INTERNATIONAL JOURNAL IN INFORMATION TECHNOLOGY IN GOVERNANCE, EDUCATION AND BUSINESS
Vol. 1, No. 1, 2019
ISSN 2686-0694 (Print)
e-ISSN 2721-0030 (Online)

A novel technique for eliminating parameter initialization in clustering

Menchita F. Dumlao
mfdumlao@pwu.edu.ph
Philippine Women's University, Manila, Philippines

Byung-Joo Oh
bjoh@hnu.kr
Hannam University, Daejeon, South Korea

Abstract

This study revealed the results of simulating the techniques for eliminate cluster initialization in clustering through series of test of different clustering techniques. Clusters of similar database schema from heterogeneous data sources was successfully clustered at an improved accuracy of 93.33%. The accuracy of clusters using self-organizing map neural network (SOM) was improved by implementing the two-step clustering method prior to SOM with an accuracy of 93.33 percent, an increase of 6.33% from the clustering result of SOM. Agglomerative clustering with hierarchical clustering are best combined to come-up with an automatic clustering task, thus, initialization was eliminated.

Keywords: clustering, agglomerative clustering, hierarchical clustering, self-organizing map neural networks

INTRODUCTION

Clustering can be considered as the most important unsupervised learning problem. Thus, every other problem of this kind deal with finding a structure in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters.

The goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding "natural clusters" and describe their unknown properties ("natural" data types), in finding useful and suitable groupings ("useful" data classes) or in finding unusual data objects (outlier detection).

There are some problems encountered in using clustering techniques that compromise its performance in different application. Current clustering techniques do not address all the requirements adequately. In dealing with large number of dimensions, the large number of data items can be problematic because of time complexity. The effectiveness of the method depends on the definition of "distance" (for distance-based clustering). Another problem is if an obvious distance measure doesn't exist, we must "define" it, which is not always easy, especially in multi-dimensional spaces. Hence, the result of the clustering algorithm can be interpreted in different ways.

Clustering algorithms can be classified into exclusive clustering, overlapping clustering, hierarchical clustering, and probabilistic clustering. Exclusive clustering is grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. On the contrary overlapping, clustering uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value. Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted. Finally, the last kind of clustering uses a completely probabilistic approach.

An important component of a clustering algorithm is the distance measure between data points. If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. However, even in this case the Euclidean distance can sometimes be misleading.

K-means is non-hierarchical approaches in forming good clusters to specify a desired number of clusters, say, k, then assign each case (object) to one of k clusters so as to minimize a measure of dispersion within the clusters. A very common measure is the sum of distances or sum of squared Euclidean distances from the mean of each cluster. The problem can be set up as an integer programming problem but because solving integer programs with a large number of variables is time consuming, clusters are often computed using a fast, heuristic method that generally produces good solutions.

K-Means training starts with a single cluster with its center as the mean of the data. This cluster is split into two and the means of the new clusters are iteratively trained. These two clusters are again split, and the process continues until the specified number of clusters is obtained. If the specified number of clusters is not a power of two, then the nearest power of two above the number specified is chosen and then the least important clusters are removed, and the remaining clusters are again iteratively trained to get the final clusters.

When the user specifies a random start, the algorithm generates the k cluster centers randomly and goes ahead by fitting the data points in those clusters. This process is repeated for as many random starts as the user specifies and the best value of start is found. The outputs based on this value are displayed.

The drawback of standard clustering methods is that they ignore measurement errors, or uncertainty, associated with the data. If these errors are available, they can play a significant role in improving the clustering decision. This approach to clustering is called error-based clustering. Error based clustering explicitly incorporates errors associated with data into the clustering algorithm.

In hierarchical clustering the data are not partitioned into a particular cluster in a single step. Instead, a series of partitions takes place, which may run from a single cluster containing all objects to n clusters each containing a single object. Hierarchical clustering is subdivided into agglomerative methods, which proceed by series of fusions of the n objects into groups, and divisive methods, which separate n objects successively into finer groupings. Agglomerative techniques are more commonly used, and this is the method implemented in XLMiner. Hierarchical clustering may be represented by a two-dimensional diagram known as dendrogram which illustrates the fusions or divisions made at each successive stage of analysis.

Clustering techniques such as K-means, hierarchical clustering, and self-organizing map (SOM) neural network has been tested to identify similar schema elements from heterogeneous data sources (Zhao & Ram, 2004). In detecting schema elements using clustering, data is represented using vectors of matrices that was extracted from sources database. These vectors are the features identified for clustering based on a combination of database characteristics such as naming similarity, document similarity, schema specification, data patterns, and usage patterns (Zhao & Ram, 2004). They apply multiple techniques to cross-validate clustering results.

