



Global model of distributed data mining is not summing of local models

M. Soliman & M. Kantardzic
*Computer Science and Computer Engineering,
University of Louisville, KY, USA*

Abstract

Discovering meaningful information from data is what is known data mining. By meaningful information we mean sequential patterns, rules and other indexes embedded in the data. So far, attention in the data mining process has always focused on extracting information from data physically located at one central site. However, most real life applications rely on data distributed in several locations. As a result, a feasible model for mining distributed data is required. The paper presents a model for mining distributed sequential patterns. The model is further verified by a simulator that can work with different mining tasks.

1 Introduction

Data mining is the science of discovering rules, patterns, and various indexes embedded in large set of data [1]. This paper concentrates on the problem of discovering sequential patterns between items over transaction databases. A typical record in transaction database consists of items bought by a customer along with information such as customer IDs of buyer, date of transaction, quantity, price, etc. The classical discovery algorithms for such database usually assume that data is centralized [1-6]. However, there are many applications in which centralizing data is infeasible due to security of sites, autonomy of sites, communication overhead, etc. In this case distributing data is a necessity [9, 10]. As a result the location of the data will have an impact on the discovery process. The new structure will dictate some changes on the discovery process procedures, logic, and methods. The discovery process will not only take place on individual level of each data site, but will be extended to a new global level

90 Data Mining IV

where sites combine their results into one set of results. Few algorithms considered distribution of data [7-12]. However they assumed the existence of global mining model, which could be obtained if data were centralized or combined. They also ignored the fact that some data sites may be homogeneous (contain data representing similar features) or heterogeneous (contain data representing different features)[12]. In this paper, we verify through simulator, a previous proposed approach [7, 8] for obtaining global model of distributed datasets. The global model if exists, is not a simple summing of local models at the sites. The simulator builds global model only if preceded by adequate proofs of both data homogeneity and results homogeneity. Section 2 explains briefly the distributed mining approach adopted in this work. Section 3 lists the different components of the simulator and then details each component separately. Section 4 concludes our work and gives directions for future work.

2 Distributed data mining approach

Distributed environment used to be viewed as a solution for intensive computations with large datasets. However, operating with real world applications in distributed environment motivated new reasons to appear different than the traditional one. One application is in distributed data mining. Data mining algorithms are known with their combinatorial results which make distributed environment a good candidate for them. But sometimes, distribution of data is intended because the distribution itself has a meaning Badia and Kantardzic [9]. A very popular example is in banking where banks like to exchange their models for fraud detection or loans approval. This is infeasible because of privacy of customers. Even if exchanging models were feasible it will lead to unexpressive results in case of loans approval because banks apply different policies for loans Wirth *et al* [10]. In these cases, the following approach for mining distributed data is suggested:

- Build local models for each site
- Discuss relations between local models of sites i.e., which sites are similar and which are different, to what extent they differ and the implication and significance of their differences.
- For similar sites build global model(s) and exclude those very different

3 Components of simulator

The above approach is a general approach that has to be experimented with different data mining tasks. To experiment a specific mining task, different algorithms has to be experimented to select the best-fit algorithm for distributed data mining purposes. Therefore the simulator is playing an essential role in accomplishing our goal and results. Four main components are the basics of the simulator.

1. Generating Synthetic Data
2. Local models of sites

3. Global Model Analysis
4. Global model(s) of sites

3.1 Generating synthetic data

The core of the mining process is data. Several parameters control the generated data. We used most of the parameters mentioned in Agrawal and Srikant [1] in generating the synthetic data. These parameters are listed in Table 1. The number of transactions / site is basically the size of the database at that site.

Table 1: Parameters of dataset.

