2

Privacy-Preserving Clustering on Distributed Databases: A Review and Some Contributions

Flavius L. Gorgônio and José Alfredo F. Costa
Federal University of Rio Grande do Norte
Brazil

1. Introduction

Clustering is the process of discovering groups within high-dimensional databases, based on similarities, with a minimal knowledge of their structure. Traditional clustering algorithms perform it over centralized databases, however, recent applications require datasets distributed among several sites. Therefore, in distributed database environments, all distributed data must be concentrated on a central site before applying traditional algorithms.

There is a series of limitations which hinder the utilization of traditional data mining techniques on distributed databases. The approach commonly taken, the gathering of all distributed databases in a central unit, followed by algorithm application, is strongly criticized, because in these cases, it is important to take into consideration some issues, namely: the possibility of existence of similar data with different names and formats, differences in data structures, and conflicts between one and another database (Zhang et al., 2003). Besides, the unification of all of the registers in a single database may take to the loss of meaningful information, once that statistically interesting values in a local context may be ignored when gathered to other ones in a larger volume.

On the other hand, integration of several database in a single location is not suggested when it is composed of very large databases. If a great organization has large disperse databases and needs to gather all the data in order to apply on them data mining algorithms, this process may demand great data transference, which may be slow and costly (Forman & Zhang, 2000). Moreover, any change that may occur in distributed data, for instance inclusion of new information or alteration of those already existing will have to be updated along with the central database. This requires a very complex data updating strategy, with overload of information transference in the system. Furthermore, in some domains such as medical and business areas whereas distributed databases occurs, transferring raw datasets among parties can be insecure because confidential information can be obtained, putting in risk privacy preserving and security requirements.

Due to all of these problems related to database integration, research for algorithms that perform data mining in a distributed way is not recent. In the end of the 90s, several researches about algorithms to effectuate distributed data mining started to appear, having been strengthened mainly by the rise of the distributed database managing systems and of the need for an analysis of such data in the way that they were dispersed (DeWitt & Gray, 1992; Souza, 1998). Currently, there is an increasing demand for methods with the ability to process clustering securely that has motivated the development of algorithms to analyze each database separately and to combine the partial results to obtain a final result. An updated bibliography about the matter can be obtained in (Bhaduri et al., 2006).

This chapter presents a wide bibliographical review on privacy-preserving data clustering. Initially, different alternatives for data partitioning are discussed, as well as issues related to the utilization of classification and clustering ensembles. Further, some techniques of information merging used in literature to combine results that come from multiple clustering processes are analyzed. Then, are discussed several papers about security and privacy-preserving in distributed data clustering, highlighting the most widely used techniques, as well as their advantages and limitations. Finally, authors present an alternative approach to this problem based on the partSOM architecture and discuss about the confidentiality of the information that is analyzed through application of this approach in geographically distributed database cluster analysis.

2. Bibliographic review

Currently, a growing number of companies have strived to obtain a competitive advantage through participation in corporative organizations, as local productive arrangements, cooperatives networks and franchises. Insofar as these companies come together to overcome new challenges, their particular knowledge about the market needs to be shared among all of them. However, no company wants to share information about their customer and transact business with other companies and even competitors, because it is needed to maintain commercial confidentiality and due to local legislation matters.

Hence, a large number of studies in this research area, called privacy preserving data mining – where security and confidentiality of data must be maintained throughout the process – have been prompted by the need of sharing information about a particular business segment among several companies involved in this process, avoiding jeopardizing the privacy of its customers. A comprehensive review of these studies is presented below.

2.1 Data partitioning methods

There are two distinct situations that demand the need for effecting cluster analysis in a distributed way. The first occurs when the volume of data to be analyzed is relatively great, which demand a considerable computational effort, which sometimes is even unfeasible, to accomplish this task. The best alternative, then, is splitting data, cluster them in a distributed way and unify the results. The second occurs when data is naturally distributed among several geographically distributed units and the cost associated to its centralization is very high.

Certain current applications hold databases so large, that it is not possible to keep them integrally in the main memory, even using robust machines. Kantardzic (2002) presents three approaches to solve this problem:

i. Storing data in a secondary memory and clustering data subsets separately. Partial results are kept and, in a posterior stage, are gathered to cluster the whole set;
ii. Using an incremental clustering algorithm, in which every element is individually brought to the main memory and associated to one of the existing clusters or allocated in a new cluster. The results are kept and the element is discarded, in order to grant space to the other one;
iii. Using parallel implementation, in which several algorithms work simultaneously on stored data, increasing efficacy.

In cases in which the data set is unified and needs to be divided in subsets, due to its size, two approaches are normally used: horizontal and vertical partitioning (Figure 1). The first approach is more used and consists in horizontally splitting database, creating homogeneous data subsets, so that each algorithm operates on different records considering, however, the same set of attributes. Another approach is vertically dividing the database, creating heterogeneous data subsets; in this case, each algorithm operates on the same records, dealing, however, with different attributes.

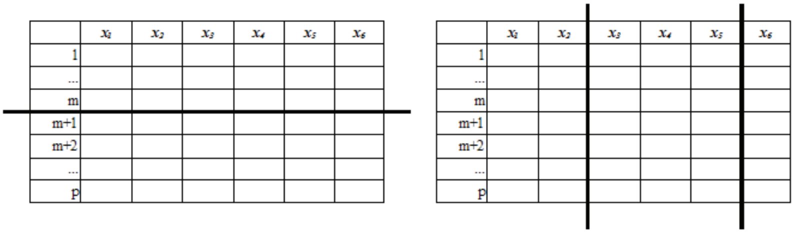


Fig. 1. Horizontal and vertical partitioning

In cases in which the data set is already partitioned, as in applications which possess distributed databases, besides the two mentioned approaches, it is still possible meet situations in which data is simultaneously disperse in both forms, denominated arbitrary data partitioning which is a generalization of the previous approaches (Jagannathan & Wright, 2005).

Both horizontal and vertical database partitioning are common in several areas of research, mainly in environments with distributed systems and/or databases, to which commercial application belongs. The way how data is disperse in a geographically distributed database environment depends on a series of factors which not always regard the task of clustering analysis as a priority inside the process. Operational needs of these systems may directly influence in the form of data distribution and data mining algorithms must be robust enough to cope with these limitations. For instance, in a distributed databases project, it is important to generate fragments which contain strongly related attributes, in order to guarantee a good performance in storage operations and information recovery (Son & Kin, 2004).

Recent studies on data partitioning technologies seek to meet this demand, particularly in situations in which incompatibilities between data distribution and queries carried out may affect system performance. When applied to distributed databases, vertical partitioning offers two great advantages which may influence system performance. First the frequency of queries necessary to access different data fragments may be reduced, once that it is possible to obtain necessary information with a smaller number of SQL queries. Second, the amount of recovered and transferred unnecessary information in a traditional query to memory may also be reduced (Son & Kin, 2004).

If, on one hand, data partition methods keeps focusing on queries performance, seeking for the more suitable number of partitions to make the recovery process of stored data quicker, the presence of redundant or strongly correlated variables in a process of cluster analysis with self-organizing maps, on the other hand, is not recommended (Kohonen, 2001). Therefore, in order to obtain better results in data analysis, the most recommended is geographically distributing data so that correlated variables stay in different units. Nonetheless in situations in which databases are already geographically distributed – not being possible to alter their structure – and the existence of strongly correlated structures may impair results, it is possible to utilize statistical techniques, such as Principal Components Analysis (PCA) or Factor Analysis to select a more suitable subset of variables and reduce these problems.

2.2 Classification and cluster ensemble

Cluster ensembles may shortly be defined as a combination of two or more solutions come from application of different algorithms or variations of a same algorithm on a dataset, or even, on subsets thereof. The combination of several clustering algorithms has the objective of producing more consistent and reliable results than the utilization of individual algorithms does, which is why cluster ensembles have been proposed in several application which involve data clustering and classification.

The definition of cluster ensembles presented in the previous paragraph is deliberately generic, in order to include several possibilities of utilization of cluster algorithms and combination of results existing in the literature. In fact, Kuncheva (2004) suggests four approaches for classifying system development, which may be extended to cluster ensemble development:

i. Application of several instances of a same algorithm on the same database, changing the initialization parameters of the algorithm and combining its results;
ii. Application of different clustering algorithms on a same database, intending to analyze which algorithm obtains the best data clustering;
iii. Application of several instances of a same clustering algorithm on subsets of slightly different samples, obtained with or without reposition;
iv. Application of several instances of a same clustering algorithm on different subset of attributes.

