Dimensionality Reduction for Unsupervised Data

M Dash, H Liu, J Yao
Department of Information Systems and Computer Science
National University of Singapore
Kent Ridge, Singapore 119260
{manoranj,liuh,yaojun}@iscs.nus.sg

Abstract

Dimensionality reduction is an important problem for efficient handling of large databases. Many feature selection methods can serve the purpose for supervised data in which each record is attached with a calss label. Little work has been done for dimensionality reduction for unsupervised data in which class information is not available. Principal Component Analysis (PCA) is often used. However, PCA creates new features or principal components which are functions of original features. It is difficult to obtain intuitive understanding of the data using the new features only. In this paper we are concerned with the problem of determining and choosing the important original features for *unsupervised data*. Our method is based on the observation that removing an irrelevant feature from the feature set may not change the underlying concept of the data, but not so otherwise. We propose an entropy measure for ranking features, and conduct extensive experiments to show that our method is able to find the important features without class informtion by testing the ranked features using different classifiers, and it compares well with a similar method (Relief) that requires class information. This work can be used for dimensionality reduction in many applications in which class labels are not readily known.

Keywords: Data Preprocessing, Unsupervised Data, Entropy.

1 Introduction

In real-world applications the size of a database is usually large. The largeness can be due to the excessive number of features (variables), the huge number of instances (records), or both. For data mining tools to work efficiently one can try to reduce the data size. In this paper we focus on reducing the size by decreasing the number of features. One approach is to determine the relative importance of the features and then select a subset of important features. This can be achieved in various ways, for example, by feature selection or feature extraction.

There are a number of feature selection methods that determine the relative importance of the features before selecting a subset of important ones. Examples are Relief [5, 7], POE+ACC [10]. A typical feature selection method tries to choose a subset of features from the original set that ideally is necessary and sufficient to describe the target concept [5]. Target concepts are denoted by class labels. Data with class labels are called supervised data; data without class labels unsupervised. Although original features are selected based on some evaluation function, *these methods require class labels*. Hence, feature selection methods largely fail for unsupervised data.

Feature extraction methods create new features which are uncorrelated and retain as much variation as possible in the database ([15], [4]). One commonly known method is principal component analysis (PCA). Although PCA does not need class labels for extracting the features or Principal Components (PCs), *usually the values of the original features are still needed in order to calculate the PCs since each PC is a function of the original features*. Also, it is difficult to get intuitive understanding of the data *using the extracted features only*.

In practice, we often encounter databases that are large and without class information. Data is normally collected for organizational or book-keeping purposes, and not particaularly for data mining. For example, transactional data does not contain any class information, and it could have a huge number of features. It is often desirable to preprocess the data before applying a knowledge discovery tool [14]. Dimensionality reduction without creating any new feature offers an effective solution to decreasing the data size. PCA or feature selection methods cannot help much for this type of dimensionality reduction due to the reasons given above. This is a new problem that surfaces as the need for data mining arises.

In this paper we specifically deal with this new problem, i.e., the problem of determining and choosing the important original features for unsupervised data. An entropy measure that determines the importance of the original features is introduced in the next Section. This measure works for both nominal and continuous data types. In Section 3 a sequential backward selection algorithm is given, and the problem of determining the threshold (number of features) is discussed. The experimental study in Section 4 shows that the proposed algorithm is able to find the important features for databases with known important features. It also shows that the performance of the proposed algorithm without using class labels is very close to and sometimes better than that of Relief-F [7] (a generalized version of Relief [5] for multiple class labels) which ranks the original features using the class labels The paper concludes in Section 5.

2 An Entropy Measure for Continuous and Nominal Data Types

In this section we introduce an entropy measure for determining the relative importance of variables. This measure is applicable to both nominal and continuous data types, and does not need class information to evaluate the variables unlike some other entropy measures, for example, information gain measure in ID3 [12].