D	Number of transactions/site
L	Average sequence length
MxSL	Maximum items code
MnSL	Minimum items code
Dist	Distribution of transactions
Seed	Starting data stream for site
Support	Minimum Support set by the user/site

It should be noted here that the generated data is sequences database. Sequence database is transactions after being sorted by customer ID and transaction Date as mentioned in Agrawal and Srikant [1]. The Minimum items code and maximum item code are two parameters representing the range of data generated. For example items in transactions could be coded from 10 to 100 for one site while another site could use a code from 35 to 150. These parameters are used to allow some overlapped items between sites if user desired so. The seed parameter gives new start for the random number generator used at each site. By this seed different numbers will be generated and therefore simulating realistic data to great extent. The last parameter provided is the minimum number of occurrences for items in database to be called frequent items.

3.2 Local models of sites

This second component of the simulator builds local models of sites. By local models we mean discovered frequent sequences resulting from applying particular mining algorithm. Therefore this component is contingent upon selecting certain mining algorithm for certain mining technique. Currently the

technique of discovering frequent sequences using Apriori by Agrawal and Srikant [1] algorithm is the only implemented mining algorithm in the simulator.

3.3 Global model analysis

This is the third main component in our simulator. This component analyses sites local models and based on this analysis global model building may or may not take place. This component contains two main analysis tools: **Similarity matrix**, and **Clustering Sites**.

3.3.1 Similarity matrix tool

The main analysis part is based on similarity matrix found in Li *et al* [12]. This matrix reflects the similarity between local models in the distributed environment. The similarity matrix is a square matrix with dimension equals to number of sites. Diagonal cells of the matrix have value 1. This value is the similarity from a site to itself. Other cells in the matrix represent the similarity from site on left column to site in top row of the matrix as in fig 1. There are many measures for calculating similarities between complex data structures found in Charikar [13]. Currently, we accepted the following measure which calculates the similarity (A, B) as a function that maps local models of two sites A, B to a number [0,1]. $\text{Sim}(A, B)=1$ corresponds to sequences that are identical while $\text{sim}(A, B)=0$ corresponds to sequences that are dissimilar totally, as stated in Charikar [13].

$$\text{Similarity}(A,B) = (A \cap B) / (A \cup B)$$

where $(A \cap B)$ is the number of common sequences between sets A, B. On the other hand $(A \cup B)$ is the union number of sequences between sets A, B.

		To	
		S ₁	S ₂
From	S ₁	1	Sim _{1,2}
	S ₂	Sim _{2,1}	1

Figure 1: Similarity matrix.

This measure was used by Border *et al* [14], and it is the same as Jaccard coefficient [15] of similarity used in information retrieval. This measure can be used to calculate similarities between data in sites and results. Applying this measure in data is straightforward. For example Table 2, shows three sites A, B, and C where the similarities between their data are calculated as follows.

$(A \cap B) = \{3,4,5,6,7,8,9\} = 7$; $(A \cap C) = \{3,4,5,6,7,8,9,11\} = 8$;
 $(B \cap C) = \{3,4,5,6,7,8,9,10\} = 8$;

$(A \cup B) = \{3, 4, 5, 6, 7, 8, 9, 10, 13, 16\} = 10$; $(A \cup C) = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 15, 16\} = 12$;
 $(B \cup C) = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15\} = 12$;
 $\text{Sim}(A, B) = (A \cap B) / (A \cup B) = 7/10 = 70\%$; $\text{sim}(A, C) = (A \cap C) / (A \cup C) = 8/12 = 66\%$;
 $\text{Sim}(B, C) = (B \cap C) / (B \cup C) = 8/12 = 66\%$;

Table 2: Datasets at sites A, B and corresponding similarities.

A Data	B Data	C Data
7/8 8/4/3,4,11/8,4/6 7/6/4,6,5/8/16 4,6/3/3,5/7,3 3/3/9,8	5/4 13,5,3/4/9,3,3/5/6/6/7 5/10,8/10/6,6/5 3,7/8/7,4/5,5 10,3/5,3,7,4,6,3/4/3,10	6,11,15,4/4 2,6,6/10,8/6,5,6/8,5 6,7/6,4/4,3,4 7/7/4/9,8,7 6/8/6/3/5,5/8/5,4

Table 3 shows the local models of sites A, B, and C. The highlighted sequences are the common sequences between sites.