Combining the result of several clustering methods, creating a cluster ensemble, appeared as a direct extension of the systems which use multiple classifiers (Kuncheva, 2004). Using of the multiple classifiers systems, based on the combination of the results of different classification algorithms, has been proposed as a method for developing high-performance classifiers systems with applications in the field of pattern recognition (Roli et al., 2001).

Theoretical and practical studies confirm that different kinds of data require different kinds of classifiers (Ho, 2000), which, at least theoretically, justifies ensembles utilization. Nevertheless, far from being consensual, the use of multiple classifier systems and cluster ensembles is questioned by several authors, both for requiring a greater computing effort, and for requiring the utilization of intricate mechanism of result combination (Kuncheva, 2003).

Roli et al. (2001) assert that the increasing interest in multiple classifier systems results from difficulties in deciding the best individual classifier for a specific problem. These authors analyze and compare six methods to project multiple classifier systems and conclude that, even though these methods have interesting characteristics, none of them is able to ensure an ideal project of a multiple classifier system.

Ho (2002) criticizes the multiple classifier systems, stating that, instead of concentrating efforts in seeking for the best set of attributes and the best classifier, the problem becomes seeking for the best set of classifiers and the best method of combining them. He also states that, later, the challenge becomes seeking for the best set of combining methods of results and the best way of using them. The focus of the problem is, then, forgotten and, more and more, the challenge becomes the usage of more complicated combining theories and schemes.

Strehl (2002) states as widely known the conception that the combination of multiple classifiers or multiple regression models may offer better results if compared to a single model. However, he alerts that there are no acknowledged effective approaches to combine clustering multiple non-hierarchical algorithms. In this work, the author proposes a solution to this problem using a framework to segmentation of consumers based on behavioural data.

In spite of all reported, both multiple classifier systems and cluster ensembles have been more and more used. Zhao et al. (2005) present a good review on the area, thus reporting several applications for classifiers ensembles based on neural networks, which include recognition of patterns, illness diagnostics and classification tasks. Oza & Tumer (2008) do the same in a more recent work, in which they present real applications, where using classifier ensembles has been obtaining a greater success in comparison to using individual classifiers, including remote sensoring, medicine and pattern recognition. Fern (2008) analyses how to combine several available solutions to create a more effective cluster ensemble, based on two critical factors in the performance of a cluster ensemble: quality and diversity of solutions.

Leisch (1998), one of the pioneers in the branch of cluster ensembles, introduced an algorithm named bagged clustering, which performs several instances of K-means algorithm, in the attempt of obtaining a certain stability in the results and combines partial results through a hierarchical partitioning method.

In another introductory work on distributed clustering analysis, Forman & Zhang (2000) present a tendency which parallelizes multiple algorithms based on centroids, like K-means and expectation maximization (EM) in order to obtain a greater efficacy in the process of data mining in multiple distributed databases. The authors reinforce the need for worrying about reducing the communication overload among the bases, reduce processing time and minimize the necessity for powerful machines with broad storage capacity.

Kargupta et al. (2001) highlight the absence of algorithms which effect clustering analysis in heterogeneous data sets using Principal Component Analysis (PCA) in a distributed way and present an algorithm denominated Collective Principal Component Analysis (CPCA) to analyze high dimension heterogeneous data clusters. The authors also discuss the effort of reducing the rate of data transference in a distributed data environment.

Haykin (2001) describes the neural networks as processors massively distributed in a parallel way, which suggests that the training of a cluster ensemble based on neural network may be done in a distributed way (Vrusias et al., 2007). Besides, there are, in literature, several researches striving to approach parallel neural network training, in particular, of self-organizing maps (Yang & Ahuja, 1999; Calvert & Guan, 2005; Vin et al., 2005).

This type of training generates innumerable challenges, once that, as a rule, neural network algorithms are non-deterministic and based on a set of initialization and training parameters. Thus, as neural networks normally are highly responsive to initialization parameters, choices done during the training process end up directly influencing the achieved results.

Some researches in this area exploit this particularity pertaining to neural networks to create ensembles based on the execution of a same algorithm with different initialization and training sets. In this approach, bootstrap aggregating, bagging and boosting are some of the techniques which have been used with some relative success in ensemble training, as described in (Breiman, 1996; Freud & Schapire, 1999; Frossyniotiset al., 2004; Vin et al., 2005). Even though such techniques have been demonstrating the existing variation of probabilities and the benefits of these approaches, some problems became evident, which need to be considered while training ensembles concurrently with subsets of distinct inputs, such as computational cost and result fusion mechanisms.

The utilization of clusters of computers and computational grids has been frequently considered in performing distributed training of several types of neural networks, as multilayer perceptron networks and self-organizing maps (SOM), as well as radial base function networks (RBF) (Calvert & Guan, 2005). Hämäläinen (2002) presents a review on several parallel implementations utilizing self-organizing maps.

Neagoe & Ropot (2001) present as neural classifying model, denominated concurrent self-organizing maps (CSOM), which is composed of a collection of small SOM networks. CSOM model present some conceptual differences from tradition SOM model – the major is in the training algorithm, which is supervised. The number of SOM networks used in the model must be equal to the number of output space classes. To each individual SOM network, a specific training subset is used, so that the network is trained to have expertise in a certain output space class. Hence, in the end of the training stage, each SOM became expert on the class that it represents.

During the classifier utilization, the map which presents the lesser quantified error is declared winner and its index is the index of the class to which the pattern belongs. In tests performed with CSOM model, the authors consider three applications in which this model presents fair results: face recognition, speech recognition and multi-spectral satellite images (Neagoe & Ropot, 2002; Neagoe & Ropot, 2004).

Arroyave et al. (2002) present a parallel implementation of multiple SOM networks using a Beowulf cluster, with application on the organization of text files. In this approach, a huge self-organizing map is divided into several parts with the same size and distributed among the machines of the cluster. The training is also performed in a distributed way, so that every slave unit receives each of the input data from the master unit and returns to its own best match unit, which is shared with the other machines in a cooperative process.

Vrusias et al. (2007) propose an algorithm to train self-organizing maps, in a distributed way, by utilizing a computational grid. The authors propose a SOM cluster training architecture and methodology distributed along a computational grid, in which it is considered: the ideal number of maps in the ensemble, the impact of the different kinds of data used in the training and the most appropriate period for weight updating.

The training foresees periodical updates in map weight, in which the partial results of each units are sent to the master unit in the beginning of each training stage, and the latter is responsible for effecting the mean of received data and send them to the slaves units. Once that there is much integration among the parts along the training, time spent in this operation may be long, directly influencing in the map training time. Therefore, according to the authors, this approach only has results in dedicated clusters.

The authors performed a series of experiments and obtained important conclusions which can be extended to other SOM network parallel training algorithms:

i. If the latency time of the ensemble members the periodical weight adjusts and the synchrony time of the maps are very short, in comparison to the computational time of each training stage, the utilization of a SOM ensemble brigs about good results, regarding training time and accuracy;
ii. In the performed tests, the ideal number of maps in an ensemble was between 5 and 10 networks;

iii. The choice of the several utilized parameters in the training (learning and decrement rate) and the frequency which calculations of the map average are also factors of great importance in reducing mean square error;

iv. SOM ensemble presents quite superior results as the dimension of the data set increases.

Georgakis et al. (2005) propose the utilization of a self-organizing ensemble in attempt to increase performance in document organization and recovery. Several maps are simultaneously trained, with slightly different subsets. In posterior stage, maps are compared and the neurons of the ensemble members are lined to create the final map. The most similar neurons of each map are combined, through an arithmetic mean of their synaptic weights, to create a new neuron in the final map. During the training, uniformly distributed samples are taken from the data set to feed each of the members of the ensemble. The algorithm is used to partition a cluster document repository, according to its semantic contents. The performed experiments show that the performance of this algorithm is superior to the performance of traditional SOM, regarding to data recovery accuracy based on its semantic contents.

