

DATA MINING BY MOUCLAS ALGORITHM FOR PETROLEUM RESERVOIR CHARACTERIZATION FROM WELL LOGGING DATA

Yalei Hao¹, Markus Stumptner¹, Gerald Quirchmayr^{1,2}, Qing He³

¹*Advanced Computing Research Centre, University of South Australia, SA5095, Australia,*

²*Institut für Informatik und Wirtschaftsinformatik, Universität Wien, Liebiggasse 4, A-1010 Wien, Austria,* ³*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China*

Abstract: Petroleum reservoir characterization is one of the most difficult and challenging tasks of the exploration of petroleum industry and usually a long and costly procedure. This paper proposes a novel kind of patterns for the classification over quantitative well logging data, which is called *MOUCLAS* (MOuntain function based CLASsification) Patterns, based on the concept of the fuzzy set membership function which gives the new approach a solid mathematical foundation and compact mathematical description of classifiers. It integrates classification, clustering and association rules mining to identify interesting knowledge in the well logging database. The aim of the study is the use of *MOUCLASS* patterns to interpret the pay zones from well logging data for the purpose of reservoir characterization. This approach is better than conventional techniques for well logging interpretation that require a precise understanding of the relation between the well logging data and the underlying property of interest.

Keywords: Data Mining, MOUCLAS Pattern, Petroleum Reservoir Characterization

1. INTRODUCTION AND MOTIVATION

Well logging data analysis, a geophysical prospecting technique, plays an essential role in petroleum exploration and exploitation. It is used to identify the pay zones of gas or oil in the reservoir formations¹. The techniques of

processing the huge databases such as well logging data is one of the main paths to increase and optimize production of oil and gas reservoirs. Accurate reservoir characterization through data analysis of well logging is an essential step in reservoir modeling & management and production optimization.

Three main types of well logging quantitative data, which are electrical, nuclear and acoustic data, can be collected by specific instruments lowering in the well borehole at different depth intervals. Then, the well logging interpreter analyzes the acquired data in order to recover the petrophysical parameters of formations across the borehole, such as porosity, permeability, lithology, grain size, amount of clay, water saturation, etc. Next, all these above findings are used for reservoir characterization so as to evaluate the reservoir formation². A large number of techniques have been introduced in order to establish an adequate interpretation model for the petrophysical parameters of formations over the past fifty years, such as the determination of permeability³.

In recent years the petroleum industry has witnessed a massive explosion in the volume and dimensions of data. It is caused by increased sampling rate, longer record acquisition, multi-component surveys, 4-D seismic, etc⁴. Thus we need efficient techniques to process such large databases. Beside the above conventional empirical and statistical techniques, data mining techniques have gained much attention since they can be an innovative option for well logging data analysis for the purpose of reservoir characterization and help identify the most information rich part of the large and high dimensional data sets.

Data mining based classification aims to build accurate and efficient classifiers not only on small data sets but more importantly also on large and high dimensional data sets, while the widely used traditional statistical data analysis techniques are not sufficiently powerful for this task^{5, 6}. With the development of new data mining techniques on association rules, new classification approaches based on concepts from association rule mining are emerging. These include such classifiers as ARCS⁷, CBA⁸, LB⁹, CAEP¹⁰, etc., which are different from the classic decision tree based classifier C4.5¹¹ and k-nearest neighbor¹² in both the learning and testing phases.

