Volume 2, No. 3, May-June 2011
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info

Integration of *K*_Means and Decision Tree for Knowledge Extraction from a Database

A. E. ELAlfy
Computer Science Department
Mansoura University,
Egypt

A. F. ELGamal
Computer Science Department
Mansoura University,
Egypt

D.L. Elshowakh*
Computer Science Department
Mansoura University,
Egypt
dalia_lotfy@mans.edu.eg

Abstract: Data mining is the process of discovering previously unknown and potentially interesting patterns in databases. Though most knowledge discovery methods have been developed for supervised data, the task of finding knowledge from unsupervised data often arises in real-world problems. In addition, techniques for unsupervised knowledge discovery are essentially different and still much less developed than those for supervised discovery. This paper introduces a novel framework for extracting a set of comprehensible rules from unsupervised database. The proposed framework depends on three techniques namely; clustering technique, fuzzification technique, and inductive learning technique. Clustering technique uses a k-means for clustering unsupervised database. Consequently the input database is converted into supervised database. Fuzzification technique transforms the continuous attributes of database into linguistic terms. This transformation leads to reduction of search space. Decision tree used as a inductive learning algorithm for extracting a set of accurate rules from supervised database.

Keywords: Unsupervised Database; K_Means; Decision Tree; Clustering Technique; Rule Extraction.

I. INTRODUCTION

The amount of data stored in databases continues to grow fast. Intuitively, this large amount of stored data contains valuable hidden knowledge, which could be used to improve the decision-making process of an organization. For instance, data about previous sales might contain interesting relationships between products and customers. The discovery of such relationships can be very useful to increase the sales of a company. However, the number of human data analysts grows at a much smaller rate than the amount of stored data. Thus, there is a clear need for automatic methods for extracting knowledge from data. This need has led to the emergence of a field called data mining and knowledge discovery [1]. This is an interdisciplinary field, using methods of several research areas (specially machine learning and statistics) to extract high-level knowledge from real-world data sets. The application of a data mining algorithm to a data set can be considered the core step of a broader process, often called the knowledge discovery process [2]. The knowledge discovery process includes several other steps. For the sake of simplicity, these steps can be roughly categorized into data pre-processing and discovered-knowledge post-processing. The data pre-processing may be included the Data Integration step, Data Cleaning step, Discretization step, and Attribute selection step [3]. Discovered-knowledge post-processing usually aims at improving the comprehensibility and/or the interestingness of the knowledge to be shown to the user. There are two main motivations for such post-processing. First, when the discovered rule set is large, we often want to simplify it - i.e., to remove some rules and/or rule conditions - in order to improve knowledge comprehensibility for the user. Second, we often want to extract a subset of interesting rules, among all discovered ones [4-8]. Clustering is a process of grouping data objects into disjointed clusters so that the data in each cluster are similar, yet different to the others. Clustering techniques are applied in many application areas such as vector quantization (VQ) [9–12], pattern recognition [13], knowledge discovery [14], speaker recognition [15], fault detection [16], and web/data mining [17]. Among clustering formulations that minimize an object function, *k*-means clustering is perhaps the most widely used and studied [18]. *K-Means* is attractive in practice, because it is simple and it is generally very fast. It partitions the input dataset into *k* clusters. Each cluster is represented by an adaptively-changing centroid (also called cluster centre), starting from some initial values named seed-points. *K-Means* computes the squared distances between the inputs (also called input data points) and centroids, and assigns inputs to the nearest centroid. The k-means clustering algorithm performs iteratively the partition step and new cluster center generation step until convergence. An iterative process with extensive computations is usually required to generate a set of cluster representatives. Inductive learning, as an active research area of machine learning, explores algorithms that reason from externally supplied examples to produce general theories, which make predictions about examples. The externally supplied examples used for generating theories are usually referred to as training examples. As created theories should be more general than the training examples from which they are derived generalization is involved during reasoning. Many inductive learning algorithms operate by analyzing examples to find intergroup similarities and inter-group differences, so inductive learning is sometimes also called similarity based learning [19-22]. If training examples are given with known labels such as the diagnoses of an illness for patients, the inductive learning is called supervised learning [23-24]. Supervised learning can solve two types of problem: classification problems in which labels are categorical [25] and regression problems in which labels are continuous [26].

