Data Reduction Based on Spatial Partitioning

Gongde Guo, Hui Wang, David Bell, and Qingxiang Wu

School of Information and Software Engineering, University of Ulster
Newtownabbey, BT37 0QB, N.Ireland, UK
{G.Guo, H. Wang, DA.Bell, Q.Wu}@ulst.ac.uk

Abstract. The function of data reduction is to make data sets smaller, while preserving classification structures of interest. A novel approach to data reduction based on spatial partitioning is proposed in this paper. This algorithm projects conventional database relations into multidimensional data space. The advantage of this approach is to change the data reduction process into a spatial merging process of data in the same class, as well as a spatial partitioning process of data in different classes, in multidimensional data space. A series of partitioned regions are eventually obtained and can easily be used in data classification. The proposed method was evaluated using 7 real world data sets. The results were quite remarkable compared with those obtained by C4.5 and DR. The efficiency of the proposed algorithm was better than DR without loss of test accuracy and reduction ratio.

1 Introduction

Data reduction is a process used to transform raw data into a more condensed form without losing significant semantic information. In data mining, data reduction in a stricter sense refers to feature selection and data sampling [1], but in a broader sense, data reduction is regarded as a main task of data mining [2]. Data mining techniques can thus, in this broad sense, be regarded as a method for data reduction. Data reduction is interpreted as a process to reduce the size of data sets while preserving their classification structures. Wang, *et al* [3] propose a novel method of data reduction based on lattices and hyper relations. The advantage of this is that raw data and reduced data can be both represented by hyper relations. The collection of hyper relations can be made into a complete Boolean algebra in a natural way, and so for any collection of hyper tuples its unique least upper bound (lub) can be found, as a reduction.

According to the method proposed in [3], the process of data reduction is to find the least upper bound of the raw data and to reduce it. The process of data reduction can be regarded as a merging process of simple tuples (raw data) and hyper tuples that have been generated in the same class to generate new hyper tuples. The success of each merging operation depends on whether the new hyper tuple generated from this merging operation covers in same sense a simple tuple of another class. If the new hyper tuple does cover a simple tuple of another class, the operation is cancelled. The merging operation repeats recursively until all the data including hyper tuples and same class simple tuples cannot be merged again. The main drawback of the method proposed in [3] is its efficiency, since much time is spent in trying probable merge. In this paper, we introduce the *complementary* operation of hyper tuples and attempt to

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2074, pp. 245–252, 2001.

use the irregular regions to represent reduced data. The main goal of the proposed algorithm is to improve its efficiency and reduction ratio whilst preserving its classification accuracy.

The remainder of the paper is organized as follows. Section 2 introduces the definitions and notation. Section 3 describes the data reduction and classification algorithm based on spatial partitioning, in which the execution process of the algorithm is demonstrated by graphical illustration. The experimental results are described and the evaluation is given in Section 4. Section 5 ends the paper with a discussion and an indication of proposed future work.

2 Definitions and Notation

In the context of the paper, *Hyper relations* are a generalization of conventional database relations in the sense that it allows sets of values as tuple entries. A *hyper tuple* is a tuple where entries are sets instead of single values. A hyper tuple is called a *simple tuple*, if all its entries have a cardinality of 1. Obviously a simple tuple is a special case of hyper tuple.

Consider two points p_i, p_j denoted as $p_i=(p_{i1}, p_{i2}, \ldots, p_{in})$, $p_j=(p_{j1}, p_{j2}, \ldots, p_{jn})$ and two spatial regions a_i, a_j denoted as $a_i=([t_{i11}, t_{i12}], [t_{i21}, t_{i22}], \ldots, [t_{in1}, t_{in2}])$, $a_j=([t_{j11}, t_{j12}], [t_{j21}, t_{j22}], \ldots, [t_{jn1}, t_{jn2}])$ in multidimensional data space, in which $[t_{il1}, t_{il2}]$ is the projection of a_i to its *l-th* component, and $t_{il2} \geq t_{il1}$, $l=1, 2, \ldots, n$. In this paper, for simplicity and uniformity, any point p_i is represented as a spatial region in multidimensional data space, viz. $p_i=([p_{i1}, p_{i1}], [p_{i2}, p_{i2}], \ldots, [p_{in}, p_{in}])$. This is often a convenient and uniform representation for analysis.

