CYBERNETICA
Institute of Information Security

Privacy preserving collaborative filtering with Sharemind

Dan Bogdanov, Richard Sassoon

T-4-2 / 2008

Dan Bogdanov[1,2], Richard Sassoon[2].
[1] AS Cybernetica, Institute of Information Security
[2] University of Tartu, Institute of Computer Science

The research reported here was supported by:

1. Estonian Science foundation, grant(s) No. 6944,
2. the target funded theme SF0012708s06 "Theoretical and Practical Security of Heterogenous Information Systems",
3. the European Regional Development Fund through the Estonian Center of Excellence in Computer Science, EXCS,
4. EU FP6-IST project AEOLUS (contract no. IST-15964).

Cybernetica research reports are available online at
`http://research.cyber.ee/`

Mailing address:
AS Cybernetica
Akadeemia tee 21
12618 Tallinn
Estonia

Privacy preserving collaborative filtering with Sharemind

Dan Bogdanov, Richard Sassoon

January 25, 2010

Abstract

The statistical analysis of data warehouses, i.e., repositories for an organization's data, is used for finding patterns or trends that are useful in decision making. This process is often called *data mining*. One way to use such data is through *collaborative filtering*, in which suggestions are given to a user based on previous transactions of other users. The question of privacy in this context is important, since one doesn't want to disclose his private transaction in order to perform this analysis. This is so even, if disclosing this information gives the user to some service.

This paper will present an alternative solution to cryptographic and statistical randomization algorithms for privacy preserving association-rule mining for *collaborative filtering* applications. This is achieved by using a multi-purpose framework called Sharemind[1], that uses share-computation in order to securely evaluate functions.

1 Introduction

We know that information controls decisions and actions and having the most accurate information can help in making better decisions. Every day, a large number of people are entering data into online databases by registering themselves at websites (like social networks), buying in an online shop, updating their personal information in accounts, going to the doctor, etc. This information can be used to identify users and this raises the issue of information privacy. Nowadays one just needs to use any search engine to find some information about a person.

The organizations, who have the data, want to extract relevant associations from it despite the privacy-related usage restrictions. They want to be able to analyze trends in sales, client behavior or possibly diseases to provide a better service. Private information must not leak during this processing. It is hard tp prevent leaks, if the party processing the data has full access to it[3].

Current techniques for privacy preserving data mining have some flaws. The randomization approach works by perturbing single entries but preserving the data's global statistical properties. These transformations only preserve privacy on average and perturbation reduces the output's precision[1]. Besides that, there are no security guarantees for individual records, which may lead to privacy breaches, i.e., the discovery of some property of an original record[2].

A second solution is to consider cryptographic methods such as multi-party computation, where all the parties want to compute some function of their inputs without leaking any information. This computation is costly if the number of participants grows beyond twenty-thirty.

An alternative approach has been proposed, where only a small number of parties performs the computations. They are also responsible for guaranteeing the privacy of the input data. In the Sharemind virtual machine[4] three data miner nodes perform the computation by applying share computing techniques.

In Section 2 of this work we outline privacy guarantees for Sharemind while in Section 3 we will present an example application to illustrate some privacy preserving concepts. Section 4 discusses the implementation of he respective algorithms in Sharemind and Section 5 presents the conclusions for this paper.

1.1 The collaborative filtering technique

The use of statistical and knowledge discovery techniques is widely used in e-commerce environments in order to generate good quality suggestions to an interacting customer. Applications that make use of such a scheme are said to have a recommender system. There are different ways of generating recommendations to a customer[5], classified as:

- Content-based recommendations. The recommended items will be similar to those the customer had showed interest in before.
- Collaborative recommendations. Identifies customers with similar taste to the interacting one and recommends items they have liked. A user has liked some item either by buying it or giving it a good rating.

- A hybrid approach. Combines the above mentioned methods.

In this report we are mostly concerned with the collaborative filtering method on an item-based configuration such as discussed in[6] and no item rating. In such a system recommendations are given based on itemsets that are likely to appear together, i.e., items that according to previous transactions have some similarity. For example, given that one likes A and B, they may also like C. In order to find this similarity between items, is proposed the use of association rule mining.

