Proceedings of PAKDD99, Beijing, China, April.26-28,1999.

Mining Classification Knowledge Based on Cloud Models

Jianhua Fan and Deyi Li

The Institute of Electronic System Engineering
No. 307, Zhengchang Zhuang, Fengtai District, Beijing, China, 100039
Jianhuaf@hotmail.com
ziqin@public2.bta.net.cn

Abstract: Data classification is an important research topic in the field of data mining and knowledge discovery. It finds the common properties among a set of objects in a database and classifies them into different classes. There have been many data classification methods studied, including decision-tree method, statistical methods, neural networks, rough sets, etc. In this paper, we present a new mathematical representation of qualitative concepts—Cloud Models. With the new models, mapping between quantities and qualities becomes much easier and interchangeable. Based on the cloud models, a novel qualitative strategy for data classification in large relational databases is proposed. Then, the algorithms for classification are developed, such as cloud generation, complexity reduction, identifying interacting attributes, etc. Finally, we perform experiments on a challenging medical diagnosis domain, acute abdominal pain. The results show the advantages of the model in the process of knowledge discovery.

Keywords: Cloud Models, KDD, Data Mining, Classification, Soft-Computing.

Proceedings of PAKDD99, Beijing, China, April.26-28,1999.

Mining Classification Knowledge Based on Cloud Models

Jianhua Fan and Deyi Li

The Institute of Electronic System Engineering
No. 307, Zhengchang Zhuang, Fengtai District, Beijing, China, 100039
Jianhuaf@hotmail.com
ziqin@public2.bta.net.cn

Abstract: Data classification is an important research topic in the field of data mining and knowledge discovery. It finds the common properties among a set of objects in a database and classifies them into different classes. There have been many data classification methods studied, including decision-tree method, statistical methods, neural networks, rough sets, etc. In this paper, we present a new mathematical representation of qualitative concepts—Cloud Models. With the new models, mapping between quantities and qualities becomes much easier and interchangeable. Based on the cloud models, a novel qualitative strategy for data classification in large relational databases is proposed. Then, the algorithms for classification are developed, such as cloud generation, complexity reduction, identifying interacting attributes, etc. Finally, we perform experiments on a challenging medical diagnosis domain, acute abdominal pain. The results show the advantages of the model in the process of knowledge discovery.

Keywords: Cloud Models, KDD, Data Mining, Classification, Soft-Computing.

1 Introduction

With massive amounts of data stored in databases, mining information and knowledge in databases has become an important issue in recent research. Researchers in many different fields have shown great interest in data mining and knowledge discovery in databases (DMKD)[1].

Data classification is an important research topic in the field of data mining and knowledge discovery. It finds the common properties among a set of objects in a database and classifies them into different classes. To construct a classification model, a small database E is treated as the training set, in which each tuple consists of the same set of multiple attributes (or features) as the tuples in a large database W, and additionally, each tuple has a known class identity (label) associated with it. The objective of the classification is to first analyze the training data and develop an accurate description or a model for each class using the features present in the data. Such class descriptions are then used to classify future test data in the database W or to develop a better description (called classification rules) for each class in the database.[2] Applications of classification include medical diagnosis, performance prediction, selective marketing, etc.

In machine learning studies, a decision-tree classification method, developed by Quinlan[2,14], has been influential. It is a supervised learning method that constructs decision trees from a set of examples. The method first chooses a subset of the training examples to form a decision tree. If the tree does not give the correct answer for all the objects, a selection of the exceptions is added to the window and the process continues until the correct decision set is found. The eventual outcome is a tree in which each leaf carries a class name, and each interior node specifies an attribute with a branch corresponding to each possible value of that attribute.

A typical decision tree learning system, ID-3[2], adopts a top-down irrevocable strategy that searches only part of the search space. It guarantees that a simple, but not necessarily the simplest, tree is found. ID-3 uses an information-theoretic approach aimed at minimizing the expected number of tests to classify an object. The attribute selection part of ID-3 is based on the plausible assumption that the complexity of the decision tree is strongly related to the amount of information conveyed by this message. An information-based heuristic selects the attribute providing the highest information gain, i.e., the attribute that minimizes the information needed in the resulting subtrees to classify the elements. The ID-3 system uses information gain as the evaluation functions form classification, with the following evaluation function:

$$i = \Sigma (p_i \, ln(p_i)),$$

where p_i is the probability that an object is in class i. An extension to ID-3, C4.5[14],extends the domain of classification from categorical attributes to numerical ones.