ARCS⁷ demonstrated the successful application of concepts of clustering for the purpose of classification. However, ARCS is limited to 2D-rules based classifiers of the format $\mathbf{A} \wedge \mathbf{B} \Rightarrow \mathbf{Class}_i$, where A and B are two predicates. It uses the method of “Binning” to discretize the value of quantitative attributes. Consequently, the accuracy of ARCS is strongly related to the degree of discretization used. A non-grid-based technique¹³ has been proposed to find quantitative association rules that can have more than two predicates in the antecedent. The authors noticed that the information

loss caused by partitioning could not be ignored and have tried to employ a measure of partial completeness to quantify the information lost, but the measure is still constrained by the framework of binning. Though there are several excellent discretization algorithms^{14, 15}, a standard approach to discretization has not yet been developed. Different approaches could lead to different collections of large itemsets even with respect to the same support threshold in a given data set. ARCS and the non-grid-based technique lead to research question 1 addressed: “Is it possible that an association rule based classifier can be developed for quantitative attributes by the concepts of clustering which can overcome the limitation caused by the discretization method?” CBA⁸ gives us an interesting indication that the idea of apriori property can be applied to a set of predicates (itemsets) for classification. Suppose an association rule based classifier in the form of $A_1 \wedge A_2 \wedge \dots \wedge A_l \Rightarrow C_i$, where $A_j (j=1, \dots, l)$ are predicate variables, C_i is the class label, the antecedent of the rule is a frequent itemset. This raises question 2: “If an association rule based classifier can be built based on the concept of clustering, is it possible that a link between CBA and ARCS can be found so that an association rule based classifier with any number of predicates in the antecedent can be setup by clustering?”

The above research issues establish a challenge that comes within our research focus. In this paper, we present a new approach to the classification over quantitative data in high dimensional databases, called *MOUCLAS* (MOUtain function based CLASsification), based on the concept of the fuzzy set membership function. It aims at integrating the advantages of classification, clustering and association rules mining to identify interesting patterns in selected sample data sets.

2. PROBLEM STATEMENT

We now give a formal statement of the problem of *MOUCLAS* Patterns (called *MPs*) and introduce some definitions. The *MOUCLAS* algorithm, similar to ARCS, assumes that the initial association rules can be agglomerated into clustering regions, while obeying the anti-monotone rule constraint. Our proposed framework assumes that the training dataset D is a normal relational set, where transaction $d \in D$. Each transaction d is described by attributes $A_j, j = 1$ to l . The dimension of D is l , the number of attributes used in D . This allows us to describe a database in terms of volume and dimension. D can be classified into a set of known classes $Y, y \in Y$. The value of an attribute must be quantitative. In this work, we treat all the attributes uniformly. We can treat a transaction as a set of (attributes,

value) pairs and a class label. We call each (attribute, value) pair an item. A set of items is simply called an itemset.

Since CBA indicates the feasibility of setting up a link between association rule and classification and ARCS proves that the idea of designing a classification pattern based on clustering can work effectively and efficiently, we design a *MOUCLAS* Pattern (so called *MP*) as an implication of the form:

$$\mathbf{Cluster}(D)_t \rightarrow y,$$

where $\mathbf{Cluster}(D)_t$ is a cluster of D , $t = 1$ to m , and y is a class label. The definitions of *frequency* and *accuracy* of *MOUCLAS* Patterns are defined as following: The *MP* satisfying minimum support is *frequent*, where *MP* has support s if $s\%$ of the transactions in D belong to $\mathbf{Cluster}(D)_t$, and are labeled with class y . The *MP* that satisfies a pre-specified minimum confidence is called *accurate*, where *MP* has confidence c if $c\%$ of the transactions belonging to $\mathbf{Cluster}(D)_t$ are labeled with class y .

Though framework of support – confidence is used in most of the applications of association rule mining, it may be misleading by identifying a rule $\mathbf{A} \Rightarrow \mathbf{B}$ as interesting, even though the occurrence of \mathbf{A} may not imply the occurrence of \mathbf{B} . This requires a complementary framework for finding interesting relations. Correlation¹⁶ is one of the most efficient interestingness measures other than support and confidence. Here we adopt the concept of reliability¹⁷ to describe the correlation. The measure of reliability of the association rule $\mathbf{A} \Rightarrow \mathbf{B}$ can be defined as:

$$\mathbf{reliability} \quad R(\mathbf{A} \Rightarrow \mathbf{B}) = \left| \frac{P(\mathbf{A} \wedge \mathbf{B})}{P(\mathbf{A})} - P(\mathbf{B}) \right|$$