The emergence of linguistic techniques like fuzzy thesaurus (Mirbel, 1997), semantic dictionary, taxonomy (Bright, Hurson, & Pakzad, 1994; Song, Johannesson, & Bubenko, 1996), conceptual graph, case grammar (Ambrosio, Metais, & Meunier, 1997) and speech act theory (Johannesson, 1997) contributed in the development of schema integration research most significantly the determination of the degree of similarity between schema elements, based on the names of the elements. An assumption of these approaches is that schema elements are named using reliable terms, which describe the meanings of the elements appropriately. However, schema elements in legacy systems are poorly named, using ad-hoc acronyms and phrases. Heuristic formulae have been designed to compute the degree of similarity between schema elements (Hayne et al., 1990). These formulae often have been derived based on experiments and experiences from particular integration projects, giving rise\to concern about the generalization of the heuristic formulae over different settings.

Overview of Cluster Analysis

Clustering is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups. Cluster analysis techniques group objects from some problem domain, into unknown groups called clusters, such that objects within the same clusters are similar to each other, while objects across clusters are dissimilar to each other. The objects to be clustered are represented as vectors of features, or variables. Clustering is useful in several exploratory pattern-analysis, grouping, decision-making, and machine-learning situations, including data mining, document retrieval, image segmentation, and pattern classification. However, in many such problems, there is little prior information (e.g., statistical models) available about the data, and the decision-maker must make as few assumptions about the data as possible.

Typical pattern clustering activity involves pattern representation (optionally including feature extraction and/or selection), definition of a pattern proximity measures appropriate to the data domain, clustering or grouping, data abstraction, and assessment of output. Figure 1 depicts a typical sequencing of the first three of these steps, including a feedback path where the grouping process output could affect subsequent feature extraction and similarity computations (Jain et al., 1999).

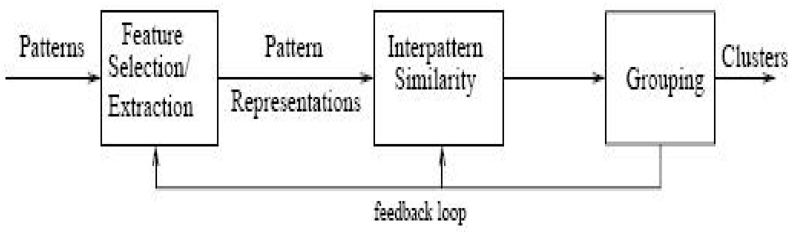

Figure 1. Stages in Clustering

Pattern representation refers to the number of classes, the number of available patterns, and the number, type, and scale of the features available to the clustering algorithm. Some of this information may not be controllable by the practitioner. Feature selection is the process of identifying the most effective subset of the original features to use in clustering. Feature extraction is the use of one or more transformations of the input features to produce new salient features. Either or both of these techniques can be used to obtain an appropriate set of features to use in clustering.

Pattern proximity is usually measured by a distance function defined on pairs of patterns. A variety of distance measures are in use in the various communities (Anderberg, 1973; E. Diday & Simon, 1976; Jain & Dudes, 1998). A simple distance measure like Euclidean distance can often be used to reflect dissimilarity between two patterns, whereas other similarity measures can be used to characterize the conceptual similarity between patterns.

The grouping step can be performed in a number of ways. The output clustering can be hard (a partition of the data into groups) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters). Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partitional clustering algorithms identify the partition that optimizes (usually locally) a clustering criterion. Additional techniques for the grouping operation include probabilistic (Dubes, 1993) and graph-theoretic (Zhan, 1971) clustering methods.

Data abstraction is the process of extracting a simple and compact representation of a data set. Here, simplicity is either from the perspective of automatic analysis (so that a machine can perform further processing efficiently) or it is human-oriented (so that the representation obtained is easy to comprehend and intuitively appealing). In the clustering context, a typical data abstraction is a compact description of each cluster, usually in terms of cluster prototypes or representative patterns such as the centroid (Diday & Simon, 1976).

Statement of the Problem

The steps mentioned above determine the data analysis techniques that can be applied using cluster analysis. However, the five steps mentioned above does not include any technique in evaluation of output of clustering. We pose on two important questions: How is the output of a clustering algorithm evaluated? What characterizes a ‘good’ clustering result and a ‘poor’ one?

All clustering algorithms will, when presented with data, produce a cluster, that is regardless of whether the data contain clusters or not. If the data does contain clusters, some clustering algorithms may obtain ‘better’ clusters than others. The assessment of a clustering procedure’s output, then, has several facets. One is actually an assessment of the data domain rather than the clustering algorithm itself— data which do not contain clusters should not be processed by a clustering algorithm. The study of cluster tendency, wherein the input data are examined to see if there is any merit to a cluster analysis prior to one being performed, is a relatively inactive research area, and will not be considered further in this study.

Cluster validity analysis, by contrast, is the assessment of a clustering procedure’s output. Often this analysis uses a specific criterion of optimality; however, these criteria are usually arrived at subjectively. Hence, little in the way of ‘gold standards’ exist in clustering except in well-prescribed sub domains. Validity assessments are objective (Dudes, 1993) and are performed to determine whether the output is meaningful. A clustering structure is valid if it cannot reasonably have occurred by chance or as an artifact of a clustering algorithm. When statistical approaches to clustering are used, validation is accomplished by carefully applying statistical methods and testing hypotheses. There are three types of validation studies. An external assessment of validity compares the recovered structure to an a priori structure. An internal examination of validity tries to determine if the structure is intrinsically appropriate for the data. A relative test compares two structures and measures their relative merit.