Shan. et. al.[9,10,11] proposed an approach, which uses rough sets to ensuring the completeness of the classification and the reliability of the probability estimate prior to rule induction. Briefly, Rough sets method performs three steps to obtain classification rules. First, it generalizes the condition attributes as necessary to increase the credibility of the classification. It applies attribute-oriented concept tree ascension to reduce the complexity of an information system, and generalizes a condition attribute to a certain level based on the attribute's concept tree, which is provided by knowledge engineers or domain experts. The number of possible values at a higher level of an attribute is always smaller than at a lower level of, so the theoretical complexity is reduced. Then, the method identifies clusters of interacting attributes, i.e., search for credible classifications of the database tuples based on these clusters. A classification is credible if it is complete or almost complete with respect to the domain from which the database was collected. Finally, it searches for acceptable classifications. Classifications, which result in the good approximation of the concept of the interest, in the rough sets sense, are subsequently selected to obtain the classification rules.

There have been many other approaches on data classification, such as statistical approaches[15]. There have also been some studies of classification techniques in the context of databases[16]. An interval classifier has been proposed in [17] to reduce the cost of decision tree generation. An attribute-oriented induction method has been developed for mining classification rules in relational databases[16]. The work in [17] explores rule extraction in a database based on neural networks.

In this paper, we present a new mathematical representation of qualitative concepts—Cloud Models. With the new models, a novel approach for data classification in large relational databases is proposed.

2 The Qualitative Strategy of Data Classification

2.1 The Abstract Model of Data Classification

A relational database can be viewed as an information system. Formally, an information system S is a quadruple $<U, A, V, F>$, where U is a nonempty set of objects called *universe*; A is a finite set of attributes consisting of condition attributes C and decision attributes D such that $A=C\cup D$ and $C\cap D=\varnothing$; $V=\cup_{p\in A}V_p$ is a nonempty finite set of values of attributes A and V_p is the domain of the attribute p (the set of values of attribute p); $F : U\times A\rightarrow V$ is an information function which assigns particular values from the domain of attributes A to objects such that $f(x_i,p)\in V_p$ for all $x_i \in U$ and $p\in A$.

Any subset of condition attributes defines a classification of the universe of objects U as follows. Let B be a nonempty subset of C, and let x_i,x_j be members of U. The projection of the function f onto attributes belongs to the subset B will be denoted as f_B . A binary relation $R(B)$, called an indiscernibility relation, is first defined as follows:

$$R(B) = \{(x_i , x_j) \in U^2 : \quad f_B (x_i) = f_B (x_j)\}$$

We say that x_i and x_j are indiscernible by a set of attributes B in S iff $f (x_i , p)=f (x_j , p)$ for every $p\in B$. $R(B)$ is an equivalence relation on U for every $B\subset C$, which classifies the objects in U into a finite, preferably small, number of equivalence classes. The set of equivalence classes is called the classification $R^*(B)$. The pair $<U, R(B)>$ is called approximation space.[18]

The above model cannot, however, be directly applied to most KDD systems. A database represents only a subset (a sample) U' of the universe U about which we are trying to discover something. Depending on the selection of the information function f, the subset of the attributes B, the size and the distribution of objects in the sample U', we may or may not have all values of the information function f_B in our database. If all values are present then our knowledge about the classification is *complete* (despite not having all domain objects in the database); otherwise our knowledge about the classification is *incomplete*. To properly reason about the relationships occurring in U, the classification must be complete; otherwise, false conclusions may be drawn.

In this situation, qualitative strategy will take the advantage to deal with such kind of problems.

2.2 Qualitative Strategy

Data Mining and knowledge discovery in databases(DMKD) is considered to be the non-trivial extraction of implicit, previously unknown, and potentially useful knowledge from large numbers of small, very specific instances. The laboriousness of the development of realistic DMKD applications, previously reported examples of knowledge discovery in the literature and our experience in real-world knowledge discovery situations all lead us to believe that knowledge discovery is a representation-sensitive, human-oriented task consisting of friendly interactions between a human and a discovery system. Current work in DMKD uses some form of (extended or modified) SQL as the data mining query language and some variant of predicate calculus for the discovered results. The variants frequently contain some form of quantitative modifier, such as confidence, support,, threshold, and so forth[19]. This tends to lead to discovered rules such as :

> *With 37.8% of support and 15.7% of confidence, patients whose age are between 20 and 30 and have acute pain on the low-right side of the abdomen more than 6.24 hours can be classified as appendicitis.*

Rather than the qualitative representation:

> *Generally speaking, young patients who have acute low-right abdominal pain for a relative long time may get appendicitis.*

This is, however, more than a simple issue of semantics and friendliness. The former rules are not robust under change to the underlying database, while the latter ones are. In a real, very large database, data are often infected with errors due to the nature of collection. In addition, an on-line discovery system supporting a real database must keep up with changing data. It does not make much sense if a very precisely quantitative assertion is made about the behavior of an application domain based on such a database. It may be necessary, to some extent, to abandon the high standards of rigor and precision used in conventional quantitative techniques. At this moment, a piece of qualitative knowledge extracted may be more tolerant and robust. Clearly, quantitative results such as confidence and support cannot remain constant under conditions of any change. By contrast, qualitative representation will remain true until there is a substantial change in the database.