1.2 Association rule mining

In order to find association rules that are significant to the application, it is necessary first find the frequent itemsets and then find rules based on these sets. Similarly to [7] and [8], we use the following definitions:

- Attributes $I{=}i_1, \ldots, i_n$ are called the *items*. A set X containing k distinct items from I is called an *itemset*.
- A *database of transactions in matrix form* T contains transactions represented by zeroes and ones, indicating if item i_k of set I was of interest (one) or not (zero). Table 1 shows an example of this representation.
- The *support* $supp(X)$ of an itemset X is defined as the number of transactions t that support the itemset, i.e., numbers of transactions X such that $X \subseteq T$. An itemset is considered frequent if its support is greater than some support threshold σ.
- An *association rule* is of the form $X \Rightarrow Y$ such that $X, Y \subseteq I$ and $X \cap Y = \emptyset$. The itemset X is called the antecedent and the itemset Y is the consequent. An association rule is considered frequent if its support is greater than some support threshold σ'.
- The *support* $supp(X, Y)$ of an association rule $X \Rightarrow Y$ is the support of $X \cup Y$, and the frequency is the frequency of $X \cup Y$.
- The confidence of an association rule $X \Rightarrow Y$ is the conditional probability of having Y in a transaction, given that X is also present in that transaction: $confidence(X \Rightarrow Y) = supp(X \Rightarrow Y)/supp(X)$. A rule is then considered confident if its confidence is higher than some confidence threshold γ. An example, 90% of transactions that have *bread* and *butter* also have *yogurt*

	i_1	i_2	i_3	...	i_n
t_1	1	0	1	...	0
t_2	1	1	0	...	0
t_3	0	1	0	...	1
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
t_n	1	0	0	...	1

Table 1: A matrix representation of a transaction database

($\{bread, butter\} \Rightarrow \{yogurt\}$). The antecedent of the rule is bread and butter and the consequent is yogurt, while 90% is the confidence.

The association rules of interest will be the ones that satisfy both support (statistical significance) and confidence (rule's strength) thresholds. Most algorithms for association rule mining follow two phases. The first one is responsible for generating all rules of the type $X \Rightarrow \{\}$, which is the same as finding all frequent itemsets. The second phase generates all frequent and confident association rules. In other terms these rules say that if a person showed interest for a set X of items, it could be useful to suggest to this person a set Y of other items, with a good confidence.

2 The Sharemind virtual machine

Sharemind is a distributed virtual computer that makes use of share computing techniques in order to securely perform multi-party computations over a set of data, thus achieving privacy preserving data processing[1]. Data is distributed by using secret sharing schemes, where each participant in the multi-party environment is given its own share of a secret value. Sharemind uses an additive secret sharing scheme over the ring $\mathbb{Z}_{2^{32}}$ which is a notifiable difference from the standard choice that is Shamir's scheme[9]. While This change allows Sharemind to securely process standard 32-bit integers.

Sharemind is information-theoretically secure in an honest-but-curious model[10] with three miners, i.e., if every participant follows the protocol and at most one node is semi-honestly corrupted, it is not possible for an adversary to find out the confidential information. It is also assumed that the parties will not collude with each other, that is, will not make an agreement to deceive the other and thus find the

secret. Intuitively, Sharemind requires three organizations or individuals who will host a miner node. None of the miner hosts can see input values in its own miner database. Also, during computations the input values are never in the memories or storage devices of any miner node in reconstructable form.

The main goal of the framework is to provide efficient protocols for basic mathematical operations in order to be able to implement more complex tasks. For this all protocols are designed to be universally composable, meaning that they can run one after the other or in parallel, without losing security guarantees.

The framework, in a more general case, is designed as follows: Each miner node P_i has a local database for persistent storage and a local stack for storing intermediate results. The miner P_i has in its database a share s_i of a secret value s such that:

$$s_1 + s_2 + \ldots + s_n = s \, mod \, 2^{32}$$

and any $n-1$ element subset $\{s_{i_1}, \ldots, s_{i_{n-1}}\}$ is uniformly distributed. The input values are distributed automatically in controller nodes that provide input data and commands for the miners. In order to perform computations on the shares the miners communicate between themselves via private channels.

The current implementation of Sharemind provides privacy preserving addition, multiplication and comparison of two shared values. This set of operations is sufficient for a wide range of applications since several algorithms for data mining and statistical analysis do not require more than that.

3 The example application

For the analysis of a Sharemind solution to privacy preserving association rules mining we propose the following application. A person visiting a museum wants to retrieve some information about a specific work of art, e.g. a painting (French, Italian, Dutch, ...), drawing, sculpture, antique, etc. The whole visit is considered as a transaction in the context of this application, since it contains the information of all the items of interest to this person. The granularity can also be adjusted if necessary, to consider, for example, types of antiques (Greek, Roman, Egyptian) or the age of origin.