Since R is the difference between the conditional probability of \mathbf{B} given \mathbf{A} and the unconditional of \mathbf{B} , it measures the effect of available information of \mathbf{A} on the probability of the association rule. Correspondingly, the greater R is, the stronger *MOUCLAS* patterns are, which means the occurrence of $\mathbf{Cluster}(D)_t$ more strongly implies the occurrence of y . Therefore, we can utilize reliability to further prune the selected *frequent and accurate and reliable MOUCLAS* patterns (*MPs*) to identify the truly interesting *MPs* and make the discovered *MPs* more understandable. The *MP* satisfying minimum reliability is *reliable*, where *MP* has reliability defined by the above formula.

Given a set of transactions, D , the problems of *MOUCLAS* are to discover *MPs* that have support and confidence greater than the user-specified minimum support threshold (called *minsup*)¹⁸, and minimum confidence threshold (called *minconf*)¹⁸ and minimum reliability threshold (called *minR*) respectively, and to construct a classifier based upon *MPs*.

3. THE MOUCLAS ALGORITHM

The classification technique, *MOUCLAS*, consists of two steps:

1. Discovery of *frequent, accurate and reliable MPs*.
2. Construction of a classifier, called *De-MP*, based on *MPs*.

The core of the first step in the *MOUCLAS* algorithm is to find all *cluster_rules* that have support above *minsup*. Let *C* denote the dataset *D* after dimensionality reduction processing. A *cluster_rule* represents a *MP*, namely a rule:

$$cluset \rightarrow y,$$

where *cluset* is a set of itemsets from a cluster ***Cluster(C)***, *y* is a class label, $y \in Y$. The support count of the *cluset* (called *clusupCount*) is the number of transactions in *C* that belong to the *cluset*. The support count of the *cluster_rule* (called *cisupCount*) is the number of transactions in *D* that belong to the *cluset* and are labeled with class *y*. The *confidence* of a *cluster_rule* is $(cisupCount / clusupCount) \times 100\%$. The support count of the class *y* (called *clasupCount*) is the number of transactions in *C* that belong to the class *y*. The *support* of a class (called *clasup*) is $(clasupCount / |C|) \times 100\%$, where $|C|$ is the size of the dataset *C*.

Given a *MP*, the *reliability R* can be defined as:

$$R(cluset \rightarrow y) = \left| (cisupCount / clusupCount) - (clasupCount / |C|) \right| \times 100\%$$

The traditional association rule mining only uses a single *minsup* in rule generation, which is inadequate for many practical datasets with uneven class frequency distributions. As a result, it may happen that the rules found for infrequent classes are insufficient and too many may be found for frequent classes, inducing useless or over-fitting rules, if the single *minsup* value is too high or too low. To overcome this drawback, we apply the theory of mining with multiple minimum supports¹⁹ in the step of discovering the frequent *MPs* as following.

Suppose the total support is *t-minsup*, the different minimum class support for each class *y*, denoted as ***minsup_i***, can be defined by the formula:

$$minsup_i = t-minsup \times freqDistr(y)$$

where *freqDistr(y)* is the function of class distributions. *Cluster_rules* that satisfy ***minsup_i*** are called *frequent cluster_rules*, while the rest are called *infrequent cluster_rules*. If the *confidence* is greater than *minconf*, we say the *MP* is *accurate*.