On the other hand, quantitative knowledge discovered at some lower levels of generalization in DMKD may still be suitable, but the number of the extracted rules increases. In contrast, as the generalization goes up to a higher level of abstraction, the discovered knowledge is more strategic. The ability to discovery quantitative and yet significant knowledge about the behavior of an application domain from a very large database diminishes until a threshold may be reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics. That is to say that, if the generalization to a very large database system exceeds a limit, the reality and exactness of its description become incompatible. This has been described as the principle of incompatibility. To describe phenomena qualitatively, we take linguistic variables and linguistic terms and show how quantitative and qualitative inference complement and interact with each other in a simple mechanism.

Our point of departure in this paper is to represent linguistic terms in logical sentences or rules. We imagine a linguistic variable that is semantically associated with a list of all the linguistic terms within a universe of discourse. For example, “age” is a linguistic variable if its values are “young”, “middle-age”, “old”, “very old” and so forth, rather than the real ages which are considered as the universe of discourse of the linguistic variable “age," say from 0 to 120. In the more general case, a linguistic variable is a tri-tuple {X, T(x), $C_x(u)$} in which X is the name of the variable, T(x) is the term-set of X; that is, the collection of its linguistic values, U is a universe of discourse, C_x (u) is a compatibility function showing the relationship between a term x in T(x) and U. More precisely, the compatibility function maps the universe of discourse into the interval [0,1] for each u in U.

It is important to understand the notion of compatibility functions. Consider a set of linguistic terms, T, in a universe of discourse, U, --- for example, the linguistic term, “young” in the interval [0,100]. T is characterized by its compatibility function C_x : u [0,1]. The statement that the compatibility of, say, “28 years old” with “young” is about 0.7, has a relationship both to fuzzy logic and probability.

In relation to fuzzy logic, the correct interpretation of the compatibility value “0.7” is that it is an indication of the partial membership to which the element “age-value 28” belongs to the fuzzy concept of the label “young”. To understand the relationship with probability on the other hand, the correct interpretation of the compatibility value “0.92” is that it is merely subjective indication. Human knowledge does not conform to such a fixed crisp membership degree “0.7” at the “28 years old”. There do not exist any unique partial membership values, which could be universally accepted by human beings to the universe of discourse U. But there is a random variable showing that the membership degree at “28 years old” takes a random value, behind which a subjective probability distribution is obeyed. The degree of compatibility takes on random value itself. This type of randomness is adhered to the fuzziness.

Regarding syntactic generation, we shall usually assume that a linguistic variable is structured in the sense that it is associated with two rules. The first is the atomic generator rule. It specifies the manner in which a linguistic atom, which cannot be spliced into any smaller parts, may be generated. The second, the semantic rule, specifies a procedure for computing composite linguistic terms based on linguistic atoms.

In addition to linguistic atoms, a linguistic term may involve connectives (such as “and”, “or”, “either” and “neither”), the negation (“not”) and the hedges (such as “very”, “more or less”, “completely”, “quite”, “fairly”, “extremely” and “somewhat”). The linguistic connectives, hedges and negation may be treated as (some form of) soft operators which modify the meaning of their operands, linguistic atoms, in a soft computing fashion to become composite linguistic terms. That is the business of the semantic rule[19].

3 Qualitative Representation Based on Cloud Models

3.1 Cloud Models

Following the important characteristics of linguistic variables and terms, we define a new concept of cloud models to represent linguistic terms. Let U be the set, U = {u}, as the universe of discourse, and T, a linguistic term associated with U. The membership degree of u in U to the linguistic term T, C_T (u), is a random variable with a probability distribution. C_T(u) takes values in [0,1]. A membership cloud is a mapping from the universe of discourse U to the unit interval [0,1].

The concept of membership clouds is often pictured as two-dimensional graphs. The geometry of membership clouds is a great aid in understanding the uncertainty. It is important to see the properties of the clouds. First of all, the mapping from all u in U to the interval [0,1], is an one-point to multi-point transition, producing a membership cloud, rather than a membership curve. Secondly, any particular drop of the cloud may be paid little attention to, however, the total shape of the cloud, which is visible, elastic, boundless and movable, is most important (see Fig. 1). That is why we use the terminology ``cloud" to name it. Thirdly, the mathematics expected curve (MEC) of a membership cloud may be considered as its membership function from the fuzzy set theory point of view. Finally, the definition has effectively integrated the fuzziness and randomness of a linguistic term in a unified way. In the cloud, fuzziness lies at the center, and there may be nothing to do with probability, but there is a probability adhered on the fuzziness from the statistical point of view. We can see the integrated uncertainty of fuzziness and randomness and the convergent properties of the cloud model. The cloud concept provides a means of both qualitative and quantitative characterization of linguistic terms.[3,4]