Assume that the person has a portable device for making such a query, through a mobile or wireless network, that is in turn sent to some service which is able to provide the requested information. The device can also store the set of items of interest to the user, in order to make up this particular transaction that will be saved at the end of the session.

Using algorithms that generate association rules for the transaction database, it is possible to find relevant suggestions for the user, for example, "if the visitor liked drawings and Italian paintings, he might also like Roman antiques". This way the user receives both the desired information and an additional recommendation for some other work of art that may be of interest.

We consider that the portable device that the user can use to access this service could be his own mobile phone or a special purpose PDA handed out at the entrance of the museum, possibly for a small fee. The PDA would have a content provider built-in, with information of all works of art available and would be responsible for showing the requested information and for sending the query to the transaction database server that would then return the suggestions based on the association rules cache. The privacy risks of such a system are low, as the PDA is not easily tracked to an individual person. If the system additionally uses Sharemind, the privacy of the system is near-perfect, as the source transactions will not be available to anyone.

In case the user decides to use his mobile, it may be necessary to download an interface to access the service, but it may not be possible to download all the information from an external content provider due to storage restrictions. In this situation it should be possible to get the information from an external content provider without disclosing which information it is, in order to keep the user's privacy. The choices made by the visitor are considered private information. Since the mobile phone of the visitor could easily be associated with the person, the risk is higher than when using the PDA.

In a Sharemind-based system the content provider could store the information in secret-shared form in the miners that would send it to the user's mobile on demand., in which the service interface would mount the information based on the shares. If the content provider sent the data directly to the user, the mobile and requested information could be identified in this process.

The privacy issue in a museum environment doesn't appear to be so important, but it is vital when the transactions contain sensitive data. This is just a simple example and can derive other applications that are more concerned about privacy, like an application that suggests news and thus could determine your political views.

4 Proposed implementation with Sharemind

Let us consider a solution for the presented application in the case where a user uses his own mobile device through a special interface as a way to access the desired functionalities. The user should be able to request and receive information without

worrying that this data could be traced back to him, for whatever purpose one may imagine.

We propose the following solution:

1. Four tables are stored in a secret-shared database:

   - contents $C = \{code(index), content\}$;
   - lookup $L = \{code, category\}$;
   - associations $A = \{X \Rightarrow Y\}$;
   - transactions
     $T = \{id, paintings, sculptures, \ldots\}$;

   Where any $x \in X$, $y \in Y$, and $paintings$, $sculptures, \ldots \subseteq \{category\}$, and T is as defined in section 1.2.

2. Split a traditional content database into C, where the contents are stored in secret-shared form.

3. User downloads service interface, if it's the first time he or she is using it, and can start making requests. The interface is built with the controller library of Sharemind.

4. When the user makes a request for information, this request is split into three shares and sent to the miners. What is sent is the work of art's unique code. We consider that a work of art is associated to just one item content.

5. The miners initialize a comparison protocol on these shares to find the requested information in the shared contents database. At the same time another comparison is instantiated on L in order to find the kind of work of art the user is interested in. With this information it's possible to start a final comparison on the association rules database with the objective of finding a set of rules that can provide some good suggestions to the user. Parallelization is an important performance aspect for Sharemind and will help the performance of these operations.

6. When the set of rules has been determined, it is not necessary that all the consequents, as defined in section 1.2, are suggested. It is possible to select, at random, k of them, find some random representatives in the contents database and use their title and location attributes as the suggestion and keep the whole content in a shared cache in case the user accepts the suggestion, thus saving a new search in the database.

7. At some point the user interface receives the shares of the content requested and the suggestions, joins and process them, showing them in a human-readable format to the user. It also stores the kind of work of art that was of interest, in order to compute the user's transaction and also to send it in some next request, so another set of rules can be chosen.

8. When the user finalizes the interface, his final transaction is uploaded to T.

See Figure 1 for a simplified architectural view of the solution.

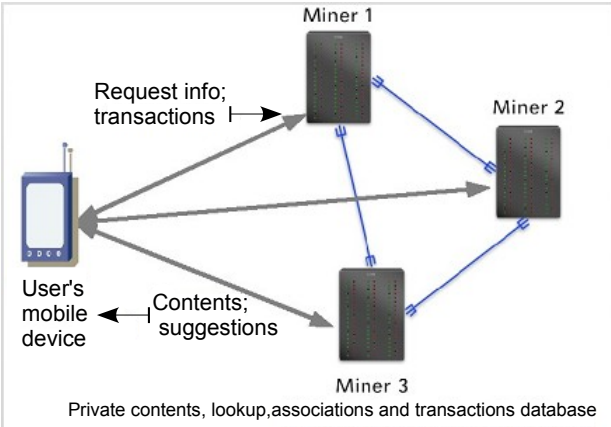

Figure 1: Solution architecture

One important aspect to be considered is the latency of this approach since, as of this writing, the comparison protocol is relatively slow. After the actual implementation of this scheme and tests with real data it will be possible to verify if the time constraints achieved are reasonable in real-world applications. In any case, for the final step we can consider that the suggestions are returned to the user after the requested content has been returned, i.e., we don't need to return both at the same time.

