Privacy-Preserving Ranking over Vertically Partitioned Data

Keivan Kianmehr
Department of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, Canada
kkianmeh@uwo.ca

Negar Koochakzadeh
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada
nkoochak@ucalgary.ca

ABSTRACT
Privacy concerns in many application domains prevent sharing of data, which limits data mining technology to identify patterns and trends from large amounts of data. Traditional data mining algorithms have been developed within a centralized model. However, distributed knowledge discovery has been proposed by many researchers as a solution to privacy preserving data mining techniques. By vertically partitioned data, each site contains some attributes of the entities in the environment. Once an existing data mining technique is executed at each site independently, the local results need to be combined to produce the globally valid result. Learning how to rank existing entities is a central part in many knowledge discovery problems. In this paper, we present a method for ranking problem based on SVMRank algorithm in situations where different sites contain different attributes for a common set of entities. Each site learns the ranking model of entities without knowing the attributes in other sites and at the end the global rank model will be built.

Categories and Subject Descriptors
H.2.8 [Database Management]: Database Applications – Data mining; H.2.4 [Database Management]: Systems – Distributed database; H.2.7 [Database Management]: Database Administration – Security, integrity, and protection.

General Terms

Keywords
Privacy, Data Mining, Ranking, Support Vector Machines.

1. INTRODUCTION
Data mining technology is a means of knowledge discovery to efficiently analyze large quantities of data to find interesting patterns and trends. Mining contains various algorithms such as association rule mining, clustering, classification and ranking. Classical data mining algorithms implicitly assume complete access to all data against which the algorithms are being run.

Privacy and security concerns may restrict access to data due to either legal or commercial reasons and there may not be a central site with authority to see all the data [1]. Therefore, privacy considerations often limit data mining projects. Although, data mining solutions generally reveal high-level knowledge in terms of global models rather than detail information about the individual entities and thus rarely violate privacy, the privacy violation through the process of mining can pose real privacy issues [2]. The reason is that gathering data and bringing them together to support data mining makes misuse easier. In other words, the problem is not data mining results, but the process that generates them. If the results were generated without sharing information, and the results could not be used to deduce private information, data mining would not reduce privacy [2]. Although obtaining globally meaningful results without sharing information seems impossible, some solutions have been proposed for that.

Growing concern about the privacy implications of data mining has recently increased the research interest in proposing privacy-preserving data mining algorithms [1–6]. In this regard, distributed knowledge discovery with the purpose of not disclosing data beyond its original source is receiving growing attention. In this methodology, there are several sites each of which holds some attributes of the entities, and the sites wish to collaborate to build a globally valid model. To preserve privacy, no site should reveal individual data or learn anything new from the process of data mining, and thus the opportunity for misuse is not increased by this process [3]. In other words, anything learned during the data mining process must be derivable from one’s own data and the final result.

Data distribution among various sites can be achieved through horizontally or vertically partitioned data. We assume vertically partitioned data in which each party collects different information about the same set of entities. For instance, bank and insurance company might gather different information about the same set of people. In vertically partitioned data, a portion of each instance is presented at each site, but no site has complete information about any entity [4].

In this paper we propose a privacy-preserving technique for ranking problem. Ranking is an important data mining task applicable in many diverse domains. For instance, in financial sector, ranking customers based on the credit risk or ranking fraudulent transactions are both considered as ranking problems. The goal of a ranking algorithm is to build a model that can predict the ranked list of entities based on their attributes’ values. There are various classical data mining techniques for learning how to rank a set of entities. A Support Vector Machines-based algorithm called SVMRank was proposed in [7] for ranking documents retrieved by a search engine in response to a query.
The proposed privacy-preserving ranking technique in this paper is based on applying SVMRank algorithm to vertically partitioned data. The method presented here works for any number of parties where each party knows a specific set of attributes while all of them are aware of the target ranking of the entities. Each party learns how to rank the entities based on its local attributes. Merging the ranking results without sharing the information might not be feasible; however in the proposed technique we transfer the data to a centric site in a manner that we are able to rank entities while their privacy is preserved.