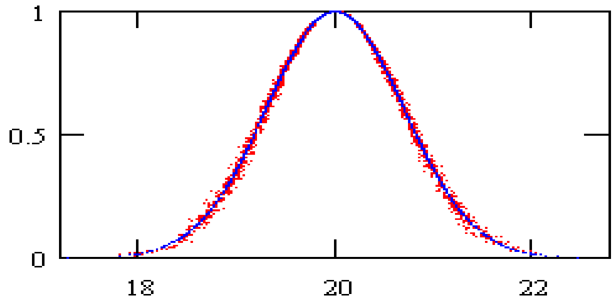


Fig.1 Normal Cloud with digital characteristic E_x=20 E_n=0.7 H_e=0.025

The bell-shaped clouds, called normal clouds are most fundamental and useful in representing linguistic terms. Since some people may get used to the concept of conventional membership functions, we could also use the normal membership function to represent the mathematical expected curve (MEC) of the cloud model. The digital parameters of a normal cloud characterizes the quantitative meaning of a linguistic atom. The Gaussian distribution transformation is used in a very effective way in characterizing normal clouds. A normal cloud is described with only three digital characteristics, expected value (Ex), entropy (En) and hyper entropy (He).

The expected value Ex of a cloud is the position at the universe of discourse, corresponding to the center of gravity of the cloud. In other words, the element Ex in the universe of discourse fully belongs to the linguistic term represented by the cloud model. The entropy, En, is a measure of the fuzziness of the concept over the universe of discourse showing how many elements in the universe of discourse could be accepted to the linguistic term. It should be noticed that the entropy defined here is a generic notion, and it need not be probabilistic. The entropy decreases as the MEC bandwidth decreases, Only if upon the narrowing cloud turns to be a precise numerical value is formed, the entropy becomes zero. Looking at the normal cloud in detail we see that its thickness is uneven. The hyper entropy, He, is a measure of the uncertainty of the entropy

En. Close to the waist of the cloud, corresponding to the center of gravity, cloud drops are most dispersed, while at the top and bottom the focusing is much better. The discrete degree of cloud drops depends on He.[3,4]

3.2 Cloud Generators

Given three digital characteristics E_x, E_n, and H_e, to represent a linguistic term, a set of cloud drops may be generated by the following algorithm:

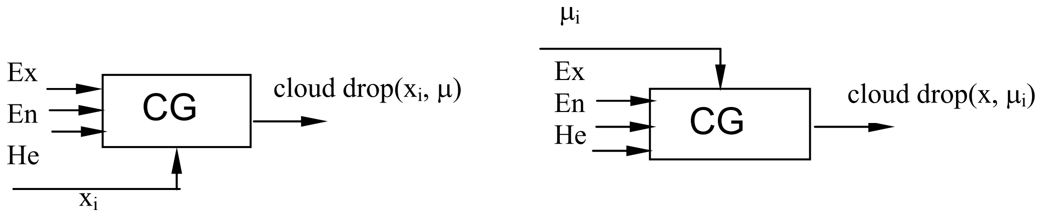
Algorithm 1: Normal Cloud Generation

Input: *the expected value of cloud E_x, the entropy of cloud E_n,*
the hyper entropy of cloud H_e, the number of drops N.

Output: *a normal cloud with digital characteristics E_x, E_n, and H_e.*

1) *Produce a random value x which satisfies with the normal distribution probability of mean = E_x, and standard error = E_n;*
2) *Produce a random value E_n' which satisfies with the normal distribution probability of mean = E_n, and standard error = H_e;*
3) *Calculate* $y = e^{\frac{-(x-E_X)^2}{2(E_N')^2}}$;
4) *Let (x, y) be a cloud drop in the universe of discourse;*
5) *Repeat 1-4 until the number of drops required all generated.*

The idea of using only three digital characteristics to generate a cloud is creative. A series of linguistic term generators have been implemented both in hardware and software and are a patented invention in China [27]. The generator could produce as many drops of the cloud as you like (Fig. 1). This kind of generators is called a forward cloud generator. All the drops obey the properties described above. Cloud-drops may also be generated upon conditions (see Fig. 2). It is easy to set up a half-up or half-down normal cloud generator with the similar strategy, if there is a need to represent such a linguistic term.



(a) On the condition of x_i

(b) On the condition of "μ_i"

Fig. 2 Generators on condition

It is natural to think about the generator mechanism in an inverse way. Given a number of drops, as samples of a normal cloud, the three digital characteristics Ex, En, and He could be obtained to represent the corresponding linguistic term. This kind of cloud generators may be called backward cloud generators. It is easy to see that some approximation has to be made if only a few drops are given. Of course, the more drops, the more accurate to the generated Ex, En, and He. The robustness of the backward cloud generators in our experiments is also very promising if there are some noisy drops mixed in. Since the cloud model represents linguistic terms, the forward and backward cloud generators can be served interchangeably to bridge the gap between quantitative and qualitative knowledge[3,4].