$(A \cap B) = \{3, 4, 5, 6, 7, 8, 4/3, 3/3\} = 8$;
 $(A \cap C) = \{3, 4, 5, 6, 7, 8, 6/5\} = 7$;
 $(B \cap C) = \{3, 4, 5, 6, 7, 8, 7/4, 5/5, 8/5\} = 9$;
 $(A \cup B) = \{3, 4, 5, 6, 7, 8, 10, 7/8, 4/8, 4/3, (4, 6), 3/8, 3/3, 6/5, 5/5, 5/4, 5/3, (5, 3), 5/6, 5/10, 4/5, 3/5, 3/4, 3/6, (3, 7), 3/7, (7, 4), 7/4, 10/5, 10/6, 10/10, 8/5\} = 32$;
 $(A \cup C) = \{3, 4, 5, 6, 7, 8, 7/8, 4/8, 4/3, (4, 6), 3/8, 3/3, 6/5, 6/6, 6/4, 6/8, 6/3, 4/4, 8/6, 8/5, 8/8, 5/8, 5/5, 7/4\} = 24$;
 $(B \cup C) = \{3, 4, 5, 6, 7, 8, 10, 5/5, 5/4, 5/3, (5, 3), 5/6, 5/10, 4/5, 4/3, 3/5, 3/4, 3/3, 3/6, (3, 7), 3/7, (7, 4), 7/4, 10/5, 10/6, 10/10, 8/5, 6/6, 6/4, 6/8, 6/5, 6/3, 4/4, 8/6, 8/8, 5/8\} = 36$.

$\text{Sim}(A, B) = (A \cap B) / (A \cup B) = 8/32 = 25\%$;

$\text{Sim}(A, C) = (A \cap C) / (A \cup C) = 7/24 = 29\%$;

$\text{Sim}(B, C) = (B \cap C) / (B \cup C) = 9/36 = 25\%$;

Table 4 shows the similarities in data and local models between sites A, B, and C. The table also shows that similarity in data differs than similarity in local models. Generally, in our approach we are measuring similarities in both data and local models but for the clustering tool in the next paragraph we use similarity of local models.

3.3.2 Clustering tool

Clustering is defined in Fasulo [16] as organizing objects into groups whose members are similar in some way. Therefore this tool will implement a clustering algorithm to organize sites into groups whose local models are similar under some definition of similarity. There are two styles of clustering: hierarchical and partitioning or k clustering.

Table 3: Local models of A, B, C.

A	B	C
7	5	6
8	4	4
4	3	8
3	6	5
6	7	7
5	10	3
7/8	8	6/6
4/8	5/5	6/4
4/3	5/4	6/8
4,6	5,3	6/5
3/8	5/3	6/3
3/3	5/6	4/4
6/5	5/10	8/6
	4/5	8/8
	4/3	8/5
	3/5	5/8
	3/4	5/5
	3/3	7/4
	3/6	
	3,7	
	3/7	
	7,4	
	7/4	
	10/5	
	10/6	
	10/10	
	8/5	

Table 4: Calculated similarities in data and local models between sites of Table 2.

Sim A, B	Sim A, C	Sim B, C
7/10=70%	8/12=66%	8/12=66%
8/32= 25%	7/24=29%	9/36=25%

In the hierarchical clustering group of size more than one is in turn decomposed to smaller groups. In partitioning or k clustering each group is assigned to exactly one cluster. Numerous algorithms follow either styles of clustering [16,17]. The one to be implemented by this tool will be PAM (Partitioning Around Medoids) like clustering algorithm. Every clustering algorithm operates on a set of input vectors. The input for the simulator will be the similarity matrix for local models of sites. This similarity matrix will be in form of graph data structures where the nodes represent distributed sites, and the arcs are distances (similarities) of local models at sites. For example the sites with local models given in Table 2 may be represented as the graph in fig. 2. The output from this tool is expected to be something as in fig. 3. In this figure ten sites are assumed to participate in distributed environment. According to the similarity matrix values for those sites, the sites are clustered as follow: sites

1,2,4,5,7,8 in one group or cluster while sites 3,6,9 in another cluster. Site 10 is considered an outlier.