Clustering using Self-Organizing Maps Neural Network

Self-organizing maps as unsupervised learning method that employ competitive learning algorithm (CLA). Given a set of cluster representatives pairs, w j, j=1…m (m is the number of clusters), the idea behind CLA is to move each of these representatives to the regions of the vector space that are dense in vectors of x.

The term “competitive” arises from the fact that when an input pattern x is presented, each wj equation always compete with each other. The winner is the wj that lies closer to x, which is then updated so as to move toward x, while losers either remain stationary or are used x slowly. For a detailed exposition of the Generalized CL Section (GCLS) see (Theodoridis & Koustroumbas, 2006).

An important component of GCLS is the update of the cluster representatives, following distance evaluation between input pattern and representative wj, in SOM, this is expanded as:

$$w_k(t) = \frac{(t-1)+n(t)(x-w_k(t-1))}{w_k(t-1)}, \quad \text{if, } w_k(\mathrm{t}) \in \ Q_1(t) \qquad (1)$$

Note in the above equation that, not only the single w_i close to x is updated but rather whole neighborhood $Q_j(t)$ but the 2nd term in the last line denotes that such an update is also dependent on the distance between x and $w_k \in Q_j(t)$ aside from the learning rate (t) .

Artificial neural networks (ANNs) are motivated by biological neural networks (Hertz et al. 1991 in Jain, 1999). ANNs have been used extensively over the past three decades for both classification and clustering (Jain and Mao 1994 in Jain, 1991). Some of the features of the ANNs that are important in pattern clustering are:

(i) ANNs process numerical vectors and so require patterns to be represented using quantitative features only.
(ii) ANNs are inherently parallel and distributed processing architectures.
(iii) ANNs may learn their interconnection weights adaptively (Jain & Mao, 1996).
(iv) More specifically, they can act as pattern normalizers and feature selectors by appropriate selection of weights. Competitive (or winner–take– all) neural networks (Jain & Mao, 1996) are often used to cluster input data.

In competitive learning, similar patterns network is represented by a single unit (neuron). This grouping is done automatically based on data correlations. Well-known examples of ANNs used for clustering include Kohonen’s learning vector quantization (LVQ) and self-organizing map (SOM) (Kohonen, 1991) and adaptive resonance theory models (Bandfield & Raftery, 1993). The architectures of these ANNs are simple: they are single layered. Patterns are presented at the input and are associated with the output nodes. The weights between the input nodes and the output nodes are iteratively changed (this is called learning) until a termination criterion is satisfied.

Competitive learning has been found to exist in biological neural networks. However, the learning or weight update procedures are quite similar to those in some classical clustering approaches. The SOM gives an intuitively appealing two-dimensional map of the multidimensional data set, and it has been successfully used for vector quantization and speech recognition (Kohonen, 1991). However, like its sequential counterpart, the SOM generates a suboptimal partition if the initial weights are not chosen properly. Further, its convergence is controlled by various parameters such as the learning rate and a neighborhood of the winning node in which learning takes place. It is possible that a particular input pattern can fire different output units at different iterations; this brings up the *stability* issue of learning systems.

The system is said to be stable if no pattern in the training data changes its category after a finite number of learning iterations. This problem is closely associated with the problem of *plasticity*, which is the ability of the algorithm to adapt to new data. For stability, the learning rate should be decreased to zero as iterations progress and this affects the plasticity.

In this experiment, we use cluster analysis technique to determine the correspondence between schemas. The schema is composed of the attributes of the database which describes how data is defined in the database. The characteristics of the attributes (i.e., fields) were identified to determine the features needed for the clustering technique. There are two main features used in this study, naming similarity and data pattern. Naming similarity is one of the features used in clustering technique. Edit distance function (Dumlao, Oh, & Agustin, 2009) is used to get the value for naming similarity. Two attributes are compared using edit distance to be able to produce a value range from 0...1.

Data pattern is another feature extracted from database. The summary of statistics of each attribute in the database are computed using the functions: average, count of missing values, count of distinct values, average of length of values, standard deviation of length of values, maximum, statistics on percentage of digits in the attribute values, statistics on the percentage of letters in the attribute values. The accumulated values from naming similarity and data patterns were combined and then preprocessed using Principal Component Analysis (PCA). The results of summary of statistics are normalized into range 0 to 1 before it is inputted to SOM.

Clustering using SOM was tested in SPSS Clementine. A total of 45 by 14 matrixes of features were accumulated after PCA. An expert training method was used to specify the topology of kohonen net and the learning rates used for training. The topology of a Kohonen network in Clementine is always a 2-dimensional rectangular grid. The map was set to a 6 by 6 grid, and the learning rate to exponential.