Furthermore, the computation of the association rules is expensive and should be done periodically while caching the results. The result should be stored in A in order to generate faster responses to user requests. The caching does not increase the privacy risks, as the recommendations usually do not contain identifiable private information.

We could also consider a more granular approach to this application, if instead of categories of work of art we consider the individual piece, e.g., *'La Gioconda'*, *'Medusa's Raft'*, *'The Thinker'*, ... In that case the transactions table would need

a large amount of columns, one for each piece of art in the museum, and that can't be achieved, not to say that hardly a user would ask for information on so many pieces. A way to solve this is to have each transaction contain only the items that were of interest to some user and the association rule mining would generate rules in the format $\{$*'La Gioconda', 'Medusa's Raft'*$\} \Rightarrow \{$*'The Thinker'*$\}$, which could be more interesting to the user. The steps mentioned above would still be valid in this case, its only the transaction that would have a different format, e.g., $\{id_1, id_2, id_3 \ldots\}$,where id_i is the unique code associated to each piece of art that the user requested information about.

The above mentioned approach can be implemented by using any sparse matrix representation, that is, a matrix that stores only non zero entries, as described in [11] and some algorithm for association rule mining that works on this kind of matrices. See [12] for a package designed for this purpose. We suggest that this kind of representation and its corresponding manipulation for mining association rules are implemented in Sharemind in order to be able to deal with more realistic amounts of data.

5 Conclusions

We have seen presented an example application applying the Sharemind framework for privacy-preserving collaborative filtering. The proposed solution is applicable to different application domains requiring strong privacy preservation guarantees. The guarantees are the standard ones provided by the Sharemind virtual machine—given three organizations that do not collude with each other, none of them is capable of reconstructing the private inputs or intermediate values during data processing.

Without further benchmarking it is not possible to evaluate the feasibility of the proposed solution in practical scenarios. The possibility of running the required operations in parallel will improve the computation time, but improvements to the Sharemind virtual machine might be required to achieve near real-time responses in the user's interface.

References

[1] Bogdanov, D., Laur, S., Willemson, J.. Sharemind: a framework for fast privacy-preserving computations. In Proceedings of 13th European Symposium on Research in Computer Security, ESORICS 2008, LNCS, vol. 5283, pp. 192-206. Springer, Heidelberg (2008).

[2] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke. Privacy preserving mining of association rules. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery in Databases and Data Mining, pp. 217–228, Edmonton, Alberta, Canada, July 23–26 2002.

[3] Pinkas, B. Cryptographic techniques for privacy preserving data mining. SIGKDD Explorations 2002.

[4] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In STOC [DBL88], pp. 1–10.

[5] Adomavicius, G. & Tuzhilin, A. (June 2005), "Toward the Next Generation of Recommender Systems: A report of the State-of-the-Art and Possible Extensions", IEEE Transactions on Knowledge and Data Engineering 17(6): pp. 734–749.

[6] G. Linden, B. Smith, and J. York, "Amazon.com Recommendations Item to item collaborative filtering", IEEE Internet Computing, Vo. 7, No. 1, pp. 76–80, Jan. 2003.

[7] R. Agrawal; T. Imielinski; A. Swami: "Mining Association Rules Between Sets of Items in Large Databases", SIGMOD Conference 1993: pp. 207–216.

[8] Goethals, B.: Report on Frequent Pattern Mining. Department of Computer Science. University of Helsinki

[9] Shamir, Adi (1979). "How to share a secret". Communications of the ACM 22 (11): pp. 612–613

[10] O. Goldreich. Foundations of Cryptography, volume 2. Cambridge University Press, May 2004.

[11] Randolph E. Bank, Craig C. Douglas. Sparse Matrix Multiplication Package (SMMP), April 2001.

[12] Hahsler, M., Grün, B., Hornik, K.: arules - a computational environment for mining association rules and frequent item sets. Journal of Statistical Software 14(15) (2005).