3.3 Construction of Qualitative Rules Using Cloud Models

We may immediately use two forward cloud generators to construct a qualitative rule, ``If A then B," if the digital characteristics of the linguistic terms A and B in that rule are given. See Figure 3, in which, the membership degree, μ, produced by an input x to the generator CGA represents the activated strength of the rule which goes to control the generator CGB to produce a set of drops quantitatively.

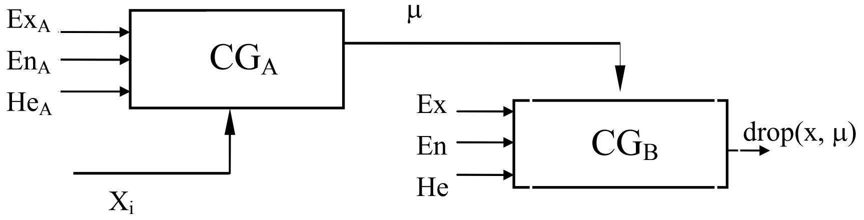


Fig. 3 A qualitative rule implemented by cloud generators

4 Classification with Cloud Models

4.1 Reduction of Classification Complexity and Softening Thresholds

In KDD-related problems, the universe U is finite and is highly desirable for it to be small. Only finite classifications are "learnable," i.e., we can potentially acquire complete knowledge about such classifications. Unfortunately, most finite classifications are not learnable due to the excessively large number of possible equivalence classes. Only a small fraction of all possible classifications expressible in terms of the indiscernibility relation are learnable.

To evaluate the computational tractability of the finite classification learning problem, we adopt the notion proposed by Ning Shan in [9]—classification complexity, defined as the number of equivalence classes in the classification. In practice, this number is usually not known in advance. Instead, a crude upper bound on the classification complexity for a subset of attributes B⊆C, can be computed "a priori" by the following fomula:

$$TC(B,V) = \prod_{p \in B} card(V_p)$$

The quantity TC(B,V) is called the theoretical complexity of the set of attributes B given the set of values V of the attributes B. If the number of attributes and the size of the domain Vp for each attribute is large, then TC(B,V) grows exponentially large. It is very difficult to find a credible classification based on a large number of attributes unless the attributes are strongly dependent (e.g., functionally dependent) on each other (limiting the number of equivalence classes).

Complexity reduction increases the credibility of the classification by generalizing condition attributes. The information generalization procedure applies attribute-oriented concept tree ascension [5,6,7,8] to reduce the complexity of an information system. It generalizes a condition attribute to a certain level based on the attribute's concept tree, which is provided by knowledge engineers or domain experts. Trivially, the values for any attribute can be represented as a one-level concept tree where the root is the most general value "ANY" and the leaves are the distinct values of the attribute. The medium level nodes in the concept tree with more than two levels are qualitative terms, which are expressed in cloud models (see Fig. 3). The data corresponds to higher level nodes must cover the data corresponds to all the descendant nodes. The transformations between qualitative terms and quantitative values of condition attributes are implemented through cloud models.