Such a problem of supervised learning can be viewed as a search problem [23] involving a large hypothesis space that is the space consisting of all possible theories (hypotheses) under consideration. The search aims to find the best theory with respect to the training examples as well as some prior knowledge and expectations. Decision trees [25], and production rules [27] are two commonly used theory description languages in supervised learning. An important advantage of decision trees and production rules is that they are relatively easy for humans to understand. Actually, they have been used by human experts to express and process their knowledge in a wide variety of domains. In addition, compared with other theory description languages, they perform reasonably well in many domains [28]. A production rule consists of two parts: antecedent and consequent. The antecedent ("IF part") contains a conjunction of conditions on predicting attribute values. The consequent ("THEN part") contains a predicted value for the goal attribute (class).

This paper introduces a novel framework for rule extraction from a database depending on two main algorithms. The first algorithm is the clustering technique, which clusters the input instances of database. Consequently, it uses the clustering technique as a learning tool to prepare the input database for the second algorithm. The second algorithm is a inductive learning technique, which induces a set of accurate If-Then rules from the clustering database. These rules are refined and may reduce the dimensionality of the input attributes.

The rest of the paper is organized as follows. The general framework is introduced in section 2. The *k*-means algorithm is described in section 3 and followed by a fuzzification technique which is presented in section 4. A decision tree algorithm is proposed in section 5. Section 6 gives some experimental results. Finally, the conclusion is presented in section 7.

II. GENERAL FRAMEWORK

This paper provides a novel framework for rule extraction from unsupervised database by integrating three techniques namely clustering technique (*k*-means), fuzzification technique, and inductive learning technique (decision tree). The flow work of the proposed framework can be summarized as follows. The unsupervised database attributes have to pass through k-means algorithm. The k-means algorithm discovers the classes by partitioning the input records of unsupervised database into clusters. Records of the unsupervised database attributes that are similar to each other (i.e. records with similar attribute values) tend to be assigned to the same cluster, whereas records different from each other tend to be assigned to distinct clusters. Consequently, once the clusters are found, each cluster can be considered as a "class" and the input unsupervised database converted into supervised database. The supervised database passes through the fuzzification technique to convert the continuous values into intervals and transforms these intervals into linguistic terms. So that, now we can run the decision tree algorithm as an inductive learning technique on the clustered data, by using the cluster name as a class label. The decision tree algorithm is responsible for generating an accurate set of rules from supervised database. These rules are refined and may reduce the dimensionality of the extracted features (neglected the irrelevant features from database). The overall methodology of the proposed framework is shown in (**figure 1).**

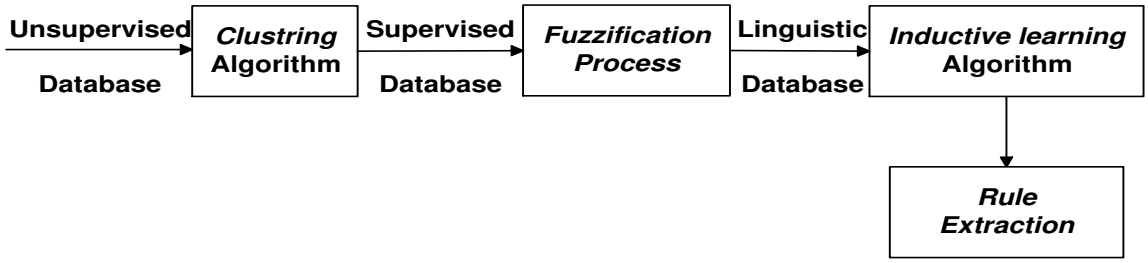

Figure 1: The proposed framework of rule extraction

III. CLUSTERING TECHNIQUE

Clustering is a search for hidden patterns that may exist in datasets. It is a process of grouping data objects into disjointed clusters so that the data in each cluster are similar, yet different to the others. Clustering techniques are applied in many application areas such as data analyses, pattern recognition, image processing, and information retrieval [29]. An operational definition of clustering can be stated as follows: Given a representation of n objects, find *k* groups based on a measure of similarity such that the similarities between objects in the same group are high while the similarities between objects in different groups are low [30]. Clustering algorithms can be broadly divided into two groups: hierarchical and partitional [31]. Hierarchical clustering algorithms starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy or starting with all the data points in one cluster and recursively dividing each Cluster into smaller clusters. Compared to hierarchical clustering algorithms, partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. The most well-known hierarchical algorithms are single-link and complete-link; the most popular and the simplest partitional algorithm is *k*-means. *K-means* is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. Ease of implementation, simplicity, efficiency, and empirical success are the main reasons for *k* -means popularity. The procedure of *k* -means follows a simple and easy way to classify a given data set through a certain number of clusters (assume *k* clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids shoud be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from

each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the *k* centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an objective function, in this case a squared error function. The objective function can be formulated as;