Cluster ensemble application on different attribute subsets has been analyzed mainly in image segmentation. Picture SOM or PicSOM in a hierarchical architecture in which several algorithms and methods can be applied jointly for image recovering based on contents (Laaksonen et al., 1999; Laaksonen et al., 2000; Laaksonen et al., 2002). Originally, PicSOM utilizes multiple instances of TS-SOM algorithm – which is composed by structured trees of self-organizing maps, hierarchically organized (Koikkalainen, 1994). Each TS-SOM is trained with a different set of characteristics, such as colour, texture or form.

PicSOM architecture is an example of SOM network combination, whose result is a solid system for image recovery based on content similarity. Georgakis & Li (2006) propose a PicSOM modification using a technique named bootstrapping during training stage. This technique divides randomly input space in a series of subsets which are used in the training stage of SOM. Then, the trained maps are combined into a single map to create the final result. According to the authors, this approach obtains more accurate results that original PicSOM.

Yu et al. (2007) propose an architecture to segment images based on an expectation maximization algorithm ensemble. This architecture starts extracting colour and texture information from the image, which are processed separately. A posterior stage combines the neighbouring regions individually segmented, taking into consideration information related to the position of pixels in an image. Jiang & Zhou (2004) present another proposal of SOM network ensemble usage to image segmentation, based only on information about colour and pixel position. The proposed approach combines partial results through a weighted voting scheme evaluated through mutual information index, which measures similitude among partitions.

Most of cluster analysis algorithms deals only with number data, even though there are some varieties of these algorithms specifically developed to handle with categorical data. Concerning databases with both kinds of values, some adjusts are necessary during the stage prior to processing, like, for instance, categorical data conversion in mutually exclusive binary data. Such a conversion elevates database dimensionality even more, once that it creates an additional column for each possible attribute value. Some alternative approaches for coding categorical variables into number variables are presented in the literature. Rosario et al. (2004) propose a method which analyzes how to determine order and spacing among nominal variables and how to reduce the number of distinct values to be considered, based on Distance-Quantification-Classing approach.

He et al. (2005) analyze the influence of data types in the process of clustering and propose a different approach, effecting a division of the set of attributes into two subsets – one only with number attributes and another with only categorical attributes. Thereafter, they propose clustering of each of the subset in an isolated way, using algorithms suitable for each of the types. Eventually, each results of clustering process are to be combined into a new database, which is submitted, again, to a categorical data clustering algorithm.

Luo et al. (2007) propose an alternative method to data partitioning for generating ensemble training subsets, based in adding noise to original data. This method proposes utilizing artificial noise to produce variability in data during execution of clustering algorithms. The artificial data generated are an approximation of real data, in which are computed the mean and standard-deviation of the sample in order to generate data from Gaussian distribution found.

Recently, several works on the branch of cluster ensembles applied to bioinformatics, particularly to genic expression analysis. Silva (2006) investigates the utilization of cluster ensembles in genic expression analysis. Data is analyzed through three different cluster algorithms (K-means, EM and average linkage hierarchical clustering) and results are combined through different techniques, such as voting, relabeling and graphs. Results show that this approach obtains a superior result than the utilization of individual techniques, particularly when composite ensembles are used by several algorithms.

Faceli (2006) proposes an architecture for exploratory data analysis through clustering techniques. Such an architecture is composed by a multi-objective cluster ensemble, which executes several conceptually different algorithms with several parameter configurations, combines partitions resulting from this algorithm and selects partitions with the best results with different validation measures. Among the databases used for validation of the proposal, some genes expressions are also included.

2.3 Combining ensemble results

A problem which is inherent to cluster ensembles is partial results combining. Strehl (2002) describes efficiently this matter and presents three most common approaches to solve this problem under different points of view. The first approach consists of analyzing similitude among different partitions produced through utilization of similarity metrics among partitions. The second uses hyper-graphs to represent relationship among the objects and applies hyper-graph partitioning algorithms on them to find the clusters. In the third approach the elements of input set are labeled and, then, labels are combined to present a final result, normally through some voting system.

Strehl & Ghosh (2002) introduce the problem of combining multiple partitions of a set of objects into a single partition consolidated from obtained partial labels. In short, the objective of this approach is obtaining a set of labels which correspond to the result of each partition and, considering only partial results, combining them in order to obtain a consensual result, not taking into consideration previous characteristics about the objects which determined the partitions. In fact, this is the most popular way of result fusion among the three presented by Strehl (2002) for data clustering tasks and difficulties associated to its utilization have been investigated in several other works which approach this issue (Dimitriadou et al., 2001; Frossyniotis et al., 2004; Zhou & Tang, 2006; Tumer & Agogino, 2008).

Some SOM ensemble-based works introduce specific result combining techniques through map fusion techniques. In the approach proposed by Vrusias et al. (2007), a great self-organizing map is divided into small sub-maps and sent to the several units of a computational grid, to be trained in parallel. Each unit trains its own sub-map with a subset of different data. In this case, result fusion is done with base on means of individually trained maps. The mean values of the neurons are obtained through the arithmetic mean, in each dimension, for each SOM instance in the ensemble.

This calculation is made after a prefixed number of interactions in training stage. Once that each neural network is trained with its base on its respective dataset, this process tends to decrease as to accuracy in comparison to training a single network with the all data available, however more efficacy is obtained as to time spent in training. On the other hand the ensemble has a potential to generate better results than a single neural network once that a greater amount of training can be performed in the same time interval.

In another proposal, Georgakis et al. (2005) suggest a SOM ensemble simultaneously trained with slightly different data subsets and used to organize and recover documents. In this case, result fusion is also performed through an arithmetical mean of its synaptic weights, but combining the most similar neurons of each map in order to compose a new neuron of the final map. The difficulty of this proposal is maintaining the topology of partial maps in the final map. The same strategy is used in a later work for image recovery based on contents (Georgakis & Li, 2006).

Hore et al. (2006) describe some ways of results fusion based on label combining and show that these methods are not suitable for application on very large databases. Which is why, they present a proposal of cluster ensembles which extracts a set of centroids, labels these centroids and combines results to identify the clusters of the original dataset. Besides, the work includes an additional process to eliminate malformed clusters due to initialization or data distribution failures or to existing noises.

2.4 Security and privacy preserving data mining

Data security and privacy-preserving are among the primal factors which motivate creation and maintenance of distributed database (Chak-Man et al., 2004). Many organizations, then, maintain their databases geographically distributed, as a way to increase the security of their information; for if, by chance, one of their security policies fails, the intruders has access to only a part of the existing information.

The need for assure information confidentiality during a knowledge extraction process in databases is a very current area of research in scientific society (Kapoor et al., 2007). Researches involving data security and privacy-preserving in databases had an unexpected increase in the last years, caused by growing preoccupation of individuals in sharing their personal information via Internet, as well as the worry of business in assuring security of this information (Verykios et al., 2004).

It is known that combining several sources of data during a KDD process increases analysis process, even though it jeopardizes security and privacy-preserving of data involved in the process (Oliveira & Zaïane, 2007). Wherefore, data mining algorithms which operate in distributed way must take into consideration not only the way data is distributed among the units, in order to avoid unnecessary transferences, rather they must also ensure that transferred data is protected against occasional attempts of undue appropriation attempts. Inasmuch as digital repositories have become more and more susceptible to attacks and business and organizations all over the world have frequently been held responsible for abuses, once that governments have been adopting more and more rigorous legislations pertaining to collected data privacy-preserving, these worries have been demanding new advances in the area of distributed data mining (Kapoor et al., 2006).

A potentially interesting market to distributed data mining is corporative organizations, composed of a significant number of businesses which work around one principal activity, such as local productive arrangements, business agglomerations, corporative networks, cooperatives and franchises. Simultaneous application of data mining algorithms on databases owned by several companies which act on the same branch allows obtaining more complete information and more accurate knowledge on this segment, augmenting the knowledge of the group about that area of business (Thomazi, 2006). Nonetheless, in spite of the obvious advantages of this approach, most of businesses participating in corporative organizations decide for analyzing on their individual databases. Security restrictions hinder sharing information from customers among partner companies in several countries and create a series of problems related to privacy-preserving, preventing companies from adopting this strategy.

Privacy-preserving cluster analysis rises as a solution to this problem, permitting that the parties to cooperate among them in knowledge extraction, preventing obligation of each of them of revealing their individual data to the others. This approach concentrates its efforts in algorithms which assure privacy and security to data involved in the process, mainly in applications in which security has fundamental importance, for instance, in medical and commercial applications (Berkhin, 2006; Silva, 2006).