In the rest of this paper, in section 2 we review some related works on privacy-preserving data mining techniques. To the best of our knowledge there is no research conducted on distributed ranking method for vertically partitioned data. In section 3 we explain the background on support vector machines (SVM) ranking followed by our proposed method in Section 4. In section 5, we empirically show the applicability of our method to a crime data set and rank states in US based on their crime criticality. Lastly the conclusion and future works are discussed in Section 6.

2. RELATED WORKS
Data mining and knowledge discovery investigate the automatic extraction of unknown patterns and trends from large amounts of data. Privacy preserving data mining is a novel research area in data mining where data mining algorithms are analyzed for their side-effects on data privacy. Several approaches have been adopted for this purpose such as data modification, data encryption and data distribution [8].

Data modification in general is to modify the original values of a database that need to be released to the public, and thus to ensure privacy protection. There are different methods for data modification such as perturbation (altering an attribute value by a new value), blocking (hiding an existing attribute value), aggregation (combining several values into a coarser category), swapping (interchanging values of individual records) and sampling (releasing data for only a sample of population) [8].

Cryptography-based techniques are Secure Multi-party Computation (SMC) where after computation nothing is learned that could not be deduced from one’s own data and results [4]. In particular, SMC problem deals with computing a probabilistic function on any input, in a distributed network, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to a participant in the computation than its input and output [9].

In data distribution the solution to preserve privacy is to avoid disclosing data beyond its source, while constructing data mining models equivalent to those that would have been learned on an integrated data set [3]. Distributed data can be classified as horizontal distribution and vertical data distribution. In horizontal distribution different database records reside in different places, while in vertical data distribution all values of disjoint subsets of attributes reside in different places.

Recently there have been several researches on preserving privacy for specific data mining techniques such as clustering, classification and association rule mining [1-6, 10]. While some of the proposed techniques make trade-offs between efficiency and information disclosure, all maintain privacy of individual information.

In [1] a privacy preserving technique for association rule mining was proposed in which each site holds some attributes of each transaction, and all sites wish to collaborate to identify globally valid association rules. Privacy-preserving clustering has been addressed by many researchers. Works described in [2, 10] are two examples of privacy preserving clustering based on k-means algorithm. Privacy-preserving data mining based on decision trees was introduced in [11, 12]. Privacy-preserving algorithm for other types of classification techniques such as Naïve Bayes, and SVM are proposed respectively in [3] and [6].

To the best of our knowledge there is no research conducted on privacy-preserving approaches to the ranking problem. Learning how to rank entities in any given domain is an important factor in the decision making process of many applications. Further, privacy concern exists on revealing information through the ranking process. Therefore, the need for a ranking algorithm that guarantees privacy has motivated us to propose a privacy preserving ranking technique based on data distribution. We utilize vertically partitioned data to distribute training set instances into several parties. For the ranking purpose, each party transfers the data to a centric party in such a way that the privacy is maintained. The proposed algorithm is based on SVMRank algorithm proposed in [7].

3. BACKGROUND ON SVM RANKING
Ranking is the central part of many decision making problems in that a decision has to be made after existing entities in an environment are ranked. In domains with multi-attribute entities it is challenging to find a single attribute as the representative of all attributes and use that attribute to sort the entities. By knowing the target rankings for different sample sets of entities, machine learning techniques can be used to learn a model that can rank entities in a new sample set.

Ranking algorithm based on Support Vector Machines optimization is a technique which uses values of all available features of entities in different sample sets to learn a ranking model. The goal of the SVM-based ranking algorithm is to maximize the Kendall’s tau coefficient which is a statistic metric to measure the association between two ranked lists. This ranking model is then utilized to rank entities in a new sample set.

Given independently and identically distributed training samples S of size n containing entities e (as a feature vector of m attributes) with their target ranking r*, the learner will build a ranking model to minimize the ranking error.