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \quad (1)$$

Where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster center c_j, is an indicator of the distance of the n data points from their respective cluster centers.

The following pseudo code explains the *k*-means algorithm steps:

A. Place k points into the space represented by the objects that are being clustered. These points represent initial group centroids.
B. Assign each object to the group that has the closest centroid.
C. When all objects have been assigned, recalculate the positions of the K centroids.
D. Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.

IV. FUZZIFICATION TECHNIQUE

If a given database includes continuous attributes (real values) then the search space based on all possible conjunction values for extracting rules yields to a burdensome computation and consumes much time. This problem can be solved by using a fuzzification process which leads to reduction of search space. The fuzzy subset of the universe of discourse $\cup$ is described by a membership function $\mu_v(V)$: $\cup \rightarrow$ [0,1], which represents the degree to which and belongs to the set V. A fuzzy linguistic variable, V, is an attribute whose domain contains linguistic values, which are labeled for the fuzzy subsets [32]. Therefore, the continuous attributes can be transformed into linguistic terms such as; Short (S), Medium (M), and Long (L). A non-overlapping rectangular membership functions may be used and the bounds of each linguistic term can be determined by using the smooth histogram of real values [33].

V. INDUCTIVE LEARNING TECHNIQUE

There are many machine learning approaches have been applied by learning module developers, including deductive reasoning (e.g. expert system), fuzzy logic, neural networks, genetic algorithm (GA), learning classifier systems (LCS), inductive logic programming, and decision tree (e.g. ID3 and C4.5). Traditional expert systems could get a very good trading performance based on a high quality knowledge base. However, it is always difficult to be acquired by human experts [34]. Fuzzy logic is used to clarify the ambiguous situations, but it is difficult to design a reasonable membership function [35]. Neural networks have been used in financial problems for a long time [36]. Unfortunately, the learned environment patterns of neural networks are like to be embedded into a black box, which is hard to explain. Simple genetic algorithm has also been used in financial problems popularly [37]. The only weak point is that GA seems result in a single rule and tries to apply for any environment states. The including inductive logic programming methods, decision tree, are based on supervised learning and apply the information gain theory to discriminate the attributes to construct a minimal-attributes decision tree. The advantages of decision tree representation are that the results are more comprehensive, easier to interpret, and they are in a well-organized knowledge structure. Since decision tree construction algorithms usually employ a greedy approach [38]. The knowledge obtained in the learning process of decision trees is represented in a tree where each internal node contains a question about one particular attribute (with one offspring per possible answer) and each leaf is labeled with one of the possible classes. A decision tree may be used to classify a given example; one begins at the root and follows the path provided by the answers to the questions in the internal nodes until a leaf is reached.

The measure of information gain is based on information entropy which computes the expected amount of information (in bits) needed for class prediction [39]. The entropy for a set of data S consisting of C classes and m features is given as follows;

$$\text{Entropy (S)} = -\sum_{j=1}^{C} \frac{|C_j|}{|S|} \text{Log}_2 \frac{|C_j|}{|S|} \quad (2)$$

Where:

C_j : Represents the subset of data belonging to the j^{th} class in set S.

Entropy for a particular feature (A_i) which has the values (v) in set (S) is given by;

$$\text{Entropy } (S, A_i) = \sum_{k=1}^{v} \frac{|S_k|}{|S|} \text{Entropy } (S_k) \quad (3)$$

Where:

v: The number of categories for the i^{th} feature.

S_k : The subset of data in S where feature A_i is assigned to the k^{th} category.