Verykios et al. (2004) discuss the state of the art in data security and privacy, presenting the most common three techniques: the ones based on heuristic, which seek purposely to alter some database values, avoiding, however, losses in the process; the ones based on cryptography, which codify data in order to avoid access to information from other parties; and the ones based on data rebuilding, which use some technique in order to introduce perturbation in data, keeping existing relations among them. The authors present one more classification of the most common data mining algorithms according to the presented techniques.

The first references to security related problems in KDD problems arose even in the 90s (O'Leary, 1991; Piatetsky-Shapiro, 1995; Clifton & Marks, 1996). Nevertheless, the first researches with concrete results on privacy-preserving data mining area were published by Agrawal & Ramakrishnan (2000) and Lindell & Pinkas (2000). The former, based on a data rebuilding technique known as randomization, which introduces noise along to actual data, avoiding that data may be reconstituted, keeping, however, existing relations among them. The latter, using a cryptography technique named Secure Multi-party Computation (SMC), to classify data on horizontally distributed bases. SMC technique was proposed by Goldreich et al. (1987), from original idea proposed by Yao (1986).

Even though both approaches do not consider the need for data transference reduction among the units, several other works which followed are direct extensions of the these techniques. Agrawal & Aggarwal (2001) made continuity of the first work, adding more privacy to data and including the utilization of EM algorithm during data reconstruction. Following, Evfimievski et al. (2002) adapt the algorithm for association rule extraction on categorical attributes, adding noise to data and measuring the influence of these noises in final result. The technique used in the second work, based on SMC, was investigated in several other works. In spite of its efficacy in guaranteeing mined data security, its application in data mining tasks has ended up being inefficient, due to its complexity

(Clifton et al., 2002; Du & Atallah, 2001). More recently, some variations of this technique have been investigated, in the sense of reducing complexity.

Kantarcioglu & Vaidya (2002) criticize the security of randomization processes and the complexity of SMC algorithms, besides the need of all of these units for being connected during the process. As an alternative, they present an architecture which cheats these limitations in association rule extraction in distributed databases with information about clients. Nevertheless, this architecture requires the database to be entirely transferred to the central unit, which makes it unfeasible in many data mining applications. Vaidya & Clifton (2003) propose distributed implementation of K-means algorithm, based on SMC technique, for cluster analysis on vertically distributed databases. Lin et al. (2005) adapt the same idea for utilization along with EM algorithm. More recently, Vaidya et al. (2006) summarize the techniques most used in privacy-preserving data mining in prediction and description, both on horizontally and vertically partitioned databases.

Statistical techniques have been used to ensure data security and privacy-preserving in clustering tasks. Merugu & Ghosh (2003) present an architecture for distributed data clustering based on a technique named generative models, which causes data perturbation based on a statistical model, in order to guarantee privacy. Klusch et al. (2003) propose a distributed clustering algorithm based on local density estimation. This algorithm works in a distributed way, using an objective function to extract local partition density and combines the partial clusters sending information about clustering nucleus to the central unit. Data privacy and security are kept, once that only information about the clustering nuclei is shared.

Estivill-Castro (2004) proposes a method which combines a protocol of communication between two or more parts based on SMC and the utilization of K-medoids, a more robust variation of K-means, for clustering vertically partitioned data. Another approach based on the usage of K-medoids proposes the use of a cryptography technique denominated homomorphic ciphering to permit data sharing among the parties without jeopardizing security (Zhan, 2007). Later, Zhan (2008) expanded this technique to other data mining tasks. Jha et al. (2005) propose the utilization of K-means through two security protocols, polynomial evaluation and homomorphic evaluation.

The problem which arises when confidential information may be deduced from data made available to non-authorized users is known as the problem of inference in databases (Verykios et al., 2004; Farkas & Jajodia, 2002). Oliveira & Zaïane (2003) introduce a set of methods for data perturbation, based on geometrical transformations (translation, scale alteration) in *p*-dimensional space. Initially any attributes that may be used for individual identification of objects are eliminated. Then, the method effects several geometrical transformations on data, keeping statistical relations among them, but preventing them to be reconstructed.

Later, Oliveira & Zaïane (2004) propose improvement in the method of transformation based on geometric rotation in order to protect attribute values while these are shared in a clustering process. The main advantage of the proposed method is that it is independent from clustering algorithms. More recently, the authors combine results of previous studies in a new method for privacy-preserving cluster analysis, denominated Dimensionality Reduction-Based Transformation (DRBT), with applications on the commercial area (Oliveira & Zaïane, 2007).

Jagannathan & Wright (2005) introduce the concept of arbitrary data partitioning, which is the generalization of horizontal and vertical partitioning and present a method for data clustering tasks with K-means algorithm on arbitrarily partitioned data bases. This method utilizes a cryptography-based protocol to guarantee data privacy. Jagannathan et al. (2006) suggest a safer variant of K-means algorithm previously proposed, however for clustering on horizontally distributed databases.

İnan et al. (2006) and İnan et al. (2007) approach privacy-preserving clustering analysis through an algorithm which permits to build a dissimilarity matrix among objects on horizontally distributed databases, through SMC to ensure security. The algorithm works suitably with numerical and categorical attributes and the built dissimilarity matrix may be applied to other data mining tasks. Kapoor et al. (2007) present an algorithm named PRIPSEP (PRIvacy Preserving SEquential Patterns), based on SMC technique, which permits mining sequential patterns on distributed database, while it maintains the privacy-preserving of the individual.

In some more recent works, Vaidya (2008) presents and discusses several data mining methods which operate in a distributed way on vertically partitioned databases, while Kantarcioglu (2008) does the same to methods which operate in a distributed way on horizontally partitioned databases. Fung et al. (2008) propose an architecture for data clustering analysis which convert a cluster analysis process in a classification activity. The proposed algorithm carried out data clustering and associates data in a set of classes. Then, it codifies actual data through labels and transmits codified data as well as respective classes to other units, thus preserving privacy of data involved in the process.

3. The partSOM architecture clustering process

This section presents a cluster ensemble methodology for privacy preserving clustering in distributed databases, using traditional and well known algorithms, such as self-organizing maps and K-means. The proposed methodology combines a clustering architecture, the partSOM architecture (Gorgônio & Costa, 2010), with principles of vector quantization, building a cluster ensemble model that can be used to cluster analysis in distributed environments composed by a set of partner companies involved in this process, avoiding jeopardizing the privacy of their customers.

The main idea of this process is focused on omission of real information about customers, changing a set of real individuals for one (or more) representative (and fictional) individual with similar statistical characteristics of the real individuals. This strategy, based on vector quantization principles, enables that a group of individuals with similar characteristics to be able to be represented by a single individual (vector) corresponding to that group. As illustrated in Figure 2, the vectors {$\mathbf{x}_1$, $\mathbf{x}_3$, $\mathbf{x}_4$, $\mathbf{x}_7$, $\mathbf{x}_8$} can be represented by $\mathbf{w}_1$ vector and {$\mathbf{x}_2$, $\mathbf{x}_5$, $\mathbf{x}_6$, $\mathbf{x}_9$} can be represented by $\mathbf{w}_2$ vector. This strategy is used to reduce the amount of space required to store or transmit a dataset and has been widely used by clustering tasks and data compression of signals, particularly voice and image.

The partSOM architecture presents a strategy to carry out cluster analysis over distributed databases using self-organizing maps and K-means algorithms. This process is separated in two stages: initially, data are analyzed locally, in each distributed unit. In a second stage, a central unit receives partial results and combines them into an overall result.

The partSOM algorithm, embedded in partSOM architecture, consists of six steps and is presented as it follows. An overview of the complete architecture is showed in Figure 3.