Definition 1 Given two regions a_i, a_j in multidimensional data space, the *merging operation* of two regions denoted by '$\cup$' can be defined as: $a_i \cup a_j=([\min(t_{i11}, t_{j11}), \max(t_{i12}, t_{j12})], [\min(t_{i21}, t_{j21}), \max(t_{i22}, t_{j22})], \ldots, [\min(t_{in1}, t_{jn1}), \max(t_{in2}, t_{jn2})])$.

The *intersection operation* '$\cap$' of two regions in multidimensional data space can be defined as:

$a_i \cap a_j =([\max(t_{i11}, t_{j11}), \min(t_{i12}, t_{j12})], [\max(t_{i21}, t_{j21}), \min(t_{i22}, t_{j22})], \ldots, [\max(t_{in1}, t_{jn1}), \min(t_{in2}, t_{jn2})])$. $a_i \cap a_j$ is empty, if and only if there exists a value of l such that $\max(t_{il1}, t_{jl1}) > \min(t_{il2}, t_{jl2})$, where $l=1,2,\ldots, n$.

A point merging (or intersecting) with a region can be regarded as a special case according to above definition.

Definition 2 Given a region a_j in multidimensional data space denoted as $a_j = ([t_{j11}, t_{j12}], [t_{j21}, t_{j22}], \ldots, [t_{jn1}, t_{jn2}])$, the *complementary operation* of a_j is defined as: $\overline{a_j} = (\overline{[t_{j11}, t_{j12}]}, [t_{j21}, t_{j22}], \ldots, [t_{jn1}, t_{jn2}]) \cup ([t_{j11}, t_{j12}], \overline{[t_{j21}, t_{j22}]}, \ldots, [t_{jn1}, t_{jn2}]) \cup \ldots \cup ([t_{j11}, t_{j12}], [t_{j21}, t_{j22}], \ldots, \overline{[t_{jn1}, t_{jn2}]}) \cup (\overline{[t_{j11}, t_{j12}]}, \overline{[t_{j21}, t_{j22}]}, [t_{j31}, t_{j32}], \ldots, [t_{jn1}, t_{jn2}]) \cup ([t_{j11}, t_{j12}], \overline{[t_{j21}, t_{j22}]}, \overline{[t_{j31}, t_{j32}]}, \ldots, [t_{jn1}, t_{jn2}]) \cup \ldots \cup ([t_{j11}, t_{j12}], \ldots, \overline{[t_{j(n-1)1}, t_{j(n-1)2}]}, \overline{[t_{jn1}, t_{jn2}]}) \cup \ldots \cup (\overline{[t_{j11}, t_{j12}]}, \overline{[t_{j21}, t_{j22}]}, \ldots, \overline{[t_{jn1}, t_{jn2}]})$, is the region in the multidimensional data space complementary region a_j.

Definition 3 Given a point p_i denoted as $p_i=(p_{i1}, p_{i2}, \ldots, p_{in})$ and a region a_j denoted as $a_j=([t_{j11}, t_{j12}], [t_{j21}, t_{j22}], \ldots, [t_{jn1}, t_{jn2}])$ in multidimensional data space, the *hyper similarity* of p_i, a_j denoted as $S(p_i, a_j)$ is defined as follows:

If a_j is a regular region, a_j=([t_{j11}, t_{j12}], [t_{j21}, t_{j22}], …, [t_{jn1}, t_{jn2}]), the hyper similarity $S(p_i, a_j)$ is equal to the number of l which satisfies $t_{jl1} \le p_{il} \le t_{jl2}$, in which $l=1, 2, \ldots, n$.

If a_j is an irregular region, consisting of h regular regions, denoted as $a_j = \{ a_{j1}, a_{j2}, \ldots, a_{jh} \}$, the *hyper similarity* $S(p_i, a_j)$ equals the value of max($S(p_i, a_{j1})$, $S(p_i, a_{j2})$, …, $S(p_i, a_{jh})$).

Definition 4 Given a point $p_i = (p_{i1}, p_{i2}, \ldots, p_{in})$ and a region a_j in multidimensional data space, the universal *hyper relation* '≤' is defined as: $p_i \le a_j$, if and only if the point p_i falls into the spatial region of a_j.

If a_j is a regular region denoted as a_j =([t_{j11}, t_{j12}], [t_{j21}, t_{j22}], …, [t_{jn1}, t_{jn2}]). $p_i \le a_j$ if for all values of l, $t_{jl1} \le p_{il} \le t_{jl2}$, where $l=1, 2, \ldots, n$.