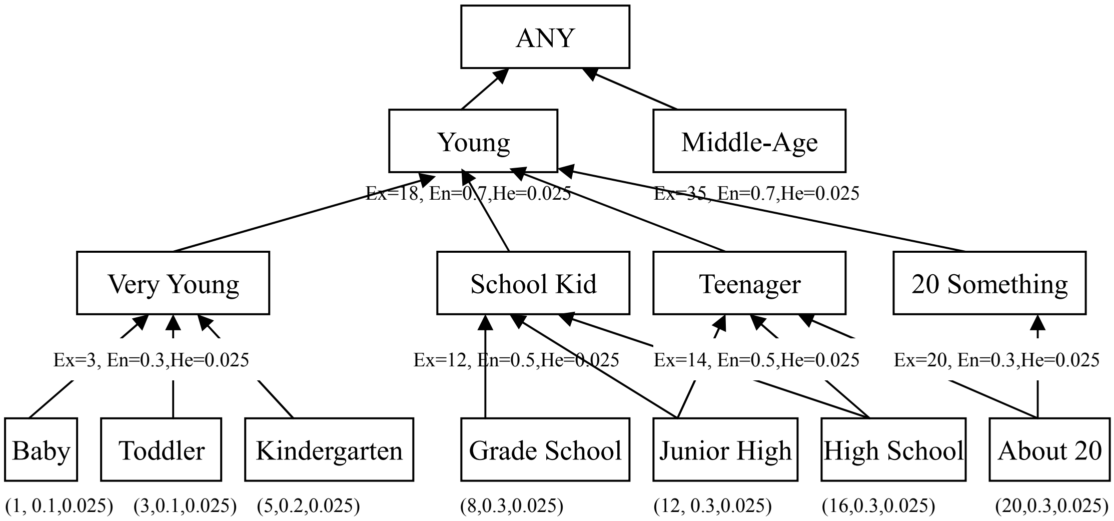


Fig. 3 Concept Tree with Qualitative Terms

We modified the algorithm proposed by Shan et al. In [9], which extracts a generalized information system. In this algorithm, there are two important concepts — the attribute threshold and the theoretical complexity threshold, which constrain the generalization process. Since the exact values of the thresholds are very hard to determine, we apply the linguistic terms to soften them in our modified algorithm. The two linguistic terms are represented with cloud models. Since the thresholds are not exact values, we call them *soft thresholds*. The entropy of soft thresholds can effectively control cycle numbers. This is a novel contribution of this paper.

Condition attributes are generalized by ascending their concept trees until the number of values for each attribute is less than or equal to the user-specified soft attribute threshold for that attribute and the theoretical complexity of all generalized attributes is less than or equal to the user-specified soft theoretical complexity threshold. For each iteration, one attribute is selected for generalization (this selection can be made in many ways. Lower level concepts of this attribute are replaced by the concepts of the next higher level. The number of possible values at a higher level of an attribute is always smaller than at a lower level, so the theoretical complexity is reduced. [5,6,7,8,9]

Algorithm 2: Reduction of Classification Complexity

```
Input: (1)The original information system S with a set of condition attributes $C_i$ ($1 \le i \le n$);
        (2) a set of H of concept trees, where each $H_i \in H$ is a concept hierarchy for the attribute $C_i$.
        (3)$S_{ti}$ is a soft threshold for attribute $C_i$ with digital characteristic ($Ex_{ti}$, $En_{ti}$, $He_{ti}$) and $d_i$ is
          the number of distinct values of attribute $C_i$;
        (4) $S_{TC}$ defined by user is a soft theoretical complexity threshold with digital characteristic
          ($Ex_{tc}$, $En_{tc}$, $He_{tc}$).
Output: The generalized information system S'
S' ← S
$TC_1 = \prod_{i=1}^{n} d_i$
Generate soft threshold values $S_{TC}$ and $S_{ti}$
while $TC_1 > S_{TC}$ and $\exists d_i > S_{ti}$ do
    Select an attribute $C_i \in C$ such that $d_i / S_{ti}$ is maximal
    Ascend tree $H_i$ one level and make appropriate substitutions in S'
    Remove duplicates from S'
    Recalculate $d_i$
    Recalculate $TC_1 = \prod_{i=1}^{n} d_i$
    Regenerate soft threshold values $S_{TC}$ and $S_{ti}$

Endwhile
```

4.2 Quality of Classification

Each combination of values of the decision attribute is a concept. Our main goal is to identify a credible classification for each such concept F∈R(D), based on some interacting attributes B. To evaluate the quality of the classification R*(B) with respect to the concept F, we use the following criterion[11]:

$$Q_B(F) = \beta \sum_{E \in R^*(B)} P(E) \times \left| P(F|E) - P(F) \right| \qquad \text{and} \quad \beta = \frac{1}{2P(F)(1-P(F))}$$

This criterion represents the average gain in the quality of information, reflected by P(F|E), used to make the classificatory decision F versus ¬F. In the absence of the classification R*(B), the only available information for this kind of the decision is the occurrence probability P(F). The quantity β is a normalization factor to ensure that Q_B is always within the range [0,1], with 1 corresponding to the exact characterization of the concept (that is, when for every equivalence class E, P(F|E) is either 0 or 1) and 0 corresponding to the situation where the distribution of F within every equivalence class E is the same as in the universe U.[9]

4.3 Identifying Interacting Attributes

The local discovery of interacting attributes has been reported in [10,11]. All condition attributes are grouped into disjoint clusters without considering the decision attribute(s). Each cluster contains attributes that are directly or indirectly dependent upon each other. In[9], a global discovery of interacting attributes is reported. The global discovery method selects a subset of condition attributes that are based on their relevance to the decision attribute(s). Here, we adopt the global generalization algorithm for attribute clusters.