1. A traditional clustering algorithm is applied in each local unit, obtaining a reference vector, known as the codebook, from each local data subset;

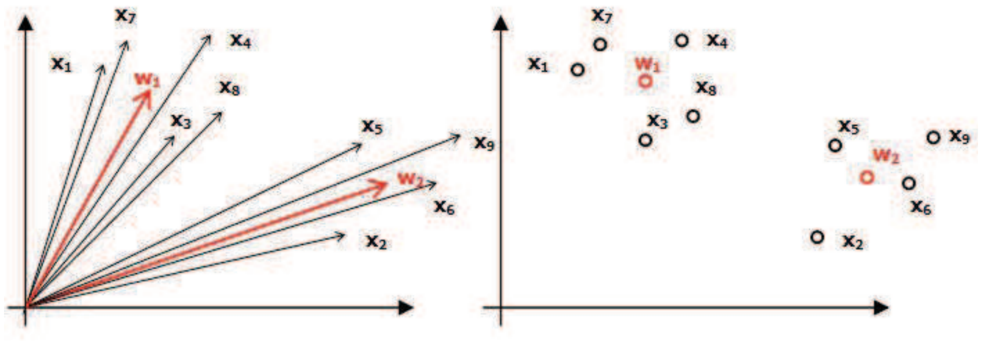
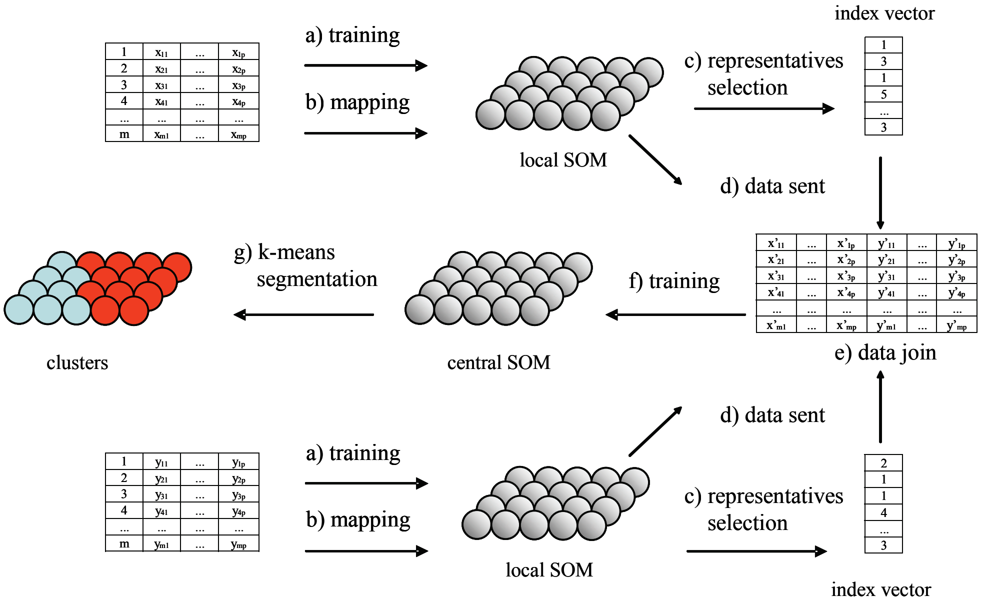


Fig. 2. Example of a vector quantization process in a bidimensional plan



1	x_{11}	...	x_{1p}
2	x_{21}	...	x_{2p}
3	x_{31}	...	x_{3p}
4	x_{41}	...	x_{4p}
...	...	...	...
m	x_{m1}	...	x_{mp}

x'_{11}	...	x'_{1p}	y'_{11}	...	y'_{1p}
x'_{21}	...	x'_{2p}	y'_{21}	...	y'_{2p}
x'_{31}	...	x'_{3p}	y'_{31}	...	y'_{3p}
x'_{41}	...	x'_{4p}	y'_{41}	...	y'_{4p}
...	...	...	...	...	...
x'_{m1}	...	x'_{mp}	y'_{m1}	...	y'_{mp}

1	y_{11}	...	y_{1p}
2	y_{21}	...	y_{2p}
3	y_{31}	...	y_{3p}
4	y_{41}	...	y_{4p}
...	...	...	...
m	y_{m1}	...	y_{mp}

Fig. 3. An overview of the partSOM architecture with SOM and K-means algorithms

2. Each input data is compared with codebook issues and the index corresponding to the most similar vector present in the codebook is stored in an index vector. So, a data index vector is created based on representative objects instead of original objects;
3. Each remote unit sends the codebook and the index vector to the central unit, which will conduct the unification of all partial results;
4. The central unit is responsible for receiving index vectors and codebooks from each local unit and combining partial results and building a whole database. In this process, index vector issues are substituted by the similar issues in the codebook;
5. The clustering algorithm is applied on the whole database obtained in previous step, to identify existing clusters in the collective database;

6. A segmentation algorithm is applied on results obtained after the final cluster process, in order to improve the quality of the visualization results.

Despite the difference between the original and the remounted database, which are slightly different, the topology and statistical characteristics from original data is maintained, because representative objects in the index vector are very similar to the original data, as shown in several experiments (Gorgônio & Costa, 2008; Gorgônio & Costa, 2010). As a matter of fact, this is an important characteristic of the partSOM architecture, since results obtained with this architecture can be generalized as being equivalent to the clustering process of the entire original databases.

The architecture presented was developed focusing on geographically distributed databases, independently of criteria used in partitioning. Wherefore, a solution which is stable in any form of partitioning has been required, whether it is horizontal vertical or arbitrary, even though additional techniques may be used to better its performance in specific domains.

4. Some contributions to partSOM clustering process

This section presents some contributions to increase security and privacy preserving in a clustering process using the partSOM architecture. First of all, it is proposed a data pre-processing stage, which are removed all information that could be used to identify an individual. Following, it is proposed a pruning algorithm to reduce the amount of data transferred between the local and central units. Finally, it is proposes the use of a covariance matrix from each local data unit to reduce losses during the process of vector quantization.

4.1 The pre-processing stage

In real world applications, raw data usually are named *dirty data*, because they can contain errors, missing values, redundant information or are incomplete and inconsistent. So, most of data mining process needs a pre-processing stage that objectives to carry out tasks such as data cleaning, data integration and transformation, data reduction, although this important step is sometimes neglected in data mining process.

Conventionality, a relational database is a set of two-dimensional tables interrelated by one or more attributes. Each table is a two-dimensional structure of rows and columns where each row represents a record from the database and each column represents an attribute associated with that record. Figure 4 suggests a sample of a typical table in a database.

After pre-processing stage, data are usually arranged in single table known as data matrix, which must satisfy the requirements of the chosen algorithm. The data matrix **D** is formed by a set of n vectors, where each vector represents an element of the input set. Each vector has p components, which correspond to the set of attributes that identify it. A data matrix example, related to the previous presented table in Figure 4, is shown in Figure 5.

In this example, some attributes were removed, others were transformed and the whole dataset was normalized. As discussed in literature (Hore et al., 2006), this stage contributes to privacy and security maintenance of data and information stored in database, because real data are replaced by a set of representatives with same statistical distribution of original data. Thus, since only codebook and index vector are sent to the central unit and no real information is transferred, the security is maintained.

#	Name	Sex	Age	Wage	Civil State	Children	...	State
1	A. Araújo	M	39	2.300,00	Married	3	...	RN
2	Q. Queiroz	F	82	1.350,80	Widowed	2	...	PB
3	W. Wang	M	21	720,50	Single	1	...	CE
4	E. Eudes	F	18	1.420,00	Single	0	...	SP
5	S. Silva	M	16	450,00	Single	0	...	RN
6	G. Gomes	M	42	32.827,52	Married	2	...	DF
7	K. Key	F	38	410,50	Divorced	1	...	SE
...	...	...	...	...	...	...	...	...
N	M. Mendes	M	21	3.500,00	Married	4	...	BA

Fig. 4. Sample of a typical table in a database

0,72457	-0,72457	0,20077	-0,27575	-0,72457	...	-0,35355
-1,20760	1,20760	2,17410	-0,36094	-0,72457	...	-0,35355
0,72457	-0,72457	-0,62526	-0,41751	1,20760	...	-0,35355
-1,20760	1,20760	-0,76294	-0,35473	1,20760	...	2,47490
...	...	...	...	...	...	...
0,72457	-0,72457	-0,62526	-0,16805	-0,72457	...	-0,35355

Fig. 5. Data matrix sample obtained after pre-processing stage

4.2 The pruning algorithm

In terms of partSOM architecture, the most suitable algorithm during the initial codification stage in the local units is the self-organizing maps (Kohonen, 2001). In this case, the codebook may contain a few entries with little or no representation in the input set, known as dead neurons. These elements occur with some frequency in clustering processes using the SOM, what has been cited in the literature (Kamimura, 2003). Although inactive neurons can help to maintain the input data topology when they are projected on the map, these units can be discarded without impairment in a process of vector quantization using SOM, because such elements are not referenced in data reassembly stage.