In linear SVM ([7]), this is equivalent to finding the weight vector so that the maximum number of the following inequalities is fulfilled.

\[ \forall (e_i, e_j) \in r_i^*: \tilde{w} e_i > \tilde{w} e_j \]

... \[ \forall (e_i, e_j) \in r_n^*: \tilde{w} e_i > \tilde{w} e_j \]

Where: \( (e_i, e_j) \in r_k^* \) if \( e_i \) has been ranked higher than \( e_j \) based on the target rank \( r_k^* \).

This optimization problem is solved in [7] based on Support Vector Machine optimization process. After finding the weight vector, it is used for ranking a new sample of n new entities. Figure 1 illustrates an example of four points as example entities.
to be ranked based on two different weighted vectors. For any weight vector \( \mathbf{W} \), the entities are ordered by their projection onto \( \mathbf{W} \). This means that for \( \mathbf{W}_1 \) the points are ordered \((1,2,3,4)\), while \( \mathbf{W}_2 \) implies the ordering \((2,3,1,4)\).

![Figure 1. Example of 2-Dimensional Ranking based on SVMRank.](image)

The projection of a point onto the weighted vector \( \mathbf{W} \) is equivalent to the signed distance to a hyperplane with normal \( \mathbf{W} \). Figure 2 shows the scalar projection of the point \( A \) onto \( \mathbf{W} \), which is equal to \( \mathbf{W} \cdot \mathbf{A} \). In summary, the rank model built by SVMRank, in terms of a weight vector \( \mathbf{W} \), is used to rank elements in a new sample based on the scalar multiplication values of the vectors corresponding to elements and vector \( \mathbf{W} \).

![Figure 2. Scalar Projection of A onto W.](image)

### 4. PROPOSED PRIVACY-PRESERVING RANKING

Privacy concerns can prevent sharing all attributes about the entities under consideration from data provider to a data collector. However, in many cases, data collectors can be provided with a subset of the attributes. To guarantee a privacy preserving ranking, a distributed ranking technique is first applied in each site and results are then merged to produce the final ranking result. Figure 3 illustrates the proposed distributed ranking technique based on SVMRank that works for a vertically partitioned dataset. This technique preserves the privacy by transferring to the central site the scalar projection of entities onto the weight vector instead of making use of the original feature values.

Suppose \( k \) privacy preserved and disjoint subsets of features \( f_{s_1}, f_{s_2}, \ldots, f_{s_k} \), such that each site is provided with one of these sets. The training set in site \( i \) contains different sample sets of \( n \) entities each of which includes the values of features in \( f_{s_i} \) as well as the target ranking \( r^* \). A new sample set, namely testing set, that includes values of all entities for the same set of features in \( f_{s_i} \) is provided to each site \( i \) as well.

In each site the learning component applies support vector machines optimization to find the local weighted vector with minimum ranking error. This vector is then used to find the scalar projection of each entity in the testing sample. The scalar projection values are then sent by each site to the coordinator to merge the results and predict the ranking of the testing sample.

Merging process is conducted by adding all the scalar projections from \( k \) sites together. This summation is then used to order the entities in the testing set and predict the final ranking result. The rationale behind this merging step is explained next.

Suppose that we have all \( m \) features available in the learning process of ranking and accordingly the weight vector \( \mathbf{W} \) is learned as the rank model. Now let \( F_{s_i} \) with \( 1 \leq i \leq k \) be an \( mxm \) diagonal matrix that represents features that are available at site \( i \). In other words, a diagonal entry of row \( r \) in \( F_{s_i} \) is 1 if \( r^i \text{th} \) feature exists in \( f_{s_i} \) (set of features distributed to site \( i \)) and 0 otherwise. Let \( I_m \) be a unit matrix of size \( m \), \( m \) is the number of features; and \( f_{s_1} \cup f_{s_2} \cup \ldots \cup f_{s_k} = f_s \) (the set of all features) and \( f_{s_1} \cap f_{s_2} \cap \ldots \cap f_{s_k} = \emptyset \); \( \mathbf{W} \) can be expanded based on equation 2-4.

\[
\mathbf{W} = \mathbf{W}_1 \times I_m \tag{2}
\]

\[
I_m = F_{s_1} + F_{s_2} + \ldots + F_{s_k} \tag{3}
\]

\[
\mathbf{W} = \mathbf{W}_1 \times F_{s_1} + \mathbf{W}_2 \times F_{s_2} + \ldots + \mathbf{W}_k \times F_{s_k} \tag{4}
\]