Kohonen net training is split into two phases. Phase 1 is a rough estimation phase, used to capture the gross patterns in the data. Phase 2 is a tuning phase, used to adjust the map to model the finer features of the data. For each phase, there are three parameters. We set the starting size (radius) of the neighborhood into 2, to determine the number of nearby units that gets updated along with the winning unit during training. During phase 1, the neighborhood size starts at phase 1 neighborhood and decreases to Phase 2 Neighborhood +1. During Phase 2, neighborhood size starts at phase 2 neighborhood and decreases to 1.0. Phase 1 Neighborhood should be larger than phase 2 neighborhood.

Clustering was tested using SPSS for customer database and used the combination of parameters that includes: (1) Phase 1: neighborhood =15, initial eta=0.3, cycles=50; (2) Phase 2: neighborhood=1, eta=0.1, cycles=150.

During phase1, eta starts at phase 1 initial eta and decreases to phase 2 initial eta. During phase 2, eta starts at phase 2 initial eta and decreases to 0. Phase 1 initial eta should be larger that phase 2 initial eta.

During training, each grid square competes with all the others to ‘win’ each record. “Strong” nodes will win more records and “weak” nodes may win no records at all. As the grid squares competes, the training regime ‘settles’ the network onto a stable classification, capturing as much of the information in the training records as possible.

Another database was used, the E-catalog database, to test the robustness of SOM. The data set in (Zhao & Ram, 2004) was used. Data pre-processing was done in the e-catalog database, normalize the features in range 0.1 and then perform PCA. The 30 rows by 44 columns of features, was reduced to 30 by 10 columns after PCA. Clustering using SPSS was used with the combination of parameters that includes: (1) Phase 1: neighborhood =2, initial eta=0.3, cycles=20; (2) Phase 2: neighborhood=1, eta=0.1, cycles=150.

RESEARCH DESIGN & METHODS

Clustering Technique using Two Step Clustering + SOM

Most of existing cluster methods (except the EM method) needs a distance measure. Different distance measures may lead to different cluster results. Some distance measures accept only continuous variables like Euclidean distance, and some only categorical variables, such as the simple matching dissimilarity measure used in the k-modes. Weights can be chosen arbitrarily, but improper weight may bias in the treatment of different variable types (Bandfield & Raftery, 1993) that introduced a model-based distance measure for data with continuous attributes. They derived this measure from a Gaussian mixture model, equivalent to the decrease in log-likelihood resulting from merging two clusters. Melia and Heckerman (1998) applied this probabilistic concept and derived another distance measure for data with categorical attributes only. The tow-step cluster component extends this model-based distance measure to situations that include both continuous and categorical variables.

None of the cluster methods directly address the issue of determining the number of clusters because the means of determining the number of clusters is difficult and it is considered a separate issue. Various strategies have been applied to determine the number of clusters. Some techniques cluster the data into a series of numbers of clusters (such as two clusters, three clusters, etc.) and calculate certain criterion statistics for each of them. The one with the best statistic is the “winner”. Fraley and Raftery (1998) proposed using the Bayesian information criterion (BIC) as the criterion statistic for the EM clustering method, (Bandfield & Raftery,1993) suggested using the approximate weight of evidence (AWE) as the criterion statistic for their model-based hierarchical clustering.

Two-step clustering is concerned with two main steps: pre-clustering and agglomerative hierarchical clustering methods. In pre-clustering, a sequential clustering approach (Theodoridis & Koustroumbas, 2006) was used. It scans the records one by one and decides if the current record should merge with the previously formed clusters or start a new cluster based on the distance criterion. It is implemented by constructing a modified cluster feature (CF) (Zhang, Ramakrishnon, & Livny, 1996). The CF-tree consists of levels of nodes, and each node contains a number of entries. A leaf entry (an entry in the leaf node) represents a sub-cluster. The non-leaf nodes and their entries guide a new record into a correct leaf node quickly.

A CF with the entry’s number of records, the mean and variance of each continuous variable, plus the counts for each category of each categorical variable characterize each entry. Each successive record, starting from the root node, is recursively guided by the closest entry in the node to find the closest child node, and then descends along the CF-tree. Upon reaching a leaf node, it finds the closest leaf entry in the leaf node. If the record is within a threshold distance of the closest leaf entry, the leaf entry absorbs it and updates the CF.

Otherwise it starts its own leaf entry in the leaf node. If the leaf node has no space to create a new leaf entry, the leaf node splits in two. The entries in the original leaf node divide into two groups using the farthest pair as seeds, redistributing the remaining entries based on the closest criteria.

If a CF-tree grows beyond the maximum number of levels, the CF-tree rebuilds the existing CF-tree by increasing the threshold distance criterion. The rebuilt CF-tree is smaller, so it has space for new input records. This process continues through a complete data pass.

The CF used in this pre-cluster is different from the one used in BIRCH, which only handles continuous variables. BIRCH’s CF comprises the entry’s number of records, mean and variance of each continuous variable. The SPSS CF extends BIRCH’s CF by including the counts for each category of each categorical variable. The entry’s CF collectively represents all records falling in the same entry. When you add a new record to an entry, you can compute the new CF from the old CF without knowing the individual records in the entry. These properties make it possible to maintain only the entry CFs, rather than the sets of individual records. Therefore, the CF-tree is much smaller and more likely to be stored in the main memory.