In terms of K-means algorithm, codebook elements with little representation may correspond to outliers or noise in the input data and, eventually, these elements can be discarded from representatives set without great impairment to the maintenance of the statistical distribution of data. So, in both cases, it is possible to include a pruning algorithm in a stage before the transfer of data to the central unit, to reduce the size of the codebook and avoiding moving items that are not used (or are not relevant) in data reconstruction. The procedure for reducing the codebook is performed by a pruning algorithm (Figure 6), which will be detailed below.

The pruning algorithm receives the input dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, the trained codebook $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k\}$, the set of representatives $\mathbf{R}$ and an integer value θ, which corresponds to the representation threshold required for each element. Then, the algorithm searches for elements whose representation is less than or equal to the threshold and eliminates them from the codebook. Finally, the representative choice algorithm is called again to reselect the representatives of each input dataset.

Importantly, the pruning algorithm is an optional step, whose objective is to reduce the amount of data transferred between the remote units and central unit. In the particular case in which the threshold value is zero, only the inactive neurons are eliminated without any change in the outcome.

```
                         Pruning Algorithm

Input:  input dataset (X); original codebook (W);
        representatives set (R); threshold (θ)
Output: modified codebook (W');
        modified representatives set (R')

procedure pruning(X,W,R,θ)
for each $w_j \in W$
        cont = 0
        for each $r_i \in R$
                if (R[i] = j)
                        cont = cont + 1
                endif
        endfor
        if (cont <= θ)
                W' = remove(W,j)
        endif
endfor
R' = choose_representative(X,W')
return(W',R')
```

Fig. 6. The pruning procedure algorithm

4.3 The covariance matrix

The first step in partSOM architecture uses a vector quantization process to effect a compression in the input data and thus reduce the amount of data transferred to the central unit. As in any process of data compression, there are losses associated with vector quantization and, possibly some of the information existing in the input data is discarded during the first stage of the algorithm.

However, as described in Gorgônio (2010), a vector quantization process approximates a probability density function of the input set by a finite set of reference vectors. Thus, if the set of reference vectors chosen to represent the input data is representative enough to capture the statistical distribution of data in the input space, the close relations between the input elements will be maintained. Thus, even if the vector quantization process holds losses, these losses tend to be minimized with proper choice of a good set of representatives.

An alternative to minimize the losses occurring in the process of vector quantization is the use of additional statistical information contained in the original sample, so that the reconstructed data are as similar as possible to the input data. The covariance matrix of a set of data allows extracting the variance and correlation between the samples, and an efficient solution to create random samples containing the same statistical characteristics of the original sample.

Thus, if the covariance matrices of each cluster are drawn in remote units and sent along with the codebooks, so that each centroid can carry information about the variance of the data that it represents, and this information could be used to generate samples with a statistical distribution even more similar to the original dataset, helping to reduce losses associated with the process of vector quantization.

5. Conclusion

This chapter discussed the utilization of cluster ensembles in data clustering and classification tasks. Matters related to existence of geographically distributed databases and mechanisms used for data partitioning were analyzed. It was also presented a wide review on algorithms and strategies used in data mining, mainly in clustering tasks. Following, matters related to distributed data clustering security and privacy were addressed. Eventually, some information fusion techniques used to combine results come from multiple clustering solutions were cited in reviewed works.

The partSOM architecture was presented as a proposal for performing cluster analysis on geographically distributed databases, such as discussed in previous works. However, this study focused specifically on issues related to security and privacy preserving in distributed databases clustering. The main contribution of this work was a bibliographic review about the theme and a discussion about some techniques that can be used in a privacy preserving distributed databases clustering process, including:

i. A data pre-processing stage, which objectives to remove all information that could be used to identify an individual;
ii. A pruning algorithm to reduce the amount of data transferred between the local and central units;
iii. The use of a covariance matrix from each local data unit to reduce losses during the process of vector quantization.

Future research directions will be focused on extent the partSOM architecture, including use of others privacy-preserving strategies. Furthermore, it is necessary to apply and to evaluate this model in real world applications.

6. Acknowledgment

This work was supported by Federal University of Rio Grande do Norte. Flavius Gorgônio (flavius@ufrnet.br) works at Laboratory of Business Applied Computational Intelligence, Department of Exact and Applied Sciences, Caicó, RN, Brazil. José Alfredo F. Costa (alfredo@dee.ufrn.br) works at Laboratory of Adapting Systems, Department of Electrical Engineering, Natal, RN, Brazil.

7. References

Agrawal, R. & Srikant, R. (2000). Privacy-preserving data mining, *ACM SIGMOD Record*, ACM Press, Vol.29, No.2, (June, 2000), pp. 439-450

Agrawal, D. & Aggarwal, C. (2001), On the design and quantification of privacy preserving data mining algorithms, *Proceedings of the Symposium on Principles of Database Systems*, pp. 247-255, Santa Barbara, May, 2001

Arroyave, G.; Lobo, O. & Marín, A. (2002). A parallel implementation of the SOM algorithm for visualizing textual documents in a 2D plane, *Encuentro de Investigación sobre Tecnologías de Información Aplicadas a la Solución de Problemas*, Medellín, Colombia

Berkhin, P. (2006). A survey of clustering data mining techniques, In: *Grouping multidimensional data: recent advances in clustering*, J. Kogan; M. Teboulle & C. Nicholas (Eds.), pp. 25–72, Springer-Verlag, Heidelberg

Bhaduri, K.; Das, K.; Liu, K. & Kargupta, H. (November 2010) Privacy Preserving Distributed Data Mining Bibliography, In: *Distributed Data Mining Bibliography*, 03.11.2010, Available from http://www.cs.umbc.edu/~hillol/DDMBIB

Breiman, L. (1996). Bagging predictors, *Machine Learning*, Vol.24, No.2, pp. 123-140

Calvert, D. & Guan, J. (2005). Distributed artificial neural network architectures, *Proceedings of the 19th Int. Symposium on High Performance Computing Systems and Applications*, pp. 2-10

Chak-Man, L.; Xiao-Feng, Z. & Cheung, W. (2004). Mining local data sources for learning global cluster models, *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, Vol.20, No.24, pp. 748-751, September, 2004

Clifton, C. & Marks, D. (1996). Security and privacy implications of data mining, *Proceedings of the ACM SIGMOD Workshop on Data Mining and Knowledge Discovery*, pp.15-19, Montreal, Canada, June, 1996

Clifton, C.; Kantarcioglu, M.; Vaidya, J.; Lin, X. & Zhu, M. (2002). Tools for privacy preserving distributed data mining, *SIGKDD Explorations*, Vol.4, No.2, (December, 2002), pp. 28-34

DeWitt, D. & Gray, J. (1992). Parallel database systems: the future of high performance database processing. *Communications of the ACM*, Vol.36, No.6, (June, 1992), pp. 85-98

Dimitriadou, E.; Weingessel, A. & Hornik, K. (2001). Voting-merging: an ensemble method for clustering, *Proceedings of the Int. Conf. on Artificial Neural Networks*, LNCS, Vol.2130, pp. 217-224, London: Springer-Verlag

Du, W. & Atallah, M. (2001). Secure multi-party computation problems and their applications: A review and open problems, *New Security Paradigms Workshop*, pp. 11-20, Cloudcroft, New Mexico, September, 2001

Evfimievski, A.; Srikant, R.; Agrawal, R. & Gehrke, J. (2002). Privacy preserving mining of association rules, *Proceedings of the 8th International Conference on Knowledge Discovery in Databases and Data Mining*, Canada, pp. 217-228, July, 2002

Estivill-Castro, V. (2004). Private representative-based clustering for vertically partitioned data, *Proceedings of the Fifth Mexican International Conference in Computer Science*, (September, 2004), pp. 160-167

Faceli, K. (2006). *Um framework para análise de agrupamento baseado na combinação multi-objetivo de algoritmos de agrupamento*, PhD Thesis, Instituto de Ciências Matemáticas e de Computação (ICMC), Universidade de São Paulo, São Paulo, Brazil