If a_j is an irregular region, consisting of h regular regions, $a_j = \{ a_{j1}, a_{j2}, \ldots, a_{jn} \}$. $p_i \le a_j$, if and only if there exists a regular region a_{jl} where $p_i \le a_{jl}$, in which, $l=1,2,\ldots, h$.

For simplicity, all the data attributes used in this paper for data reduction and classification are numerical. Set union operation (respectively intersection and complementation operations) can be used to replace the '$\cup$' operation ('$\cap$' and '-' operations respectively) defined above for categorical data or binary data. In addition, the standard set inclusion operation and subset operation can be used to replace the hyper similarity operation and universal hyper relation operation respectively.

3 Data Reduction & Classification Algorithm

Let a training data set D_t=$\{d_1, d_2, \ldots, d_m\}$, where $d_i = (d_{i1}, d_{i2}, \ldots, d_{in})$. d_i is represented as d_i=([d_{i1}, d_{i1}], [d_{i2}, d_{i2}], …, [d_{in}, d_{in}]) using spatial region representation as a point in multidimensional data space. Supposing that there are k classes in the training data set and d_{in} is a decision attribute, value $d_{in} \in \{t_1, t_2, \ldots, t_k\}$, the spatial partitioning algorithm is as follows:

1. $t=0$
2. $M_i^t = \bigcup_{d_{jn}=t_i} \{d_j\}$, $d_j \in D_t$, $i=1, 2, \ldots, k$, $j=1, 2, \ldots, m$
3. $M_{i,j}^t = M_i^t \cap M_j^t$, $i \ne j$, $i=1, 2, \ldots, k$, $j=1, 2, \ldots, k$
4. $S_i^t = M_i^t \cap \overline{M_{i,1}^t} \cap \ldots \cap \overline{M_{i,j-1}^t} \cap \overline{M_{i,j+1}^t} \cap \ldots \cap \overline{M_{i,k}^t}$, $i=1, 2, \ldots, k$
5. $D_{t+1} = \{d_i \mid d_i \in M_{i,j}^t, i \ne j, i=1, 2, \ldots, k, j=1, 2, \ldots, k\}$
6. If $(D_{t+1}=\phi)$ go to 8
7. $t=t+1$, go to 2
8. $R_i = \{ S_i^0, S_i^1, \cdots, S_i^t \}$, $i=1, 2, \ldots, k$.

Some symbols used in the above algorithm are: S_i^t -the biggest irregular region of t_i class obtained in the t-th cycle; D_t -the training data set used in the t-th cycle; M_i^t - the merging region of t_i class data obtained in the t-th cycle; $M_{i,j}^t$ - the intersection of M_i^t and M_j^t obtained in the t-th cycle. R_i - the results obtained via running the algorithm are a series of distributive regions of t_i class data, in which, $i=1, 2, \ldots, k$.

Given the training data set shown in Figure 2-1, the running process of the algorithm in 2-dimensional space is illustrated graphically below.

The training data set includes 30 data points and is divided into three classes of black, grey and white. The distribution of data points in 2-dimensional data space is shown in Figure 2-1.

At the beginning of the running process, we merge all the data in the same class for each individual class and gain three data spatial regions M_1^0, M_2^0, M_3^0 represented by bold line, fine line and broken line respectively. The intersections of $M_{1,2}^0, M_{1,3}^0, M_{2,3}^0$ are also represented by bold line, fine line and broken line in Figure 2-2 and Figure 2-3, in which, $M_{1,2}^0 = M_1^0 \cap M_2^0$, $M_{1,3}^0 = M_1^0 \cap M_3^0$, $M_{2,3}^0 = M_2^0 \cap M_3^0$. In the first cycle, three partitioning regions S_1^0, S_2^0, S_3^0 shown in Figure 2-4 are obtained.