The first step of *MOUCLAS* algorithm works in three sub-steps, by which the problem of discovering a set of *MPs* is solved:

Algorithm: Mining *frequent* and *accurate* and *reliable* MOUCLAS patterns (*MPs*)

Input: A training transaction database, D ; minimum support threshold (*minsup*); minimum confidence threshold (*minconf*); minimum reliability threshold (*minR*)

Output: A set of *frequent*, *accurate* and *reliable* MOUCLAS patterns (*MPs*)

Methods:

(1) Reduce the dimensionality of transactions d , which efficiently reduces the data size by removing irrelevant or redundant attributes (or dimensions) from the training data, and

(2) Identify the clusters of database C for all transactions d after dimensionality reduction on attributes A_j in database C , based on the Mountain function, which is a fuzzy set membership function, and specially capable of transforming quantitative values of attributes in transactions into linguistic terms, and

(3) Generate a set of *MPs* that are both *frequent*, *accurate* and *reliable*, namely, which satisfy the user-specified minimum support (called *minsup*), minimum confidence (called *minconf*) and minimum reliability (called *minR*) constraints.

In the first sub-step, we reduce the dimensionality of transactions in order to enhance the quality of data mining and decrease the computational cost of the MOUCLAS algorithm. Since, for attributes A_j , $j=1$ to l in database, D , an exhaustive search for the optimal subset of attributes within 2^l possible subsets can be prohibitively expensive, especially in high dimensional databases, we use heuristic methods to reduce the search space. Such greedy methods are effective in practice, and include such techniques as stepwise forward selection, stepwise backward elimination, combination of forwards selection and backward elimination, etc. The first sub-step is particularly important when dealing with raw data sets. Detailed methods concerning dimensionality reduction can be found in some papers²⁰⁻²³.

Fuzzy based clustering is performed in the second sub-step to find the clusters of quantitative data. The Mountain-climb technique proposed by R. R. Yager and D. P. Filev²⁴ employed the concept of a mountain function, a fuzzy set membership function, in determining cluster centers used to initialize a Neuro-Fuzzy system. The subtractive clustering technique²⁵ was defined as an improvement of Mountain-climb clustering. A similar approach is provided by the DENCLUE algorithm²⁶, which is especially efficient for clustering on high dimensional databases with noise. The techniques of Mountain-climb clustering, Subtractive clustering and Denclue provide an effective way of dealing with quantitative attributes by mountain functions (or influence functions), which has a solid mathematical

foundation and compact mathematical description and is totally different from the traditional processing method of binning. It offers us an opportunity of mining the patterns of data from an innovative angle. As a result, question 1 presented in the introduction can now be favorably answered.

The observation that, a region which is dense in a particular subspace must create dense regions when projected onto lower dimensional subspaces, has been proved by R. Agrawal and his research cooperators in CLIQUE²⁷. In other words, the observation follows the concepts of the apriori property. Hence, we may employ prior knowledge of items in the search space based on the property so that portions of the space can be pruned. The successful performance of CLIQUE has again proved the feasibility of applying the concept of apriori property to clustering. It brings us a step further towards the solution of problem 2, that is, if the initial association rules can be agglomerated into clustering regions, just like the condition in ARCS, we may be able to design a new classifier for the purpose of classification, which confines its search for the classifier to the cluster of dense units of high dimensional space. The answer to question 2 can contribute to the third sub-step of the MOUCLAS algorithm to the forming of the antecedent of *cluster_rules*, with any number of predicates in the antecedent. In the third sub-step, we identify the candidate *cluster_rules* which are actually *frequent* and *accurate* and *reliable*. From this set of *frequent* and *accurate* and *reliable cluster_rules*, we produce a set of *MPs*.

```

1 X = reduceDim (I); // reduce the dimensionality on the set of all items I of in D
2 Cluster(C)i = genCluster (C); // identify the complete clusters of C
3 for each Cluster(C)i do
    E = genClusterrules(cluset, class); // generate a set of candidate cluster_rules
4 for each transaction d ∈ C do
5     Ed = genSubClusterrules (E, d); // find all the cluster_rules in E whose
    cluset are supported by d
6 for each e ∈ Ed do
7     e.clusupCount++;
    // accumulate the clusupCount of the cluset of cluster_rule e
8     if d.class = e.class then e.cisupCount++
    // accumulate the cisupCount of cluster_rule e supported by d
9 end
10 end
11 F = {e ∈ E | e.cisupCount ≥ minsupi }; // construct the set of frequent cluster_rules
12 MP = genRules (F); // generate MP using the genRules function by minconf and minR
13 end
14 MPs = □ MP; // discover the final set of MPs