The information gain, which measures the effectiveness of a particular feature, is given by:

$$\text{Information Gain}(S, A_i) = \text{Entropy}(S) - \text{Entropy}(S, A_i) \quad (4)$$

The following pseudo code explains the decision tree steps:

Input: A data set, S

Output: A decision tree

A. If all the instances have the same value for the target attribute then return a decision tree that is simply this value.
B. Else

(a) Compute Gain values for all attributes and select an attribute with the highest value and create a node for that attribute.
(b) Make a branch from this node for every value of the attribute
(c) Assign all possibe values of the attribute to branches.
(d) Follow each branch by partitioning the dataset to be only instances whereby the value of the branch is present and then go back to 1.

VI. EXPERIMENTAL RESULTS

The proposed framework is applied in the iris database [40]. The database contains four real valued features {Sepal Length (SL), Sepal Width (SW), Petal Length (PL) and Petal Width (PW)}. They are used to classify three different classes of iris plant {Iris Setosa, Iris Versicolour and Iris Virginica}. There are 50 instances of each class of iris with no missing attributes. A sample of the original iris database is shown in **(table 1)**.

Table 1. A sample of original iris database

Sepal Length	Sepal Width	Petal Length	Petal Width	Target Class
5.1	3.7	1.5	0.4	setosa
5.5	3.9	1.7	0.4	setosa
4.9	3.1	4.8	0.1	setosa
7	3.2	4.7	1.4	versicolor
6.4	3.2	1.9	1.5	versicolor
5.7	2.9	4.2	1.2	versicolor
6.3	3.3	6	2.5	virginica
5.9	3.8	6.4	2	virginica
5.9	3	5.1	1.8	virginica

In order to evaluate the performance of the proposed framework, first the k –means is applied with number of cluster equal 3 (k = 3) to iris data without using the class information. Table 2 shows the confusion matrix by K-means clustering technique. The clustering accuracy of each class can be calculated by dividing the total number of correctly clustered cases in the desired class in the class by the total true number of cases. The proposed k-means technique clustering 50 cases in class Setosa from 50 true cases (accuracy 100%), clustering 47 cases in class Versicolor from 50 true cases (accuracy 94%), and clustering 46 cases in class Virginica from 50 true cases (accuracy 92%). Consequently, the average clustering accuracy against the true classes by the proposed K-means is 95.34%.

Table 2. Cluster result of iris data by the proposed k-means technique

True	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	0	47	3
Virginica	0	7	46

The proposed framework transforms the input continuous attributes into three linguistic terms called; Short, Medium and Wide. The smoothing histograms of the individual attributes for each class is performed and the bounds of linguistic terms are shown in (**table 3) [41].**

Consequently, all features in the given database are transformed into the linguistic terms as shown in (**table 4).**

The inductive learning technique can be applied on this database for extracting a set of comprehensible rules.

Table 3. Linguistic Terms for Iris Database.

Attribus Ling. Terms	Short (S)	Medium (M)	Wide (W)
Sepal Length (SL)	$4.3 \leq SL_S < 5.5$	$5.5 \leq SL_M < 6.1$	$6.1 \leq SL_W < 8.0$
Sepal Width (SW)	$2.0 \leq SW_S < 2.75$	$2.75 \leq SW_M < 3.2$	$3.2 \leq SW_W < 4.5$
Petal Length (PL)	$1.0 \leq PL_S < 2.0$	$2.0 \leq PL_M < 4.93$	$4.93 \leq PL_W < 7.0$
Petal Width (PW)	$0.1 \leq PW_S < 0.6$	$0.6 \leq PW_M < 1.7$	$1.7 \leq PW_W < 2.6$

Table 4. The given database as linguistic terms.

Sepal Length	Sepal Width	Petal Length	Petal Width	Target Class
S	L	S	S	setosa
M	S	S	S	setosa
L	M	M	M	versicolor
L	M	S	M	versicolor
L	M	L	L	virginica
M	M	L	L	virginica

Table 5 shows the rule extraction and the corresponding fuzzification rules from the proposed algorithm.