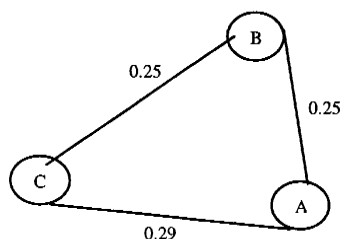


Figure 2: Clustering component expected output.

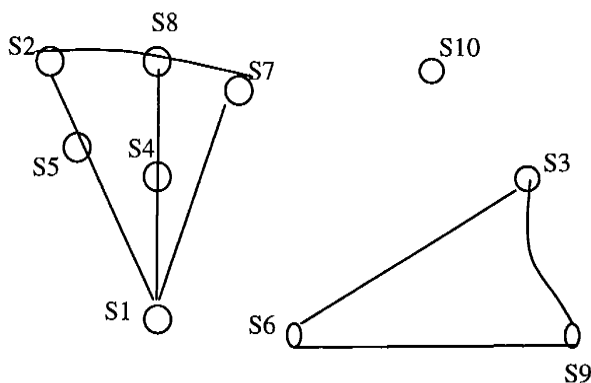


Figure 3: Expected output of the clustering component.

3.4 Global model building component

This component establishes the global model(s) for sites lying in one group or cluster. As illustrated in [7,8,9], the global model involves supports exchange between sites for candidate sequences. Only the supports of locally large sequences at sites have to be exchanged. This helps knowing which locally large sequences will be globally large. Accordingly the global model building is accomplished on two phases.

1. Logical analysis of local models and generation of candidates for global model.
2. Verification of global candidates against global threshold in the database.

For example, fig.4 represents three sites distributed environment S1, S2, S3 belonging to same cluster. The minimum support constraint is 25%. S1 local

large items are the set: {A, B, C, ABC}, S2 local large items are the set: {C, B, D, CB, BD}, and S3 local large items are the set: {A, C, G, AC}. To build the global model we categorize large sequences at each site into: sequences locally large at each site and therefore are globally large without further calculations such as {C} in this example. Sequences locally large at some sites, which will need further calculations to prove if they are globally large or not.

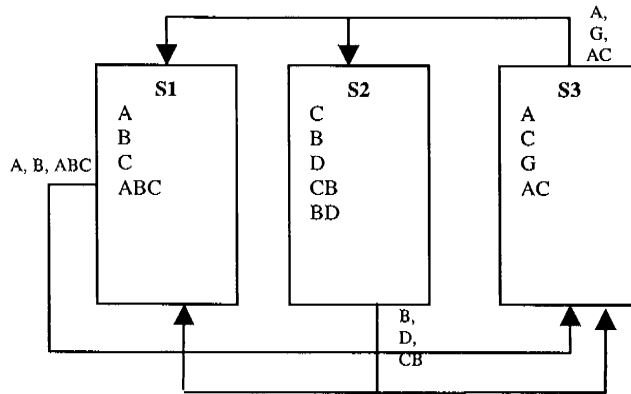


Figure 4: Global model establishment.

Then sites communicate as follow: S1 requests support for its locally large sequences {A, B, ABC} from sites S2, S3. Also S2, and S3 request supports of their large sequences {B, D, CB, BD}, {A, G, AC} respectively. For 1-length frequent sequence such as A, B, D, G calculations are straightforward. Only sequences above global minimum supports are included in the global sequences set. However, for sequences like ABC, some considerations have to be added in our calculations. We have to bear in mind that the whole sequence ABC is not found in other sites, but subsequences of ABC are found at S2 {B, C} and S3 {A, C, AC}. Therefore tentative calculations for these types of sequences have to be made. The work is still going on this part and results will be shown in coming works.