```

Figure 1: The First Step of the MOUCLAS Algorithm

Let I be the set of all items in D , C be the dataset D after dimensionality reduction, where transaction $d \in C$ contains $X \subseteq I$, a k -itemset. Let E denote the set of candidates of cluster_rules, where $e \in E$, and F denote the set of frequent cluster_rules. The first step of the *MOUCLAS* algorithm is given in Figure 1.

The task of the second step in *MOUCLAS* algorithm is to use a heuristic method to generate a classifier, named *De-MP*, where the discovered *MPs* can cover D and are organized according to a decreasing precedence based on their confidence and support. Suppose R be the set of *frequent, accurate and reliable MPs* which are generated in the past step, and $MP_{default_class}$ denotes the default class, which has the lowest precedence. We can then present the *De-MP* classifier in the form of

$\langle MP_1, MP_2, \dots, MP_n, MP_{default_class} \rangle$,
 where $MP_i \in R$, $i = 1$ to n , $MP_a \succ MP_b$ if $n \geq b > a \geq 1$ and $a, b \in i$, $C \subseteq$
 \square cluset of MP_i .

The second step of the *MOUCLAS* algorithm also consists of three sub-steps, by which the *De-MP* classifier is formed:

Algorithm: Constructing *De-MP* Classifier

Input: A training database after dimensionality reduction, C ; The set of *frequent and accurate and reliable MOUCLAS patterns (MPs)*

Output: *De-MP* Classifier

Methods:

- (1) Identify the order of all discovered *MPs* based on the definition of precedence and sequence them according to decreasing precedence order.
- (2) Determine possible *MPs* for *De-MP* classifier from R following the descending sequence of *MPs*.
- (3) Discard the *MPs* which cannot contribute to the improvement of the accuracy of the *De-MP* classifier and keep the final set of *MPs* to construct the *De-MP* classifier.

In the first sub-step, the *MPs* are sorted in descending order, which has the training transactions surely covered by the *MPs* with the highest precedence when possible in the next sub-step. The sort of the whole set of *MPs* is performed following the definition of *precedence*:

Given two *MPs*, we say that MP_a has a higher precedence than MP_b , denoted as $MP_a \succ MP_b$,

if $\forall MP_a, MP_b \in MP_s$, it holds that: the confidence of MP_a is greater than that of MP_b , or if their confidences are the same, but the support of MP_a is greater than that of MP_b , or if both the confidences and supports of MP_a and MP_b are the same, but MP_a is generated earlier than MP_b .

In the second sub-step, we test the *MPs* following decreasing precedence and stop the sub-step when there is no rule or no training transaction. For each *MP*, we scan C to find those transactions satisfying the cluset of the

MP. If the *MP* can correctly classify one transaction, we store it in a set denoted as *L*. Those transactions satisfying the cluset of the *MP* will be removed from *C* at each pass. Each transaction can be identified by a unique ID. The next pass will be performed on the remaining data. A default class is defined at each scan, which is the majority class in the remaining data. At the end of each pass, the total number of errors that are made by the current *L* and the default class are also stored. When there is no rule or no training transaction left, we terminate this sub-step. After this sub-step, every *MP* in *L* can correctly classify at least one training transaction in *C*.