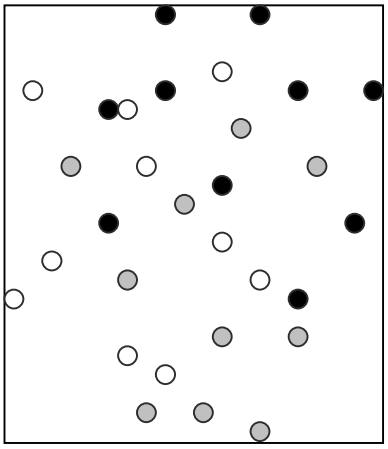


Fig. 2-1. The distribution of data points

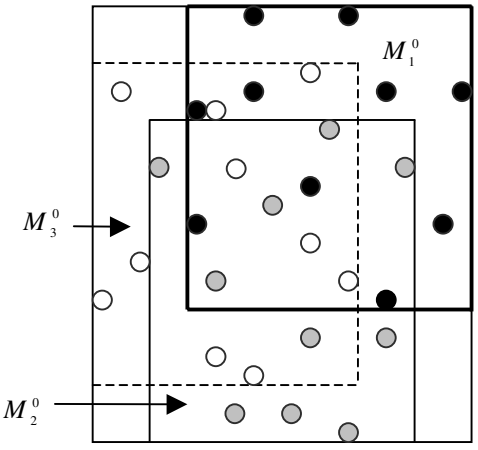


Fig. 2-2. Three merging regions

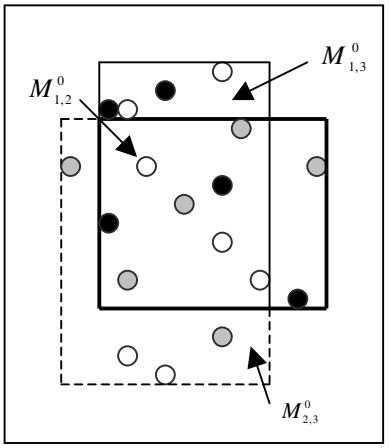


Fig. 2-3. Three intersection regions

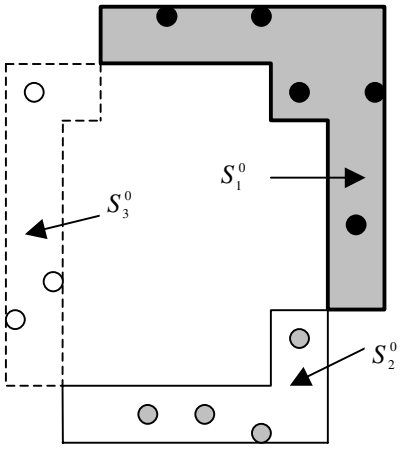


Fig. 2-4. Three partitioning regions

In which, $S_1^0 = M_1^0 \cap \overline{M_{1,2}^0} \cap \overline{M_{1,3}^0}$, $S_2^0 = M_2^0 \cap \overline{M_{1,2}^0} \cap \overline{M_{2,3}^0}$, $S_3^0 = M_3^0 \cap \overline{M_{1,3}^0} \cap \overline{M_{2,3}^0}$. Obviously, if test data falls into S_1^0 (or S_2^0, S_3^0), it belongs to the black class (or the grey, white class respectively). If it falls into none of S_1^0, S_2^0 and S_3^0, it should fall into $M_{1,2}^0$, or $M_{1,3}^0$ or $M_{2,3}^0$. If so, it can not be determined which class it belongs to. All the data in the original data set which belong to $M_{1,2}^0$ or $M_{1,3}^0$ or $M_{2,3}^0$ are taken out and form a new training data set. This new training data set is partitioned again and another three merging regions: M_1^1, M_2^1, M_3^1 as well as another three intersection regions: $M_{1,2}^1, M_{1,3}^1, M_{2,3}^1$ are obtained in the second cycle. The process of merging and partitioning is executed recursively until there is no data in the new training set. This process is illustrated graphically below from Figure 2-4 to Figure 2-9.

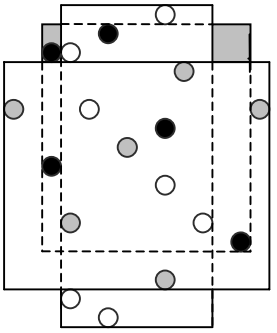


Fig. 2-5

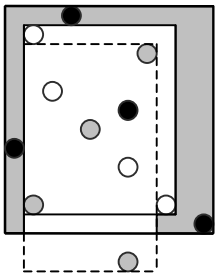


Fig. 2-6

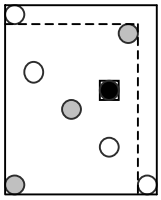
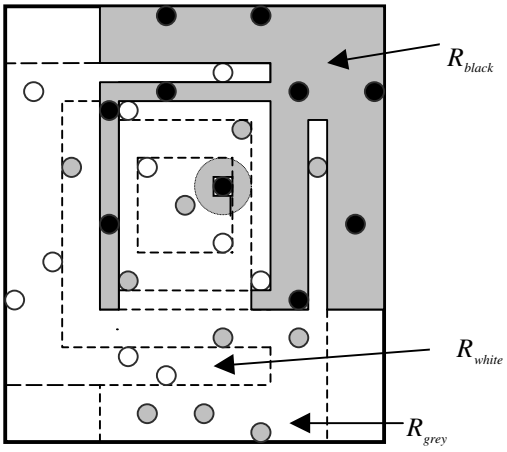