However, we have assumed that not all \( m \) features are available in a single site to find \( \mathbf{W}_1 \), and each site only has access to a subset of features. The weight vector \( \mathbf{W}_i \) calculated locally by site \( i \) is a \( m \)-dimension vector including values for only available features in this site and 0 value for the rest of features. We argue that \( \mathbf{W}_1 \) is a reasonable estimation of \( \mathbf{W} \times F_{s_i} \) and thus \( \mathbf{W} \) can be estimated by equation 5.

\[
\mathbf{W} = \mathbf{W}_1 + \mathbf{W}_2 + \ldots + \mathbf{W}_k \tag{5}
\]

The rationale behind this argument is that the target of local
optimization process in each site is toward the global ranking optimization target. Local optimization process in site \(i\) deals with finding the weight vector \(\vec{w}_i\) so that the maximum number of the inequalities (1), where \(\vec{W}\) is replaced by \(\vec{W}_1\), is fulfilled. For all those inequalities that are locally fulfilled in each site, we can add elements from two sides of inequalities for the same pair of entities in the same sample. As an example, suppose that \((a, b) \in r^*\) and the global optimization process by knowing all the features find \(\vec{w}\) which fulfills \(\vec{w}_a > \vec{w}_b\). However, the local optimization process in each site finds the corresponding weight vectors, all \(\vec{w}_i\), which fulfill \(\vec{w}_a > \vec{w}_b\) except the one with \(i = k\).

In this case, \(\sum_{i=1}^{n} \vec{w}_a > \sum_{i=1}^{n} \vec{w}_b\) which is almost the same as \(\vec{w}_a > \vec{w}_b\). By plugging equation (5) into inequalities (1) we obtain inequalities (6).

\[
\forall (e_i, e_j) \in r^* : \vec{W}_i e_i + \ldots + \vec{W}_k e_i > \vec{W}_i e_j + \ldots + \vec{W}_k e_j (6)
\]

5. CASE STUDY

To empirically evaluate the proposed privacy-preserving ranking technique, we define a case study that provides an example of using the proposed technique to rank communities in US based on their crime level. In this study, a set of available attributes about different communities in each state of US is used to build a model for crime ranking. The ranking of communities in each state can be predicted based on the learned model from other states. In this study we use a publicly available data set on communities and crime from UCI Machine Learning Repository [13].

5.1 Crime Dataset

This dataset contains socio-economic attributes as well as crime data about communities in US [13]. Socio-economic attributes are from Law Enforcement Management and Admin Stats survey and the crime features are from FBI UCR.

The dataset contains 125 attributes called predictors and 18 potential goals about number of different types of crime in each community from 1995 for 48 states in US. The complete list of 125 predictors with their descriptions is available in [13]. As an example, NumUnderPov is the number of people under the poverty level, PctImmigReg5 is percentage of immigrants who immigrated within last 5 years, or PctBornSameState is percent of people born in the same state as currently living.

The extra 18 crime attributes describe the number of 9 different crimes and the ratio of these numbers per population in each state. The 9 different types of crime are murder, rape, robbery, assault, burglary, larceny, auto theft, arson and violent crime.

To be able to apply SVMRank to rank the communities based on their crime level, some preprocessing steps need to be performed on this dataset. For instance, as numbers of communities in various states are different, they cannot be used as training or testing instance (in SVMRank number of items in all training instances as well as testing instances should be equal). Therefore, we first selected a subset of 27 states which all had more than 20 communities. Then from each state, we only used 20 random communities to build our train and test sets. The 27 selected states are AR, AZ, CA, CO, CT, FL, GA, IN, KY, MA, MO, NC, NH, NJ, NY, OH, OK, OR, PA, RI, SC, TN, TX, UT, VA, WA, and WI.