In the third sub-step, though we would like to find as many *MPs* as possible to give good coverage of the training transactions in the second sub-step, we prefer strong *MPs* which have relatively high support and confidence, due to their characteristics of corresponding to larger coverage and stronger differentiating power. Meanwhile, we hope that the *De-MP* classifier, consisting of a combination of strong *MPs*, has a relatively smaller number of classification errors, because of greedy strategy. In addition, the reduction of *MPs* can increase the understandability of the classifier. Therefore, in this sub-step, we identify the first *MP* with the least number of errors in *L* and discard all the *MPs* after it because these *MPs* produce more errors. The undiscarded *MPs* and the default class corresponding to the first *MP* with the least number of errors in *L* form our *De-MP* classifier.

The second step of the *MOUCLAS* algorithm is shown in Figure 2.

```

1 R = sort(R); // sort MP's based on their precedence
2 for each MP ∈ R in sequence do
3   temp = ∅;
4   for each transaction d ∈ C do
5     if d satisfies the cluset of MP then
6       store d.ID in temp;
7       if MP correctly classifies d then
8         insert MP at the end of L;
9       delete the transaction who has ID in temp from C;
10      selecting a default class for the current L;
        // determine the default class based on majority class of remaining transactions in C
11    end
12    compute the total number of errors of L;
        // compute the total number of errors that are made by the current L and the default class
13  end
14 Find the first MP in L with the lowest total number of errors and discard all the MP's
   after the MP in L;
15 Add the default class associated with the above mentioned first MP to end of L;
16 De-MP classifier = L

```

Figure 2: The Second Step of the *MOUCLAS* Algorithm

In the testing phase, when we classify a new transaction, the first *MP* in *De-MP* satisfying the transaction is used to classify it. In *De-MP* classifier, *default_class*, having the lowest precedence, is used to specify a default class for any new sample that is not satisfied by any other *MPs* as in C4.5¹¹, CBA⁸.

4. EXAMPLE OF *MOUCLAS* APPLICATION IN RESERVOIR CHARACTERIZATION

Oil/gas formation identification is a vital task of reservoir characterization in the petroleum industry, where the petroleum database contains such records (or attributes) as seismic data, various types of well logging data and petrophysical property data whose values are all quantitative.

An illustration of using well logging data for purpose of oil/gas formation identification is illustrated in Figure 3. The well logging data sets include attributes (well logging curves) of GR (gamma ray), RDEV (deep resistivity), RMEV (shallow resistivity), RXO (flushed zone resistivity), RHOB (bulk density), NPHI (neutron porosity), PEF (photoelectric factor) and DT (sonic travel time). Since most of the reservoirs are horizontally and vertically heterogeneous, no depth information is used for training.

One transaction of the database can be treated as a set of the items corresponding to the same depth and a class label (oil/gas formation or not). A hypothetically useful *MP* may suggest a relation between well logging data and the class label of oil/gas formation since. In this sense, a selected set of such *MPs* can be a useful guide to petroleum engineers to identify possible drilling targets and their depth and thickness at the stage of exploration and exploitation.

MOUCLAS aims at deriving an explicit or implicit heuristic relationship between measured values (well logging data) and properties to be predicted (oil/gas formation or not). The *MOUCLAS* method is ideally suitable to establish such implicit relationships through proper training. The notable advantage of *MOUCLAS* over more traditional processing techniques such as model based well logging analysis is that a physical model to describe the relationship between the well logging data and the property of interest is not needed; nor is an very precise understanding of the physical phenomena of the well logging data. From this point of view, *MOUCLAS* provides a complementary and useful technical approach towards the interpretation of petroleum data and benefits petroleum discovery.

we attempt to establish a relationship between different well logs, seismic attributes, laboratory measurements and other reservoir properties.

ACKNOWLEDGEMENT

This work was partially supported by the Australia-China Special Fund for Scientific and Technological Cooperation under grant CH030086. We are grateful to Dr. Alexander Hinneburg for fruitful discussion.