Algorithm 3: Identifying Interacting Attributes

```
Input: C is a set of condition attributes
       D is a set of decision attributes, and
       $\gamma$ is a soft dependency threshold with digital characteristic (Ex, En, He)
Output: AttriCluster is a set of attribute's clusters.
AttriCluster $\leftarrow a \in C$
$C \leftarrow C - \{a\}$
Dep $\leftarrow$ DEP{AttriCluster, D}
Generate soft dependency threshold value $\gamma$
While $C \neq \emptyset$ and Dep $< \gamma$ do
    Forall attribute $a \in$ AttriCluster do
        C' $\leftarrow$ AttriCluster $\cup \{a\}$
        $Dep_a \leftarrow$ DEP(C',D)
    Endfor
Find the attribute x that has the maximum value of $Dep_a$
AttriCluster $\leftarrow$ AttriCluster $\cup \{x\}$
$C \leftarrow C - \{x\}$
Dep $\leftarrow$ DEP(AttriCluster, D)
Regenerate soft dependency threshold value $\gamma$
Endwhile
```

DEP is a generalization of the concept quality measure Q_B. DEP(X,Y) measures degree of dependency between two groups of attributes X and Y:

$$DEP(X,Y) = \sum_{E \in R^*(Y)} P(E) Q_X(E)$$

4.4 Search for Classifications

After reduction of complexity and identifying interacting attributes, we search for credible classifications of the database tuples based on some selected interacting attributes. A classification is credible if it is complete or almost complete with respect to the domain from which the database was collected. Here, we adopt the SLIQ(Supervised Learning In Quest) method, which was developed by Mehta et al.[12,13]. It is a supervised learning method that constructs decision trees from a set of examples. It uses a novel pre-sorting technique in the tree growing phase. This sorting procedure is integrated with a breadth-first tree growing strategy to enable classification of disk-resident datasets. SLIQ also uses a new tree-pruning algorithm that is inexpensive, and results in compact and accurate trees. Since we can interchange between qualitative and quantitative representation, the supervising process is much easier, and the results are robust. The combination of these techniques enables it to scale for data sets with many attributes and classify data sets irrespective of the number of classes, attributes, and examples.

4.5 Experiments and Results

In this section, we discuss the domain of acute abdominal pain, focusing on the models used for the diagnosis, which will test and verify our model and algorithms. The most serious common cause of acute abdominal pain is appendicitis, and in many cases a clear diagnosis of appendicitis is difficult, since other diseases such as Non-Specific Abdominal Pain (NSAP) can present similar signs and symptoms (findings). The tradeoff is between the possibility of an unnecessary appendectomy and a perforated appendix, which increases mortality rates five-fold. The high incidence of acute abdominal pain coupled with the poor diagnosis accuracy, make any improvements in diagnostic accuracy significant.

The abdominal pain data used for this study consists of 10270 cases, each with 169 attributes. The class variable, final diagnosis, has 19 possible values, and the variables have a number of values ranging from 2 to 32 values. The resulting database addresses acute abdominal pain of gynaecological origin, based on case-notes for patients of reproductive age admitted to hospital, with no recent history of abdominal or back pain. In compiling the database, the first 202 cases were used in the design of the database itself; thus, they cannot be used for the purpose of testing any model. Moreover, out of the 10270 cases, the diagnosis of only 8950 cases was definitely known (definite diagnoses); the remaining 1320 cases were assigned the best possible diagnosis, as a presumed diagnosis. Finally, 120 patients occur more than once in the database.

Class Variable	Cloud Models Method	C4.5	Expert Diagnosis
Appendicitis	3707	3269	3770
Stomach disease	3025	2750	3108
Liver diseases	636	567	669
Spleen diseases	304	288	310
Gallbladder diseases	247	243	235
Small intestine diseases	236	235	240
Large intestine diseases	225	229	224
Uterus diseases	221	220	212
Kidney diseases	211	199	214
Gallstone	163	167	180
Duodenitis	118	139	145
Colonitis	159	168	165
Caecitis	138	150	156
Rectitis	166	184	187
Alimentary intoxication	134	141	145
Acid intoxication	77	84	87
Parcreatitis	54	61	68
Intimitis	69	82	83
Other diseases	380	1094	72

Table 1. Classification Results for Acute Abdominal Pain

Our results show that 2 of 19 classes accounted for almost 67% of the cases, whereas each of the other classes accounted for 7% or less of the cases. For each of the 2 most common classes, since the probability distribution was induced from many cases, our model was significantly better than C4.5 methods (shown as table 1), correctly classifying about 89% of the cases.

On the other hand, on the cases involving the other 17 classes, C4.5 classifier performed better than the cloud models approach (not significantly better). This because the cloud models could not accurately estimate the complicated distributions from so few cases, leading to poor predictive accuracy.

These results offer some insights into the cloud models. In complex domains with many attributes, such as the abdominal pain domain, feature selection may play a very important part in classifiers for diagnosis; this is especially true when the data set is relatively small. In such cases, it is difficult to accurately acquire classification rules for the larger data set. Moreover, in domains where there are sufficient cases (as for the two main classes in the abdominal pain data set), cloud models method plays very well since they can easily model attribute dependencies. However, if the number of cases is small, then the simple decision tree method may perform better.