The algorithm for two step clustering is illustrated in figure 2.

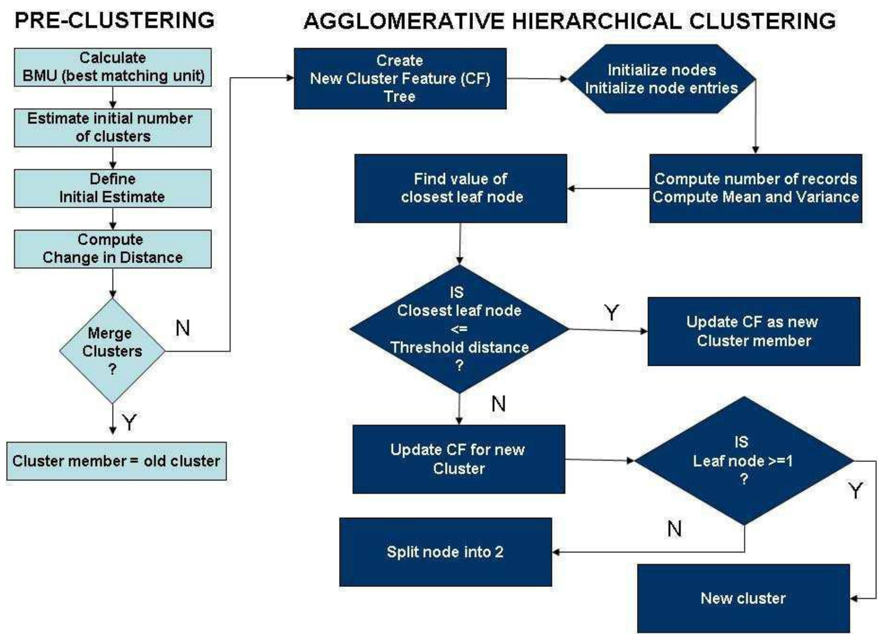
PRE-CLUSTERING
AGGLOMERATIVE HIERARCHICAL CLUSTERING
Calculate BMU (best matching unit)
Estimate initial number of clusters
Define Initial Estimate
Compute Change in Distance
Merge Clusters ?
N
Y
Cluster member = old cluster
Create New Cluster Feature (CF) Tree
Initialize nodes Initialize node entries
Compute number of records Compute Mean and Variance
Find value of closest leaf node
IS Closest leaf node <= Threshold distance ?
Y
Update CF as new Cluster member
N
Update CF for new Cluster
IS Leaf node >=1 ?
Y
N
Split node into 2
New cluster

Figure 2. Two-Step Clustering Algorithm

In agglomerative hierarchical clustering, the cluster step takes sub-clusters resulting from the first step as input and then groups them into the desired number of clusters. Since the number of sub-clusters is much less than the number of original records, the traditional clustering method works effectively. Agglomerative hierarchical clustering method is used in two-step because it works well with the auto-cluster the component structure allows the deployment of future methods easily as they become available. How do we know how many clusters are there? The answer depends on your dataset.

Hierarchical clustering characteristically produces a sequence of partitions at one run: 1, 2, 3 … clusters. The k-means and EM method would need to run multiple times (one for each specified number of clusters) in order to generate the sequence. To determine the number of clusters automatically, SPSS developed a two-step procedure that works well with the hierarchical clustering method. The first step calculates BIC for each number of clusters within a specified range and uses it to find the initial estimate for the number of clusters. The second step refines the initial estimate by finding the greatest change in distance between the two closest clusters in each hierarchical clustering stage.

A new distance measure is used in both the pre-cluster and cluster steps. In order to handle both continuous and categorical variables, define the distance between two clusters as the corresponding decrease in log-likelihood by combining them into one cluster. In calculating log-likelihood, assume normal distributions for continuous variables and multinomial distributions for categorical variables. We assume that the variables are independent of each other, as well as the records.

After the successful testing of two-step clustering that eliminates cluster initialization, self-organizing map (SOM) neural network was used to cluster similar schema from 3 heterogeneous data sources. Figure 3 shows the final algorithm for this technique.