Fig. 2-7



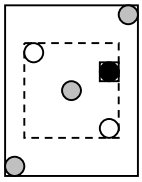


Fig. 2-8

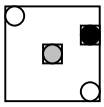


Fig. 2-9

Fig. 2-10. The distributive regions of black data

A series of distributive regions of data of each class are obtained via learning from the training data set. The distributive regions of data of the black class are represented against a grey background in Figure 2-10.

Using irregular regions representing spatial distributive regions of different classes can give higher data reduction efficiency. The term 'irregular regions' in this paper means the projection of the spatial region to each dimension might be a series of discrete intervals.

It is probable that the partitioning process cannot continue for some data distributions because equal merging regions could be obtained in the partitioning process. See Figure 3-1 for instance. The three merging regions of M_{black}, M_{grey} and M_{white} are totally equal to each other.

In this situation, one resolution is to select an attribute as a partitioning attribute to divide the data set into two subsets and then for each subset according to the above algorithm, continue partitioning until all the data has been partitioned.

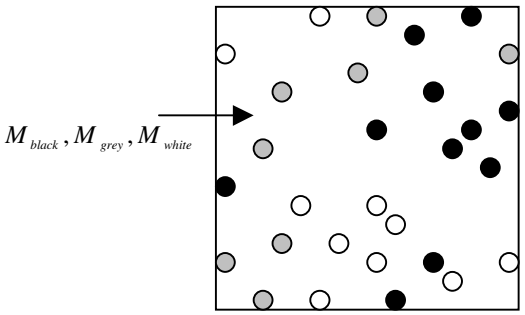


Fig. 3-1. Special training data set

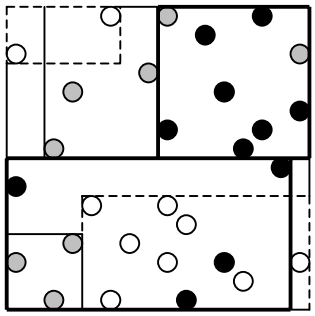


Fig. 3-2. Dividing and partitioning

Figure 3-1 shows that the partitioning cannot continue because $M_{black} = M_{grey} = M_{white}$. What we can do is to divide the data set into two subsets on the Y-axis. We then execute the spatial partitioning operation for each subset respectively shown in Figure 3.2 to let the partitioning process continue. A series of distributive regions obtained from the partitioning process can easily be used in classification.

Given a testing data t, R_i is a set of distributive regions of t_i class obtained by running the partitioning algorithm. We use universal hyper relation $\leq$ to classify the testing data t, the partitioning algorithm is as follows:

- If $t \leq R_i$ viz. t falls into the regions of R_i, then t is classified by the class of R_i.
- If there is not such a region of R_i which can satisfy $\leq$ operation, the class of t can be classified by using hyper similarity defined above, viz. t is classified by the class of R_j, where j is defined as $S(t, R_j)=\max(S(t, R_1), S(t, R_2), \ldots, S(t, R_k))$.

A system called Partition&Classify or P&C for simplicity was developed using the proposed method, it can classify unlabeled testing data effectively. The algorithm was evaluated for classification using some real world data sets and the results are quite remarkable. The experimental results are reported in next section.

This sort of data reduction and classification is very helpful for large databases and data mining based on some of the following reasons [3]:

- It reduces the storage requirements of data used mainly for classification;
- It offers better understandability for the knowledge discovered;
- It allows feature selection and continuous attribute discretization to be achieved as by-products of data reduction.

4 Experiment and Evaluation

The ultimate goal of data reduction is to improve the performance of learning, hence the main goal of our experiment is set to evaluate how well our proposed method performs for data reduction and to calculate its accuracy of prediction and performance for some real world data sets. We use the 5-fold cross validation method to evaluate its prediction accuracy and compare the results obtained from experiment with some of standard data mining methods.