5 Conclusion

Data classification is a well-recognized operation in data mining and knowledge discovery research field and it has been studied extensively in statistics and machine learning literature. We described a novel approach to search for domain classification. The goal of the search is to find a classification or classifications that jointly provide a good, in the qualitative terms' sense, approximation of the interest. We have presented a new mathematical representation of qualitative concepts—Cloud Models. With the new models, mapping between quantities and qualities becomes much easier and interchangeable. Based on them, we introduced the concept of soft threshold and concept tree with qualitative terms. We also developed algorithms for clouds generation, complexity reduction, and identifying interacting attributes, etc. After classification search, further steps in the qualitative approach to knowledge discovery involve classification analysis and simplification, rule induction and prediction, if required by any application. These aspects will require a lot of work, and have been omitted here, as they will be presented in detail in other publications.

References

[1] U. Fayyad, G. Piatetsky-Shapiro And P. Smyth, "Knowledge Discovery and Data Mining: Towards a Unifying Framework," In Proceedings of KDD-96: Second International Conference on Knowledge Discovery & Data Mining, Menlo Park, CA: AAAI Press, 1996 , p. 82-88.

[2] J.R. Quinlan , "Induction of decision trees," Machine Learning, Vol. 1, 1986, pp81-106.

[3] Li Deyi，Shi Xuemei, Meng Haijun. "Membership clouds and clouds generators", The Research and Development of Computers,1995,42(8):32-41.

[4] D.Li, X.Shi,P.Ward and M.M.Gupta, "Soft Inference Mechanism Based on Cloud Models,"

in Proceedings of the First International Workshop on Logic Programming and Soft Computing, Edited by Francesca Arcelli Fontana, Ferrante Formato and Trevor P. Martin, Bonn, Germany, Sept 6, 1996, p.38-62.

[5] Y. Cai, N. Cercone, and J. Han, "Attribute-Oriented Induction in Relational Databases," In Knowledge Discovery in Database, 1991, p.213-228.

[6] J. Han and Y. Fu, "Exploration of the Power of Attribute-Oriented Induction in Data Mining," Advances in Knowledge Discovery and Data Mining, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, eds., AAAI/MIT Press, 1996, p.399-421.

[7] J.Han, Y. Cai, N. Cercone, "Knowledge Discovery in Databases: An Attribute-Oriented Approach", in Pro. 18th International Conference on Very Large Databases, Aug. 1992, p.547-559.

[8] J. Han and Y. Fu, "Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases," Pro. AAAI'94 Workshop Knowledge Discovery in Databases, Seattle, July 1994, p.157-168.

[9] Shan,N., Ziarko, W., Hamilton, H.J., and Cercone, "Discovery Classification Knowledge in Databases Using Rough Sets", In Proceedings of KDD-96: Second International Conference on Knowledge Discovery & Data Mining, Menlo Park, CA: AAAI Press, 1996.

[10] Shan,N., Ziarko, W., Hamilton, H.J., and Cercone, "Using Rough Sets as Tools for Knowledge Discovery". In Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada, 1995.

[11] Ziarko, W. And Shan, N. "On Discovery of Attribute Interactions and Domain Classifications", In Lin.T.Y eds., Special Issue in Journal of Intelligent Automation and Soft Computing, 1995.

[12] Rakesh Agrawal , Manish Mehta , John Shafer and Ramakrishnan Srikant, "The Quest Data Mining System", In Proceedings of KDD-96: Second International Conference on Knowledge Discovery & Data Mining, Menlo Park, CA: AAAI Press, 1996, p.244-249.

[13] M. Mehta, R. Agrawal, and J. Rissanen, "A fast scaleable classifier for data mining," In Proc. of the Fifth Int'l Conference on Extending Database Technology, 1996.

[14] J.R.Quinlan, "C4.5: Programs for Machine Learning," Morgan Kaufmann,1993.

[15] G. Piatetsky-Shapiro, "Discovery, analysis, and presentation of strong rules", In G. Piatetsky-Shapiro and W. J. Frawley, editors, Knowledge Discovery in Databases, AAAI/MIT Press,1991, p.229-238.

[16] J. Han, Y. Cai, and N. Cercone, "Data-driven discovery of quantitative rules in relational databases", IEEE Trans. Knowledge and Data Engineering, Vol. 3, 1993, p.29-40.

[17] A.Agrawal, S.ghosh, T. Imielinkski, B. Iyer, and A. Swami, "An Interval Classifier for Database Mining Applications," Proceedings of the 18th International Conference on Very Large Data Bases, August 1992, p.560-573.

[18] Pawlak, Z., Rough Sets: Theoretical Aspects of Reasoning About Data, Kluwer,1991.

[19] Deyi Li, Jiawei Han, Xuemei Shi, and Man Chung Chan, "Knowledge representation and discovery based on linguistic atoms", Knowledge-Based Systems, Elsevier Science B.V. 10(1998):431-440.

[20] Jianhua Fan, Deyi Li, "An Overview of Data Mining and Knowledge Discovery", the Journal of Computer Science and Technology, No.4, 1998.