Farkas, C. & Jajodia, S. (2002). The inference problem: a survey, *ACM SIGKDD Explorations Newsletters*, Vol.4, No.2, (December, 2002), pp. 6-11

Fern, X. & Lin, W. (2008). Cluster Ensemble Selection, *Proceedings of the 2008 SIAM Int. Conf. on Data Mining*, Atlanta, Georgia, April 24-26, 2008

Forman, G. & Zhang, B. (2000). Distributed data clustering can be efficient and exact. *ACM SIGKDD Explorations Newsletter*, Vol.2, No.2, (December, 2000), pp. 34-38

Freud, Y. & Schapire, R. (1999). A short introduction to boosting, *Journal of Japanese Society for AI*, Vol.14, No.5, pp. 771-780

Frossyniotis, D.; Likas, A. & Stafylopatis, A. (2004). A clustering method based on boosting, *Pattern Recognition Letters*, Vol.25, pp. 641-654

Fung, B.; Wang, K.; Wang, L. & Debbabi, M. (2008). A framework for privacy-preserving cluster analysis, *Proceedings of the IEEE International Conference on Intelligence and Security Informatics*, (June, 2008), pp. 46-51

Georgakis, A.; Li, H. & Gordan, M. (2005). An ensemble of SOM networks for document organization and retrieval, *Proceedings of the Int. Conf. on Adaptive Knowledge Representation and Reasoning*, pp. 141-147, (June, 2005), Espoo, Finland

Georgakis, A. and Li, H. (2006). Content based image retrieval using a bootstrapped SOM network, LNCS, Vol.3972, pp. 595-601, London: Springer-Verlag

Goldreich, O.; Micali, S. & Wigderson, A. (1987). How to play any mental game – a completeness theorem for protocols with honest majority, *Proceedings of the 19th ACM Symposium on the Theory of Computing*, pp. 218-222

Gorgônio, F. & Costa, J. (2008) Parallel self-organizing maps with application in clustering distributed data. *Proceedings of the International Joint Conference on Neural Networks*, Vol.1, (June, 2008), pp. 420, Hong-Kong

Gorgônio, F. & Costa, J. (2010) PartSOM: PartSOM: A Framework for Distributed Data Clustering Using SOM and K-Means. In: Matsopoulos, G. (ed.), *Self-Organizing Maps*, InTech Education and Publishing, Vienna, Austria

Hämäläinen, T. (2002). Parallel implementation of self-organizing maps, In: *Self-Organizing Neural Networks: Recent Advances and Applications*, U. Seiffert & L. Jain (Eds.), Vol.78, pp. 245-278, New York: Springer-Verlag

Haykin, S. (2001). *Redes neurais: princípios e prática*, 2^a ed., Porto Alegre: Bookman

He, Z.; Xu, X. & Deng, S. (2005), Clustering mixed numeric and categorical data: a cluster ensemble approach, Technical report, 07.06.2010, Available from http://aps.arxiv.org/ftp/cs/papers/0509/0509011.pdf

Ho, T. (2000). Complexity of classification problems and comparative advantages of combined classifiers. *Proceedings of the 1st International Workshop on Multiple Classifier Systems*, LNCS, Vol.1857, pp. 97-106, London: Springer-Verlag

Hore, P.; Hall, L. and Goldgof, D. (2006). A cluster ensemble framework for large data sets, *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, Vol.4, pp. 3342-3347, October, 2006

İnan, A.; Saygın, Y.; Savaş, E.; Hintoğlu, A. & Levi, A. (2006). Privacy preserving clustering on horizontally partitioned data, *Proceedings of the 22nd Int. Conf. on Data Engineering Workshops*, pp. 95-103

İnan, A.; Kaya, S.; Saygın, Y.; Savaş, E.; Hintoğlu, A. & Levi, A. (2007). Privacy preserving clustering on horizontally partitioned data, *Data & Knowledge Engineering*, Vol.63, No.3, (December, 2007), pp. 646-666

Jagannathan, G. & Wright, R. (2005). Privacy-preserving distributed k-means clustering over arbitrarily partitioned data, *Proceedings of the 11th ACM SIGKDD Int. Conf. on Knowledge Discovery in Data Mining*, pp. 593-599

Jagannathan, G.; Pillaipakkamnatt, K. & Wright, R. (2006). A new privacy-preserving distributed k-clustering algorithm, *Proceedings of the 2006 SIAM International Conference on Data Mining*, pp. 492-496

Jha, S.; Kruger, L. & McDaniel, P. (2005). Privacy Preserving Clustering, *Proceedings of the 10th European Symposium on Research in Computer Security*, pp. 397-417

Jiang, Y. & Zhou, Z. (2004). SOM ensemble-based image segmentation, *Neural Processing Letters*, Vol.20, No.3, (November, 2004), pp. 171-178

Kantarcioglu, M. & Vaidya, J. (2002). An architecture for privacy-preserving mining of client information, In: *ACM International Conference Proceeding Series*, C. Clifton & V. Estivill-Castro (Eds), Vol.144, pp. 37-42, Australian Computer Society, Darlinghurst

Kantarcioglu, M. (2008). A survey of privacy-preserving methods across horizontally partitioned data, In: *Privacy-preserving data mining*, C. Aggarwal & P. Yu, pp. 313-336, Springer

Kantardzic, M. (2002). *Data mining: concepts, models, methods, and algorithms*, Wiley-IEEE Press

Kapoor, V.; Poncelet, P.; Trousset, F. & Teisseire, M. (2006). Privacy preserving sequential pattern mining in distributed databases, *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, New York, NY, pp. 758-767

Kapoor, V.; Poncelet, P.; Trousset, F. & Teisseire, M. (2007). Préservation de la vie privée: recherche de motifs séquentiels dans des bases de données distribuées. *Revue Ingénierie des Systèmes d'Information*, Vol.12, No.1, (Décembre, 2007), pp. 85-107

Kargupta, H.; Huang, W.; Sivakumar, K. & Johnson, E. (2001). Distributed clustering using collective principal component analysis, *Knowledge and Information Systems*, Vol.3, No.4, pp. 422-448

Kamimura, R. (2003). Competitive learning by information maximization: eliminating dead neurons in competitive learning, *Proceedings of the Joint International Conference ICANN/ICONIP*, LNCS, Vol.2714, pp. 99-106, Springer, Berlin, German

Klusch, M.; Lodi, S. & Moro, G. (2003). Distributed clustering based on sampling local density estimates, *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pp. 485-490

Kohonen, T. (2001). *Self-organizing maps*, 3rd edition, Berlin: Springer

Koikkalainen, P. (1994). Progress with the tree-structured self-organizing map, *Proceedings of the 11th European Conference on Artificial Intelligence*, New York: Wiley

Kuncheva, L. (2003). That elusive diversity in classifier ensembles. *Proceedings of the 1st Iberian. Conference on Pattern Recognition and Image Analysis*, LNCS, Vol.2652, pp. 1126-1138, London: Springer-Verlag

Kuncheva, L. (2004). *Combining pattern classifiers: methods and algorithms*, New Jersey: John Wiley & Sons

Laaksonen, J.; Koskela, M. & Oja, E. (1999). PicSOM: self-organizing maps for content-based image retrieval, *Proceedings of the 1999 Int. Joint Conf. on Neural Networks*, Vol.4, pp. 2470-2473

Laaksonen, J.; Koskela, M.; Laakso, S. & Oja, E. (2000). PicSOM – content-based image retrieval with self-organizing maps, *Pattern Recognition Letters*, Vol.21, No.13-14, (December, 2000), pp. 1199-1207

Laaksonen, J.; Koskela, M. & Oja, E. (2002). PicSOM – Self-organizing image retrieval with MPEG-7 content descriptors, *IEEE Transactions on Neural Networks*, Vol.13, No.4, (July, 2002), pp. 841-853

Leisch, F. (1998). *Ensemble methods for neural clustering and classification*. PhD Thesis, Institut für Statistik, Wahrscheinlichkeitstheorie und Versicherungsmathematik, Teschnische Universität Wien, Austria

Lin, X.; Clifton, C. & Zhu, M. (2005). Privacy-preserving clustering with distributed EM mixture modeling, *Knowledge Information Systems*, Vol.8, No.1, (July, 2005), pp. 68-81