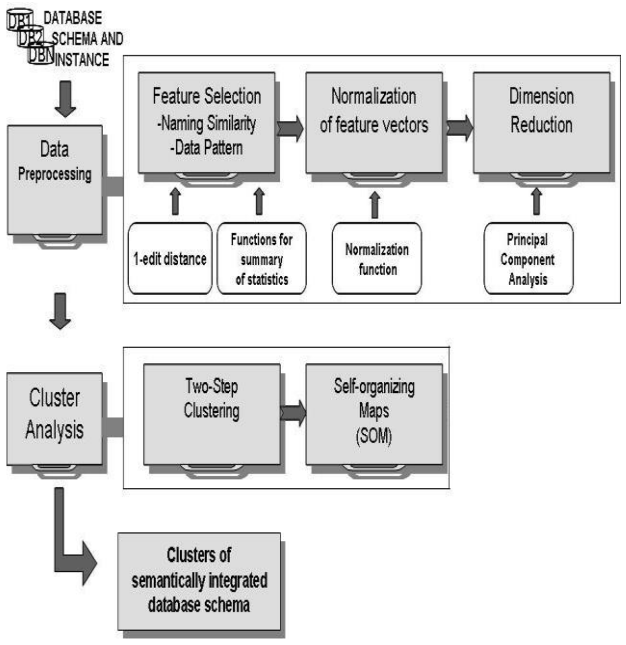
DB1
DB2
DBN
DATABASE SCHEMA AND INSTANCE
Data Preprocessing
Feature Selection -Naming Similarity -Data Pattern
Normalization of feature vectors
Dimension Reduction
1-edit distance
Functions for summary of statistics
Normalization function
Principal Component Analysis
Cluster Analysis
Two-Step Clustering
Self-organizing Maps (SOM)
Clusters of semantically integrated database schema

Figure 3. Algorithm for Two-Step Clustering with SOM

RESULTS

Two-Step Clustering with SOM for Customer Database

To improve the clustering result of SOM, this study implemented the proposed model named "two-step clustering with SOM". This technique explored the potential of cluster analysis mechanism in detecting similarity among attributes. The model is composed of two main stages, namely two-step clustering and SOM. The input vectors are clustered into two main clusters using two-step clustering algorithm. The output of the two-step model was used with SOM.

INTERNATIONAL JOURNAL IN INFORMATION TECHNOLOGY IN GOVERNANCE, EDUCATION AND BUSINESS
Vol. 1, No. 1, 2019
ISSN 2686-0694 (Print)
e-ISSN 2721-0030 (Online)

In a two-step clustering method, the first step makes a single pass through the data, during which it compresses the raw input data into a manageable set of sub-clusters. The second step uses a hierarchical clustering method to progressively merge the sub-clusters into larger and larger clusters, without requiring another pass through the data. Hierarchical clustering has the advantage of not requiring the number of clusters to be selected ahead of time. Many hierarchical clustering methods start with individual records as starting clusters and merge them recursively to produce ever larger clusters. Though such approaches often break down with large amounts of data, Two- step's initial pre-clustering makes hierarchical clustering fast even for large data sets.

This method can be used to cluster the data set into distinct groups without prior knowledge of the number on clusters. As with kohonen and k-means, two-step clustering model do not use a target field. Instead of trying to predict an outcome, two-step cluster tries to uncover patterns in the set of input fields. Records are grouped so that records within a group or cluster tend to be similar to each other, but records in different groups are dissimilar.

Two-step clustering uses one or more fields. Fields with direction out, both, or none are ignored, and it does not handle missing values. Records with blanks for any of the input fields will be ignored when building the model. It can handle mixed field types and large data sets efficiently. It also has the ability to test several cluster solutions and choose the best, so the number of clusters does not need to be set. Cluster can be set to automatically exclude outliers, or extremely unusual cases that can contaminate results.

The model name is automatically based on the target or ID field (or model type in cases where no such field is specified) or specifies a custom. By default, two-step will standardize all numeric input fields to the same scale, with a mean of 0 and a variance of 1. To retain the original scaling for numeric fields, deselect this option. Symbolic fields are not affected. When scoring data with a two-step model that uses outlier handling, new cases that are more than a certain threshold distance (based on the log-likelihood) from the nearest substantive cluster are considered outliers and are assigned to the "noise" cluster.

Two-step clustering can very rapidly analyze a large number of cluster solutions to choose the optimal number of clusters for the training data. The maximum and the minimum number of clusters can be set.

Figure 4 is a scatter plot that illustrates the clustering result of proposed two-step clustering + SOM for customer database. The points of similar attributes were found to be closer to each other which make one cluster enclosed in solid line.

