filtering recommender based on regularized matrix factorization, *Knowledge-Based Systems* 27 (2012) 271–280.
- [31] J. Mao, A.K. Jain, A self-organizing network for hyperellipsoidal clustering (HEC), *IEEE Transactions on Neural Networks* 7 (1996) 16–29.
- [32] OECD, *Guidelines for Consumer Protection in the Context of Electronic Commerce*, 2000.
- [33] OECD, *Guidelines on the Protection of Privacy and Transborder Flows of Personal Data*, 2005.
- [34] R. Parameswaran, D.M. Blough, Privacy-preserving collaborative filtering using data obfuscation, in: *Proceedings of IEEE International Conference on Granular Computing*, 2007, pp. 380.
- [35] H. Polat, W. Du, Achieving private recommendations using randomized response techniques, in: W.-K. Ng, M. Kitsuregawa, J. Li, K. Chang (Eds.), *Advances in Knowledge Discovery and Data Mining*, 2006, pp. 637–646.
- [36] H. Polat, W. Du, Effects of inconsistently masked data using RPT on CF with privacy, in: *Proceedings of the 2007 ACM Symposium on Applied Computing*, Seoul, Korea, 2007, pp. 649–653.
- [37] H. Polat, W. Du, Privacy-preserving top-N recommendation on distributed data, *Journal of the American Society for Information Science and Technology* 59 (2008) 1093–1108.
- [38] T.H. Roh, K.J. Oh, I. Han, The collaborative filtering recommendation based on SOM cluster-indexing CBR, *Expert Systems with Applications* 25 (2003) 413–423.
- [39] B. Sarwar, G. Karypis, J. Konstan, J.T. Riedl, Application of dimensionality reduction in recommender systems – a case study, in: *Proceedings of ACM WebKDD Workshop*, 2000, pp. 264–272.
- [40] K. Taek-Hun, P. Seok-In, Y. Sung-Bong, Improving prediction quality in collaborative filtering based on clustering, in: *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Sydney, Australia, 2008, pp. 704–710.
- [41] G.-R. Xue, C. Lin, Q. Yang, W. Xi, H.-J. Zeng, Y. Yu, Z. Chen, Scalable collaborative filtering using cluster-based smoothing, in: *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Salvador, Brazil, 2005, pp. 114–121.
- [42] Y. Yang, W. Tan, T. Li, D. Ruan, Consensus clustering based on constrained self-organizing map and improved Cop-Kmeans ensemble in intelligent decision support systems, *Knowledge-Based Systems* (2011). <http://dx.doi.org/10.1016/j.kbs.2011.03.031>.
- [43] W. Yuan, D. Guan, Y.-K. Lee, S. Lee, S.J. Hur, Improved trust-aware recommender system using small-worldness of trust networks, *Knowledge-Based Systems* 23 (2010) 232–238.
- [44] S. Zhang, J. Ford, F. Makedon, A privacy-preserving collaborative filtering scheme with two-way communication, in: *Proceedings of the 7th ACM Conference on Electronic Commerce*, Ann Arbor, MI, USA, 2006, pp. 316–323.