Lindell, Y. & Pinkas, B. (2000). Privacy preserving data mining, *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, pp. 36-54, August, 2000

Luo, H-L.; Xie, X-B. & Li, K-S. (2007). A new method for constructing clustering ensembles, *Proceedings of the Int. Conf. on Wavelet Analysis and Pattern Recognition*, Vol.2, pp.874-878, November 2-4, 2007

Merugu, S. & Ghosh, J. (2003). Privacy-preserving distributed clustering using generative models, *Proceedings of the 3rd IEEE International Conference on Data Mining*, pp. 211-218

Neagoe, V-E. & Ropot, A-D. (2001). Concurrent self-organizing maps for automatic face recognition, *Proceedings of the 29th International Conference of the Romanian Technical Military Academy*, pp. 35-40, Bucharest, Romania, November, 2001

Neagoe, V-E. & Ropot, A-D. (2002). Concurrent Self-organizing maps for pattern classification, *Proceedings of 1st IEEE Int. Conf. on Cognitive Informatics*, pp. 304

Neagoe, V-E. & Ropot, A-D. (2004). Concurrent self-organizing maps – a powerful artificial neural tool for biometric technology, *Proceedings of IEEE World Automation Congress*, Vol.17, Seville

O'Leary, D. (1991). Knowledge discovery as a threat to database security, In: *Knowledge discovery in databases*, G. Piatetsky-Shapiro & W. Frawley (Eds.), pp. 507-516, AAAI/MIT Press, Menlo Park

Oliveira, S. & Zaïane, O. (2003), Privacy preserving clustering by data transformation, *Proceedings of the 18th Brazilian Symposium on Databases*, pp. 304-318, Manaus, Brasil

Oliveira, S. & Zaïane, O. (2004). Privacy preservation when sharing data for clustering, *Proceedings of the Int. Workshop on Secure Data Management in a Connected World*, Vol.1, pp. 67-82, Toronto, Canada

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

Oza, N. & Tumer, K. (2008). Classifier ensembles: select real-world applications, *Information Fusion*, Vol.9, No.1, (January, 2008), pp. 4-20

Piatetsky-Shapiro, G. (1995). Knowledge discovery in personal data vs. privacy: a mini-symposium, *IEEE Expert: Intelligent Systems and Their Applications*, Vol.10, No.2, (April, 1995), pp. 46-47

Roli, F.; Giacinto, G. & Vernazza, G. (2001). Methods for designing multiple classifier systems, *Proceedings of the 2nd Int. Workshop on Multiple Classifier Systems*, LNCS, Vol.2096, pp. 78-87, London: Springer-Verlag

Rosario, G.; Rundensteiner, E.; Brown, D. & Ward, M. (2004). Mapping nominal values to numbers for effective visualization, *Information Visualization*, Vol.3, No.2, (June, 2004) pp. 80-95

Silva, S. (2006). *Comitês de agrupamento aplicados a dados de expressão gênica*, Master Thesis, Universidade Federal do Rio Grande do Norte, Natal, Brazil

Son, J. & Kim, M. (2004). An adaptable vertical partitioning method in distributed systems, *Journal of Systems and Software*, Vol.73, No.3, pp. 551-561

Sousa, M. (1998). *Mineração de dados: uma implementação fortemente acoplada a um sistema gerenciador de banco de dados paralelo*. Master Thesis, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

Strehl, A. (2002). *Relationship-based clustering and cluster ensembles for high-dimensional data mining*, PhD Thesis, The University of Texas at Austin, Austin, Texas, USA

Strehl, A. & Ghosh, J. (2002). Cluster ensembles: a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, Vol.3, (March, 2002), pp. 583-617

Thomazi, S. (2006). *Cluster de turismo: introdução ao estudo de arranjo produtivo local*, Aleph, São Paulo, Brasil

Tumer, K. & Agogino, A. (2008). Ensemble clustering with voting active clusters. *Pattern Recognition Letters*, Vol.29, No.14, (October, 2008), pp. 1947-1953

Vaidya, J. & Clifton, C. (2003). Privacy-preserving k-means clustering over vertically partitioned data. *Proceedings of the Ninth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, New York, pp. 206-215

Vaidya, J.; Clifton, C. & Zhu, Y. (2006). *Privacy preserving data mining*, Springer, New York

Vaidya, J. (2008). A survey of privacy-preserving methods across vertically partitioned data, In: *Privacy-preserving data mining*, C. Aggarwal & P. Yu, pp. 337-358, Springer

Verykios, V.; Bertino, E.; Fovino, I.; Provenza, L.; Saygin, Y. & Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *ACM SIGMOD Records*, Vol.33, No.1, (March, 2004), pp. 50-57

Vin, T.; Seng, M.; Kuan, N. & Haron, F. (2005). A framework for grid-based neural networks, *Proceedings of the 1st Int. Conf. on Distributed Frameworks for Multimedia Applications*, pp. 246-253

Vrusias, B.; Vomvoridis, L. & Gillam, L. (2007). Distributing SOM ensemble training using grid middleware, *Proceedings of the 2007 Int. Joint Conf. on Neural Networks*, pp. 2712-2717

Yang, M-H. & Ahuja, N. (1999). A data partition method for parallel self-organizing map, *Proceedings of the 1999 International Joint Conference on Neural Networks*, Vol.3, pp. 1929-1933

Yao, A. (1986). How to generate and exchange secrets, *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pp. 162-167

Yu, Z.; Zhang, S.; Wong, H-S. & Zhang, J. (2007). Image segmentation based on cluster ensemble, *Proceedings of the 4th Int. Symposium on Neural Networks*, pp. 894-903, China, June, 2007

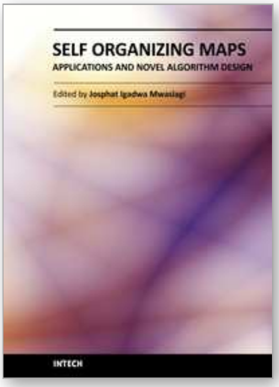
Zhan, J. (2007). Privacy preserving K-medoids clustering, *IEEE International Conference on Systems, Man and Cybernetics*, (October, 2007), pp. 3600-3603

Zhan, J. (2008). Privacy-preserving collaborative data mining, *IEEE Computational Intelligent Magazine*, (May, 2008), pp. 31-41

Zhang, S.; Wu, X. & Zhang, C. (2003). Multi-database mining, *IEEE Computational Intelligence Bulletin*, Vol.2, No.1, (June, 2003), pp. 5-13

Zhao, Y.; Gao, J. & Yang, X. (2005). A survey of neural network ensembles, *Proceedings of the Int. Conf. on Neural Networks and Brain*, Vol.1, pp. 438-442, October, 2005

Zhou, Z-H. & Tang, W. (2006). Clusterer ensemble, *Knowledge-Based Systems*, Vol.19, No.1, (March, 2006), pp. 77-83


SELF ORGANIZING MAPS
APPLICATIONS AND NOVEL ALGORITHM DESIGN
Edited by Josphat Igadwa Mwasiagi
INTECH

Self Organizing Maps - Applications and Novel Algorithm Design
Edited by Dr Josphat Igadwa Mwasiagi

ISBN 978-953-307-546-4
Hard cover, 702 pages
Publisher InTech
Published online 21, January, 2011
Published in print edition January, 2011

Kohonen Self Organizing Maps (SOM) has found application in practical all fields, especially those which tend to handle high dimensional data. SOM can be used for the clustering of genes in the medical field, the study of multi-media and web based contents and in the transportation industry, just to name a few. Apart from the aforementioned areas this book also covers the study of complex data found in meteorological and remotely sensed images acquired using satellite sensing. Data management and envelopment analysis has also been covered. The application of SOM in mechanical and manufacturing engineering forms another important area of this book. The final section of this book, addresses the design and application of novel variants of SOM algorithms.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:

Flavius L. Gorgônio and José Alfredo F. Costa (2011). Privacy-Preserving Clustering on Distributed Databases: A Review and Some Contributions, Self Organizing Maps - Applications and Novel Algorithm Design, Dr Josphat Igadwa Mwasiagi (Ed.), ISBN: 978-953-307-546-4, InTech, Available from: http://www.intechopen.com/books/self-organizing-maps-applications-and-novel-algorithm-design/privacy-preserving-clustering-on-distributed-databases-a-review-and-some-contributions

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821