Rules induction *Fuzzification*	Rules induction *DeFuzzification*
If Petal Length is Short Then Setosa	If $1.0 \leq$ Petal Length < 2.0 Then Setosa
If Petal Width is Short Then Setosa	If $0.1 \leq$ Petal Width < 0.6 Then Setosa
If Petal Length is Medium and Petal Width is Medium Then Versicolor	If $2.0 \leq$ Petal Length< 4.93 and $0.6 \leq$ Petal Width < 1.7 ThenVersicolor
If Sepal Length is Short and Sepal Width is Wide Then Setosa	If $4.3 \leq$ Sepal Length <5.5 and $3.2 \leq$ Sepal Width < 4.5 Then Setosa
If Sepal Width is Wide and Petal Length is Wide Then Virginica	If $3.2 \leq$ Sepal Width < 4.5 and $4.93 \leq$ Petal Length<7.0 Then Virginica
If Sepal Length is medium and Petal Length is Wide and Petal width is Wide Then Virginica	If $5.5 \leq$ Sepal Length < 6.7 and $4.93 \leq$ Petal Length<7.0 and $1.7 \leq$ Petal width< 2.6 Then Virginica

VII. CONCLUSION

This paper focused on developing a knowledge extraction framework based on clustering the unsupervised database in order to reduce the search space for extracting

accurate rules. It integrates three algorithms namely clustering algorithm, fuzzifcation algorithm, and inductive learning algorithm. K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The main advantages of this algorithm are its simplicity and speed which allows it to run on large datasets. However, rule extraction from numerical data is a high computational complexity problem. Therefore the fuzzifcation algorithm is used as a preprocessing process for converting the numerical attributes into linguistic terms. The advantages of decision tree representation are that the results are more comprehensive, easier to interpret, and they are in a well-organized knowledge structure. The future work should consist of more experiments with other data sets, as well as more elaborated experiments to search on the other parameters, which enhance the performance of the proposed framework.

VIII. REFERENCES

[1] Weiss SM and Indurkhya N. Predictive Data Mining: a practical guide. Morgan Kaufmann, 1998.

[2] Fayyad UM, Piatetsky-Shapiro G and Smyth P. From data mining to knowledge discovery: an overview. In: Fayyad UM, Piatetsky-Shapiro G, Smyth P and Uthurusamy R. Advances in Knowledge Discovery & Data Mining, 1-34. AAAI/MIT, 1996.

[3] Pyle D. Data Preparation for Data Mining. Morgan Kaufmann, 1999.

[4] Klemettinen M, Mannila H, Ronkainen P, Toivonen H and Verkamo AI. Finding interesting rules from large sets of discovered association rules. Proc. 3rd Int. Conf. on Information and Knowledge Management. Gaithersburg, Maryland. Nov./Dec. 1994

[5] Liu B, Hsu W. and Chen S. Using general impressions to analyze discovered classification rules. Proc. 3rd Int. Conf. Knowledge Discovery & Data Mining, 31-36. AAAI Press, 1997.

[6] Freitas AA. Onobjective measures of rule surprisingness. Lecture Notes in Artificial Intelligence 1510: Principles of Data Mining and Knowledge Discovery (Proc. 2nd European Symp., PKDD´98, Nantes, France), 1-9. Springer-Verlag, 1998.

[7] Freitas AA. On Rule Interestingness Measures. Knowledge-Based Systems 12(5-6), 309-315. Oct. 1999.

[8] Gebhardt F. Choosing among competing generalizations. Knowledge Acquisition 3, 1991, 361-380.

[9] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Boston, MA, 1991.

[10] Y.C. Liaw, J.Z.C. Lai, Winston Lo, Image restoration of compressed image using classified vector quantization, Pattern Recognition 35 (2) (2002) 181–192.

[11] J. Foster, R.M. Gray, M.O. Dunham, Finite state vector quantization for waveform coding, IEEE Trans. Inf. Theory 31 (3) (1985) 348–359.

[12] J.Z.C. Lai, Y.C. Liaw, Winston Lo, Artifact reduction of JPEG coded images using mean-removed classified vector quantization, Signal Process. 82 (10) (2002), 1375–1388.

[13] S. Theodoridis, K. Koutroumbas, Pattern Recognition, second ed., Academic Press, New York, 2003.

[14] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, Advances in Knowledge Discovery and Data Mining, MIT Press, Boston, MA, 1996.

[15] D. Liu, F. Kubala, Online speaker clustering, in: Proceedings of IEEE Conference on Acoustic, Speech, and Signal Processing, vol. 1, 2004, pp. 333–336.

[16] P. Hojen-Sorensen, N. de Freitas, T. Fog, On-line probabilistic classification with particle filters, in: Proceedings of IEEE Signal Processing Society Workshop, vol. 1, 2000, pp. 386–395.