Seven public databases are chosen from the UCI machine learning repository. Some information about these databases is listed in Table 1.

Table 1. General information about the data sets

Data set	NA	NN	NO	NB	NE	CD
Aust	14	4	6	4	690	383:307
Diab	8	0	8	0	768	268:500
Hear	13	3	7	3	270	120:150
Iris	4	0	4	0	150	50:50:50
Germ	20	11	7	2	1000	700:300
TTT	9	9	0	0	958	332:626
Vote	18	0	0	18	232	108:124

In Table 1, the meaning of the title in each column is follows: NA-Number of attributes, NN-Number of Nominal attributes, NO-Number of Ordinal attributes, NB-Number of Binary attributes, NE-Number of Examples, and CD-Class Distribution.

We also selected the C4.5 algorithm installed in the Clementine' software package as our benchmark for comparison and the DR algorithm [3] as a reference to data reduction. A 5-fold cross validation method was used to evaluate the performance of C4.5, DR and the P&C algorithm, the classification accuracy and the data reduction ratio were obtained and shown in Table 2. The reduction ratio we used is defined as follows:

(*The number of tuples in the original data set - The number of the biggest irregular regions in the model*) / (*The number of tuples in the original data set*).

The experimental results in Table 2 show that P&C outperforms C4.5 with respect to the cross validation test accuracy for the data sets but Vote. For the data sets with more numerical attributes (e.g. Iris and Diab data sets) P&C excels in ratio of data reduction compared to DR while preserving the accuracy of classification. Both DR and P&C were tested on the same PC with Pentium(r) III Processor, experiments show that DR has the highest testing accuracy among the three tested algorithms and

P&C has more higher reduction ratio than DR. In particular, on average P&C is about 2 times faster than DR.

Table 2. A comparison of C4.5, DR and P&C in testing accuracy and reduction ratio (TA-Testing Accuracy, RR-Reduction Ratio).

Data set	TA:C4.5	TA:DR	TA:P&C	RR:DR	RR:P&C
Aust	85.2	87.0	86.9	70.6	69.1
Diab	72.9	78.6	77.4	68.6	71.1
Hear	77.1	83.3	82.5	74.1	74.2
Iris	94.0	96.7	96.7	94.0	97.1
Germ	72.5	78.0	77.2	73.1	73.2
TTT	86.2	86.9	86.1	81.5	80.7
Vote	96.1	87.0	86.3	99.1	98.8
Average	83.4	85.4	84.7	80.1	80.6

5 Conclusion

In this paper, we have presented a novel approach to data reduction and classification (and so data mining) based on spatial partitioning. The reduced data can be regarded as a model of the raw data. We have shown that data reduction can be viewed as a process to find the biggest irregular region to represent the data in the same class. It executes union, intersection and complement operations in each dimension using the projection of spatial regions in multidimensional data space and represents the raw data of the same class by the local biggest irregular regions to realize the goal of data reduction. A series of spatial regions obtained from the learning process can be used in classification. Further research is required into how to eliminate noise and resolve the marginal problem to improve testing accuracy as current P&C is sensitive to noise data and data in marginal areas has lower testing accuracy.

References

1. Weiss, S. M., and Indurkhya, N. (1997). *Predictive Data Mining*: *A Practical Guide.* Morgan Kaufmann Publishers, Inc.
2. Fayyad, U. M. (1997). Editorial. *Data Mining and Knowledge Discovery – An International Journal* 1(3).
3. Hui Wang, Ivo Duntsch, David Bell. (1998). *Data reduction based on hyper relations.* In proceedings of KDD98, New York, pages 349-353.
4. Duntsch,I., and Gediga, G. (1997). *Algebraic aspects of attribute dependencies in information systems*. Fundamenta Informaticae 29:119-133.
5. Gratzer, G. (1978). *General Lattice Theory*. Basel: Birkhauser.
6. Ullman, J. D. (1983). *Principles of Database Systems.* Computer Science Press, 2 edition.
7. Wolpert, D. H. (1990). *The relationship between Occam's Razor and convergent guessing*. *Complex Systerms* 4:319-368.
8. Gongde Guo, Hui Wang and David Bell. (2000). *Data ranking based on spatial partitioning*. In proceedings of IDEAL2000, HongKong, pages78-84. Springer-Verlag Berlin Heidelberg 2000.