REFERENCES

1. D.V. Ellis, *Well Logging for Earth Scientists*, Elsevier Science Publishing Co. (1987)
2. M. Rider, *The Geological Interpretation of Well Logs*, Second Edition, Whittles Publishing, (1996)
3. B. Balan, S. Mohaghegh, S., and S. Ameri, "State-Of-The-Art in Permeability Determination From Well Log Data: Part 1 -A Comparative Study, Model Development," SPE Technical Report 30978, (1995)
4. F. Aminzadeh, *Future Geoscience Technology Trends in, Stratigraphic Analysis, Utilizing Advanced Geophysical, Wireline, and Borehole Technology For Petroleum Exploration and Production*, GCSEPFM pp 1-6, (1996)
5. Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. From data mining to knowledge discovery: An overview. *Advances in knowledge discovery and data mining*. AAAI/MIT Press. (1996) 1-34
6. Han, J., & M. Kamber. *Data mining: concepts and techniques*. Morgan Kaufmann Publishers. (2000)
7. B. Lent, A. Swami, and J. Widom. Clustering association rules. *ICDE'97*, (1997) 220-231
8. B. Liu, W.Hsu, and Y.Ma. Integrating classification and association rule mining. *KDD'98*. (1998) 80-86
9. Meretakis, D., & Wuthrich, B. Extending naive Bayes classifiers using long itemsets. *Proc. of the Fifth ACM SIGKDD*. ACM Press. (1999) 165-174
10. Dong, G., & Li, J. Efficient mining of emerging patterns: Discovering trends and differences. *Proc. of the Fifth ACM SIGKDD*. (1999)
11. Quinlan, J. R. *C4.5: Programs for machine learning*. San Mateo, CA: Morgan Kaufmann. (1993)
12. Cover, T. M., & Hart, P. E. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13. (1967) 21-27
13. R. Skikant and R. Agrawal. Mining quantitative association rules in large relational tables. *SIG-MOD'96*, (1996) 1-12.
14. Fayyad, U., & Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. *Proc. of the 13th Int'l Conf. on Artificial Intelligence*. Morgan Kaufmann. (1993) 1022--1029
15. Dougherty, J., Kohavi, R., & Sahami, M. Supervised and unsupervised discretization of continuous features. *Proc. of the Twelfth Int'l Conf. on Machine Learning* pp. 94--202. Morgan Kaufmann. (1995)
16. Han, J., Pei, J., & Yin, Y. Mining frequent patterns without candidates generation. *Proc. of the 2000 ACM-SIGMOD*. ACM Press. (2000) 1-12

17. Khalil M. Ahmed, Nagwa M. El-Makky, Yousry Taha: A note on "Beyond Market Baskets: Generalizing Association Rules to Correlations". In The Proceedings of SIGKDD Explorations Volume 1, Issue 2, (2000) 46-48
18. Agrawal, R., Srikant, R. Fast algorithms for mining association rules. Proc. of the 20th VLDB (1994) 487-499
19. Bing Liu, Wynne Hsu, Yiming Ma, "Mining Association Rules with Multiple Minimum Supports" Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-99), August 15-18, San Diego, CA, USA (1999)
20. Dong, G., & Li, J. Feature selection methods for classification. *Intelligent Data Analysis: An International Journal*, 1, (1997)
21. H. Liu and H. Motoda, editors. *Feature Selection for Knowledge Discovery and Data Mining*. Boston: Kluwer Academic Publishers, (1998)
22. W.Sarawagi and M. Stonebraker. On automatic feature selection. *Int'l J. of Pattern Recognition and Artificial Intelligence*, 2, (1988) 197-220.
23. R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, (1997) 273-324
24. Yager, R. and D. Filev, "Generation of Fuzzy Rules by Mountain Clustering," *Journal of Intelligent & Fuzzy Systems*, Vol. 2, No. 3, (1994) 209-219
25. Chiu, S. L. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy System*, 2(3), (1994)
26. A. Hinneburg and D. Keim. An efficient approach to clustering in large Multimedia dataset with noise. *KDD'98*, (1998) 58-65
27. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. *SIGMOD'98*. (1998)