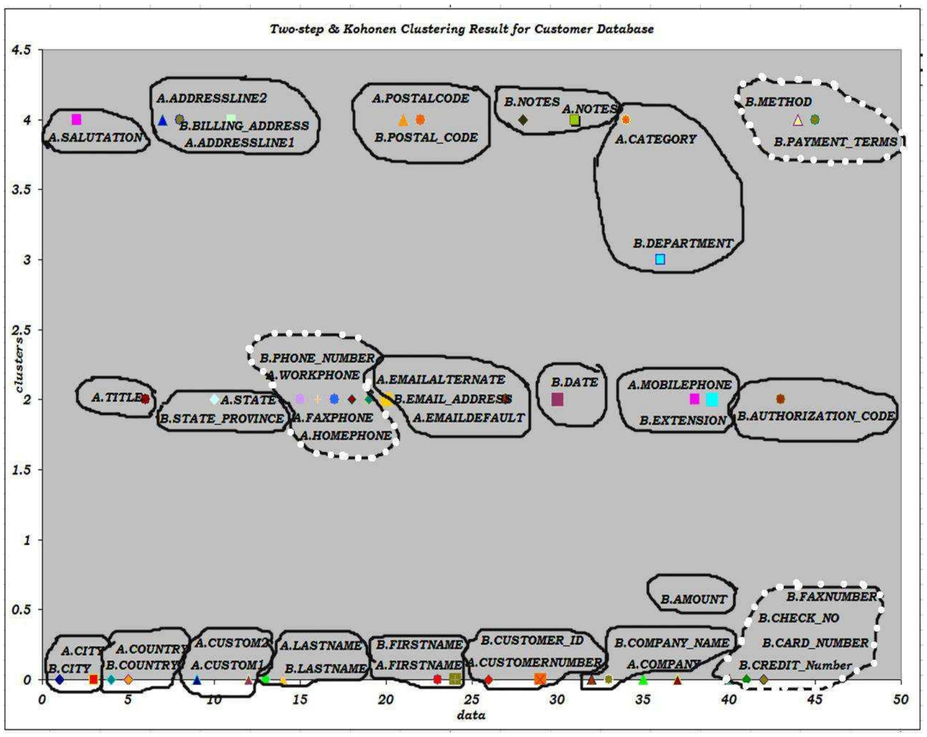

Figure 4. Result of two-step Clustering with SOM for Customer Database

However, three of the clusters were considered an incorrect cluster because the cluster members (attributes) found in clusters does not represent similar records of values in the database. B.METHOD must cluster with A.CUSTOM1 and A.CUSTOM2; A.FAXPHONE must cluster with B.FAXPHONE. The clustering accuracy of two-step clustering + SOM for customer database resulted into 86.4% with 19 correct clusters and 3 incorrect clusters. This is higher that than with the result of SOM clustering. The clustering performance of two-step + SOM improved significantly by 16.4% over SOM.

For the cleansed customer database, 93.33% clustering accuracy was garnered over 87% of SOM clustering, increasing at 6.33% over SOM. The scatter plot in figure 4 illustrates the clustering result for reduced db using two-step clustering + SOM.

However, three of the clusters were considered an incorrect cluster because the cluster members (attributes) found in clusters does not represent similar records of values in the database.

B.METHOD must cluster with A.CUSTOM1 and A.CUSTOM2; A.FAXPHONE must cluster with B.FAXPHONE.

The clustering accuracy of two-step clustering + SOM for customer database resulted into 86.4% with 19 correct clusters and 3 incorrect clusters. This is higher that than with the result of SOM clustering. The clustering performance of two-step + SOM improved significantly by 16.4% over SOM.

For the cleansed customer database, 93.33% clustering accuracy was garnered over 87% of SOM clustering, increasing at 6.33% over SOM. The scatter plot in figure 5 illustrates the clustering result for reduced db using two-step clustering + SOM.

There is only one incorrect cluster which gives an improved impression of clustering results over all techniques that was used. The cluster in broken lines is considered incorrect because A.HOME_PHONE must cluster with B.PHONE_NUMBER and A.WORKPHONE. A.FAXPHONE must cluster with B.FAXNUMBER.

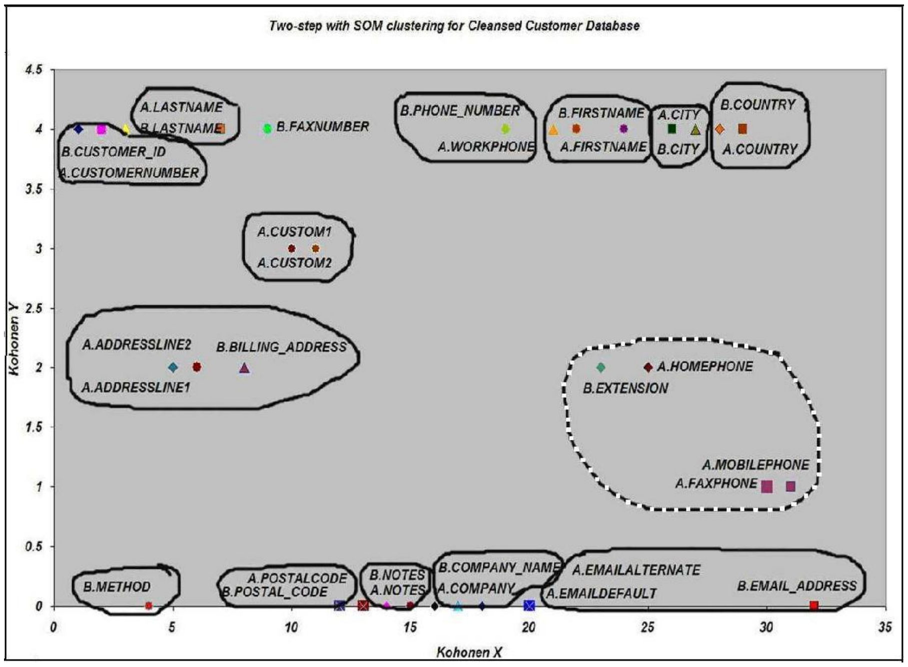

Figure 5. Result of Two-step Clustering with SOM for Cleansed Customer Database

CONCLUSION

This study proposed a novel algorithm for clustering database schema that can be used for semantic integration frameworks which can discover the semantics of data coming from different sources and represented in different platform by focusing on the meaning and nature of data. Experiments showed that cluster analysis is capable of determining the semantics of data, by grouping similar attributes into one cluster. The efficiency of cluster analysis using SOM depends on two main aspects, namely, quality of data inputs and parameters. In dealing with the quality of data inputs, the nature of real-world data that is noisy and complicated is a crucial factor that must be considered first. Most of the time, data should be cleaned by using appropriate data preprocessing techniques and must be chosen carefully to be able to meet the suitable techniques relative to type of data and data analysis needed.