Among 125 predictor features for the above 27 states, some of them has missing values. Missing values will negatively influence the result of any data mining task. There are different approaches that replace missing values with calculated or estimated values. However dealing with missing values was not our concern in this work. We simply excluded features with missing values from our dataset. This simple feature reduction step resulted in eliminating 23 features. Therefore, our dataset includes 102 features none of which contains any null value. Further, The SVMRank implementation that we have used for our experiment requires all feature values to be in range \([0, 1]\). Therefore as the next preprocessing step we normalize feature values so that they are mapped to range \([0, 1]\). For this purpose, we simply use min-max normalization technique. Min-max normalization transforms a value \(A\) to \(B\) which fits in the range \([\text{min}, \text{max}]\). The mapping formula is as follows:

\[
B = \frac{A - \text{min}}{\text{max} - \text{min}} (7)
\]

Once the predictor features are prepared for our case study, the target rank score of each community in each state should be set. This rank score is then used by SVMRank to build the ranking model. We use the Index Crime metric to rank the communities. This metric is defined as the total number of crime per population. To measure this index we calculate the summation of 9 crime ratios in our dataset. This index is then used to sort communities. The higher the index crime of a community is, the higher the crime rank score of that community will be in the sorted list.

5.2 Experimental Result

The final dataset that is eventually obtained after performing all the aforementioned preprocessing steps contains 102 features as well as the target rank score for 20 comminutes of each 27 states in US. This dataset is then used in our experiment. We select one state at a time as the testing instance and the rest of states (26 others) as our training instances. Each training instance includes the crime data of 20 communities located in that particular state and their crime rank scores. In other words, SVMRank learns a ranking model from the crime data and known crime rank score of 26 states and predicts the crime ranking of communities in a new state (test instance), given the crime data of the communities in the test instance. We conduct this experiment 10 times repeatedly to ensure the reliability of the results. In each iteration, a different state is used as a test instance so that in total 10 different states are selected as testing instances. These states are CA, CT, IN, MA, NY, PA, SC, TX, UT, and WA.

For the aim of comparison, for each one of these 10 experimental points we run original SVMRank technique and our proposed privacy-preserving SVMRank algorithm to build prediction models. To simulate vertically partitioned data, we randomly distribute 102 predictor features in two different sites. These two sites then apply SVMRank algorithm locally to get the scalar projections of communities in the testing state on their corresponding weighted vector \(\vec{W}\). The projection values are then sent to a centric site in that the results are merged and the final ranking is predicted.

The rankings resulted from the two models, the original SVMRank and the privacy-preserving SVMRank, are compared in terms of ranking accuracy. The accuracy is calculated based on the correlation of the predicted rank scores of communities in the
testing set and the actual target rank score of those communities. Figure 4 demonstrates the accuracy of 10 experimental points where two different models were applied.

![Figure 4. Crime Ranking Prediction Accuracy.](image)

As the result shows, ranking accuracy of the proposed technique in different states follows the same trend of accuracy result of using all available features (with some deviation for each cases). Interestingly, for some states such as IN, TX, and UT the accuracies are exactly the same. This result confirms that the proposed technique can be applied for ranking of entities presented in form of vertically partitioned data to preserve the privacy while maintaining the prediction accuracy.

However, there are many threats to validity of our experimental study in this work which should be more thoroughly investigated in future. For instance, distributing features to several independent sites can be performed based on the semantic of the features instead of randomly distribution them into two sites.

6. CONCLUSION AND FUTURE WORK
Privacy concerns limit data mining tasks in many applications. As a solution several privacy-preserving data mining techniques have been proposed recently for different mining tasks such as association rule mining, clustering and classification. Learning how to rank the entities is another interesting type of data mining techniques that can play important role in many decision-making problems while privacy concerns still exist in the ranking process. The proposed technique in this paper presents a privacy-preserving algorithm for ranking the entities, based on vertically partitioned data distribution. This algorithm is based on locally executing SVMRank algorithm on independent sites and transferring the local ranking results to a central site where they are merged while the privacy is preserved.

In this work we empirically validated the practically of the proposed approach. More evaluation is needed to be performed in future on data from different application domains while the semantic aspects of privacy for distributing features will be taken into consideration as well.

7. REFERENCES