4 Conclusion

This paper proposes a distributed data-mining simulator to verify a proposed approach in mining frequent sequences when datasets are distributed. This approach is based on the following assumptions:

1. Distribution of datasets does not necessarily imply building global results.
2. Global model building does not mean aggregating local models of sites but rather it is a fusion of local models.

3. The fusing process into one global model should only take place when it is strongly justified. We define the justifications of building global model when there are adequate positive signs signifying that the resulting global model is strongly representing local models.

The approach has three phases. First, building local models for sites using the predefined mining algorithm takes place. Next global model analysis is conducted, which involves both comparing local models using similarity matrix and clustering of datasets. At last building global model for datasets lying in one cluster and excluding outliers. Next phase of research includes experimenting real-world data.

References

- [1] R. Agrawal and R. Srikant. Mining sequential patterns. *Proc of Int. Conf. On Data Engineering (ICDE'95)* Taipei, Taiwan, pp. 3-14, 1995.
- [2] R. Srikant and R. Agrawal. Mining sequential patterns: Generalizations and performance improvements. *Proc. of the 5th Intl. Conf. On Extending Database Technology*, March 1996.
- [3] J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. *Proc. of the ACM SIGMOD Int. Conf. On Management of Data (SIGMOD'00)*, Dallas, TX, May 2000.
- [4] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. Hsu. FreeSpan: Frequent pattern-projected sequential pattern mining. *Proc. of the 6th ACM SIGKDD Int. Conf. On Knowledge Discovery and Data Mining (KDD'2000)*, Boston, MA, August 2000.
- [5] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. *Proc. of the Int. Conf. On Data Engineering (ICDE'01)*, Heidelberg, Germany, April 2001.
- [6] M. J. Zaki, SPADE: An Efficient Algorithm for Mining Frequent Sequences. *Machine learning Journal*, special issue on Unsupervised Learning (Doug Fisher, ed.), pp. 31- 60, 2001.
- [7] M. Kantardzic, M. Soliman, An Approach for Mining Frequent Sequences in Distributed Environment. *Proc. of the Int. Conf. On Information and Knowledge Engineering*, Las Vegas, USA, pp.553-559, 2002.
- [8] M. Soliman, M. Kantardzic, Towards Building a Distributed Mining Approach. *Proc. of IEEE ISSPIT*, Marrakesh, Morocco, pp 305-308, 2002.
- [9] A. Badia, and M. Kantardzic, A Data Mining Framework for Semantically Distributed Databases. *Proc. of IASTED International Conference on Artificial Intelligence and Soft Computing*: Banff, Canada, July 2003.
- [10] R. Wirth, M. Borth, J. Hipp. When Distribution is Part of the Semantics: A New problem Class for Distributed Knowledge Discovery. *Proc. of the PKDD Workshop on Ubiquitous Data Mining for Mobile and Distributed Environments*, Freiburg, Germany, pp. 56-64, 2001.
- [11] C. Clifton. Privacy Preserving Distributed Data Mining. *Project funded by the Purdue Research Foundation*, August 2002-August 2004.



98 Data Mining IV

- [12] T. Li, S. Zhu and M. Ogihara. A New Distributed Data Mining Model Based on Similarity. *Proc. of ACM SAC Data Mining Track*, Melbourne, Florida, USA 2003.
- [13] M. Charikar. Similarity Estimation Techniques from Rounding Algorithms. *Proc. of the 34th annual ACM Symposium on Theory of Computing*, pp 380-388, 2001.
- [14] A.Z. Border, M. Charikar, A. Fricze, and M. Mitzenmacher. Min-wise independent permutations. *Proc. 30th STOC*, pp.327-336, 1998.
- [15] Extended Jaccard Similarity, www.lans.ece.utexas.edu/~strehl/diss/node56.html
- [16] D. Fasulo. An Analysis of Recent Work on Clustering Algorithms. *Technical Report 01-03-02, Department of Computer Science and Engineering*, University of Washington, Seattle, WA 98195, April 1999.
- [17] L. Kaufman and P. J. Rousseeuw *Finding Groups of Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc., 1990.