For semantic integration of heterogeneous databases, the two-step + SOM approach was proven to be an effective tool in determining the similarity of attributes. It performs better than the typical SOM algorithm most especially in addressing the complexity of data used as inputs.

Recommendations

Data integration is a vast area of research spanning from business, academic functions, military tactics and other sectors that uses data as a critical tool in communications and operations. Semantic integration will always be a part of any data integration process as it will be a strong foundation that must be given further analysis and development. The future of semantic integration would be more progressive if data preprocessing will be improved. An automatic data preprocessing technique for database schema and instances must be developed to shorten the lifespan of any data integration research. All of the databases that exist in most organizations are not perfect so data cleaning must be a standard procedure of data integration.

Feature selection technique is another area that must be developed because SOM could reach its optimum performance if the features used in the network are appropriate to the purpose that it served. Researches should collect a useful and relevant feature that perfectly represents the structure of data. The greater number of features, the more likely SOM perform accurately.

Further improvement of the SOM algorithm is recommended because SOM and other neural network techniques are one of the most useful techniques applicable for data integration because of its ability to recognized pattern among data.

REFERENCES

Ambrosio, A., Metais, E., & Meunier, J. (1997). The linquistic level: Contribultion for conceptual design, view integration, reuse and documentation. *Data & Knowledge, vol. 21*, pp. 111-129.

Anderberg, M. (1973). *Cluster analysis for Applications.* New York: Academic Press, Inc.

Banfield, J., & Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics, vol. 49*, pp. 803–821.

Bright, M., Hurson, A., & Pakzad, S. (1994). *Automated resolution of semantic heterogeneity in multi databases.* ACM.

Diday, E., & Simon, J. (1976). Clustering Analysis in Digital Pattern Recognition. pp. 47-94.

Dubes, R. (1993). Cluster analysis and related issues. In C. H. Chen, L. F. Pau, & P. S. Wang, *In Handbook of Pattern Recognition & Computer Vision*, pp. 3–32. River Edge, NJ: Eds. World Scientific Publishing Co., Inc.

Dumlao, M., Oh, B., & Agustin, O. (2008). Data Preprocessing Techniques for Data Integration. *Proceeding: 3rd Asia Pacific International Conference on Information Science and Technology.*

Fraley, D., & Raftery, A. (1998). How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Jounal, vol. 4*, pp. 578-588.

Hayne, S., & Ram, S. (1990). Multi-user view integration system (MUVIS): An expert system for view integration. *6th International Conference on Data Engineering*, pp. 402-410.

Jain, A., & Dubes, R. (1998). *Algorithms for clustering data.* Upper Saddle River, New Jersey, 1998: Prentice-Hall Advanced Reference Series. Prentice-Hall, Inc.

Jain, A., & Mao, J. (1996). Artificial neural networks: A tutorial. vol. *29*, pp. 31-44.

Jain, A., Murty, M., & Flynn, P. (1999). Data Clustering: A Review. *ACM Computing Surveys, vol. 31*.

Johannesson, P. (1997). Supporting schema integration by linguistic Instruments. *Data & Knowledge Engineering, vol. 21*, pp. 165-182.

Kohonen, T. (1989). Self-Organization and Associative Memory. In *Springer information sciences series*, 3rd ed.. New York: Springer-Verlag.

Koustroumbas, S. (1991). Pattern Recognition. In H. e. al., *Pattern Recognition*,3rd edition ed., pp.102. USA: Elsevier.

Melia, M., & Heckerman, D. (1998). *An experimental comparison of several clustering and initialization methods.* Microsoft Research Technical Report.

Mirbel, I. (1997). Semantic integration of conceptual schemas. *Data & Knowledge Engineering, vol. 21*, pp. 183-195.

Song, W., Johannesson, P., & Bubenko, J. (1996). Semantic similarity relations and computation in schema integration. *Data & Knowledge Engineering, vol. 19(1)*, pp. 65-97.

Zhan, C. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transaction Computing*, pp. 68-86.

Zhang, T., Ramakrishnon, R., & Livny, M. (1996). BIRCH: An efficient data clustering method for very large databases. *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 103– 114. Canada.

Zhao, H., & Ram, S. (2004). Clustering Schema Elements for Semantic Integration of Heterogeneous Data Sources. *Journal of Database Management, vol. 15*, pp. 88-106.