[17] M. Eirinaki, M. Vazirgiannis, Web mining for web personalization, ACM Trans.Internet Technol. 3 (2003) 1–27.

[18] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, A. Wu, An efficient k-means clustering algorithm: analysis and implementation, IEEE Trans. PAMI 24 (7) (2002) 881–892.

[19] J.R. Quinlan, "Induction of Decision Trees". Machine Learning, 1, 81-106, 1986.

[20] R.S. Michalski, "A Theory and Methodology of Inductive Learning", Machine Learning: An Artificial Intelligence Approach (Vol. I), Palo Alto, CA: Tioga Press, 83-134, 1983.

[21] D. Kibler and D.W. Aha, "Learning Representative Examples of Concepts: An Initial Case Study". Proceedings of the Fourth International Workshop on Machine Learning, Irvine, CA: Morgan Kaufmann, 24-30, 1987.

[22] D.H. Fisher, "Knowledge Acquisition via Incremental Conceptual Clustering". Machine Learning, 2, 139-172, 1987.

[23] T.M. Mitchell, "Generalization as Search". Artificial Intelligence, 18, 203-226. 1982.

[24] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation". Parallel Distributed Processing (Vol. I), Cambridge, MA: MIT Press, 318-362. 1986.

[25] J.R. Quinlan, "Learning Efficient Classification Procedures and Their Application to Chess Endgames", Machine Learning: An Artificial Intelligence Approach, Vol. I, CA: Tioga Press, 463-482, 1983.

[26] S.M. Weiss and N. Indurkhya, "Rule - Based Machine Learning Methods for Function Prediction". Journal of Artificial Intelligence Research, 3, 383-403, 1995.

[27] J.R. Quinlan, "Generating Production Rules from Decision Trees". Proceedings of the Tenth International Joint Conference on Artificial Intelligence, San Mateo,CA: Morgan Kaufmann, 304-307, 1987.

[28] R. Mooney, J. Shavlik, G. Towell, and A. Gove, "An Experimental Comparison of Symbolic and Connectionist Learning Algorithms". Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, San Mateo, CA: 775-780, 1989.

[29] Krista Rizman Zalik, "An efficient k0-means clustering algorithm", Pattern Recognition Letters 29 (2008) 1385–1391

[30] Merriam-Webster Online Dictionary, 2008. Cluster analysis. <http://www.merriamwebster- online.com>

[31] Amir Ahmad, Lipika Dey, "A k-mean clustering algorithm for mixed numeric and categorical data", Data & Knowledge Engineering 63 (2007) 503–527

[32] Vassilios Petridis, Vassilis G. Kaburlasos, " Clustering and Classification in Structured Data Domains Using

Fuzzy Lattice Neurocomputing (FLN)", IEEE Transactions on Knowledge and Data Engineering, March/April,Vol. 13, No. 2, 2001.

[33] J.H.Wang, Wen-Jeng Liu and Lian-Da Lin, " Histogram-Based Fuzzy Filter For Image Restoration", IEEE Trans On Systems, Man, and Cybernetics, Vol.32, No.2, PP. 230-238, Apr.2002.

[34] Giarratano, J., & Riley, G. (1998). Expert systems—principle and programming (3rd ed.). Boston, MA: PWS Publishing Company.

[35] McIvor, R. T., McCloskey, A. G., Humphreys, P. K., & Maguire, L. P. (2004). Using a fuzzy approach to support financial analysis in the corporate acquisition process. Expert Systems with Applications, 27, 533–547.

[36] Wang, Y. F. (2003). Mining stock price using fuzzy rough set system. Expert Systems with Applications, 24, 13–23.

[37] Oh, K. J., Kim, T. Y., & Min, S. (2005). Using genetic algorithm to support portfolio optimization for index fund management. Expert Systems with Applications, 28, 371–379.

[38] Kovalerchuk, B., & Vityaev, E. (2000). Data mining in finance. Dordrecht: Kluwer.

[39] J. R. QUINLAN, "Simplifying Decision Trees", International Journal of Human-Computer Studies, Volume 51, Issue 2, August 1999, Pages 497-510.

[40]ftp://ftp.ics.uci.edu/pub/machine-learning-databases/iris/

[41]http://www.fizyka.umk.pl/~duch/ref/kdd-tut/iris.html.