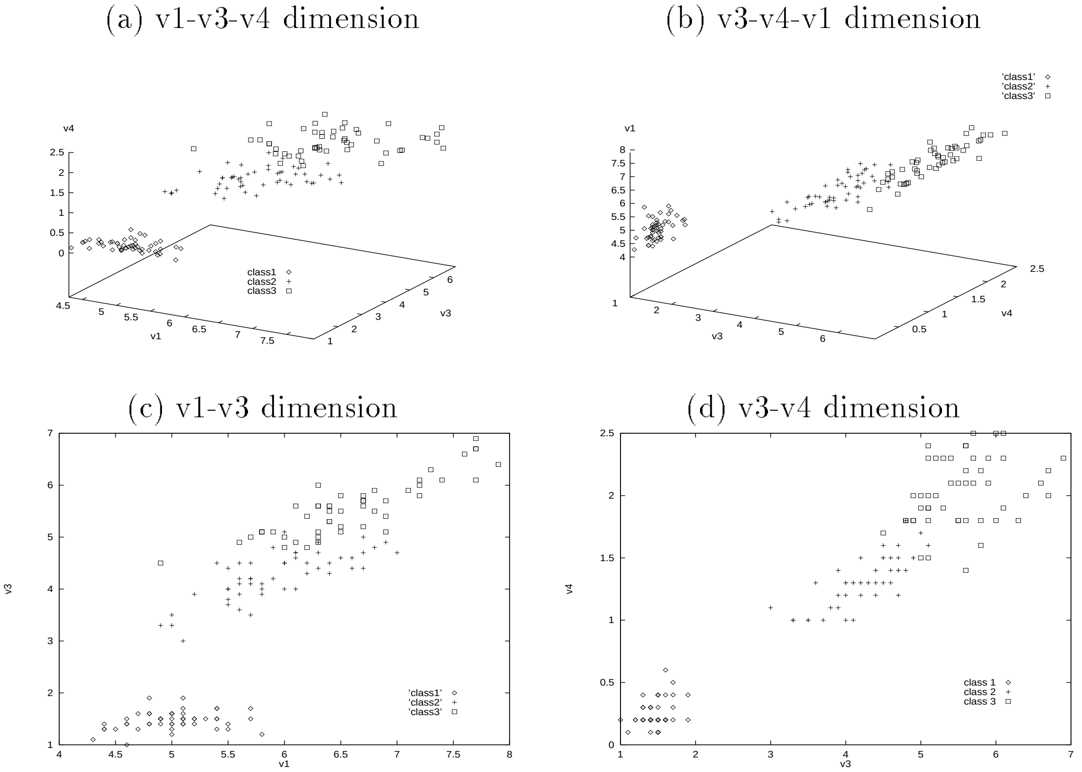

Figure 1: Iris data in different variable dimensional spaces.

2.1 Physical Significance

Consider an instance in a dataset, which is described by a set of variables. If the data has distinct clusters, this instance should belong to some cluster, and the instance must be well separated from some instances and very close to others. Irrelevant variables have no influence, but relevant ones do, on forming distinct clusters. Our method is based on the observation that removing an irrelevant variable from the variable set may not change the underlying concept of the data, but not so otherwise. Consider a set of N instances in an M dimensional hyper-space, where each instance x_i is represented by a vector of M values (i.e., $x_{i1}, x_{i2}, ..., x_{iM}$). Removing a variable from the original set of variables (i.e., decreasing dimensionality of the hyper-space by one) is to project the data onto the $M-1$ dimensional hyper-space. If the distinctness among the clusters remain after projecting instances from M to $M-1$ dimensions, then the variable is not important for describing the underlying structure of the data.

An Example: The Iris data (summarized in Table 1) has 4 variables in total. Figure 1(a) and (b) show the instances in a 3-dimensional space of subset (v_1, v_3, v_4). Removing variable v_4 is equivalent to projecting the data onto a 2-dimensional plane of subset (v_1, v_3) (Figure 1(c)), and removing variable v_1 is equivalent to projecting the data onto a 2-dimensional plane of subset (v_3, v_4) (Figure 1(d)). By examining the figures, we notice that data in Figure 1(d) for subset (v_3, v_4) displays more distinct clusters than data in Figure 1(c) for subset (v_1, v_3). This implies that variable v_4 is more important than variable v_1. Also, as the clusters in Figure 1(d) for subset (v_3, v_4) are roughly as distinct as the clusters in Figure 1(b) for subset (v_3, v_4, v_1), variable v_1 may be removed if only two variables are allowed. This work is to provide a measure that can capture the difference of two subsets by comparing their importance values.

2.2 The Measure

Unsupervised data does not have class information, so we do not have *a priori* information about clusters of classes. There is a need to have an entropy measure that evaluates the distinctness among the underlying clusters given a subset of variables. We say the data has orderly configurations if it has distinct clusters, and has disorderly or chaotic configurations otherwise. From entropy theory [3], we know that entropy (or probability) is less for orderly configurations, and more for disorderly configurations for the simple reason that there are few orderly configurations compared to disorderly configurations. Entropy should be very low (close to 0.0) for very close or very distant pairs of instances, and very high (close to 1.0) for those instances separated by a distance close to the mean distance. We use a similarity measure that is (S) based on distance, and assumes a very small value (close to 0.0) for very close pairs of instances that probably fall to a single cluster, and a very large value (close to 1.0) for very distant pairs of instances that probably fall to different clusters. The idea is to remove as many variables as possible but still maintain the level of distinctness as if no variables were removed. For two instances, the entropy measure is $E = -S\log_2 S - (1-S)\log_2(1-S)$ which assumes the maximum value of 1.0 for $S = 0.5$, and the minimum value of 0.0 for $S = 0.0$ and $S = 1.0$ [6].

For a data set of N instances the entropy measure is given as:

$$\mathbf{E} = -\sum_{i=1}^{N}\sum_{j=1}^{N}(S_{ij} \times \log S_{ij} + (1-S_{ij}) \times \log(1-S_{ij})) \tag{1}$$

where S_{ij} is the similarity value, normalized to [0,1], between the instances x_i and x_j.

When all variables are *numeric* or *ordinal*, the similarity value of two instances is:

$$S_{ij} = e^{-\alpha \times D_{ij}} \tag{2}$$

where D_{ij} is the distance between the instances x_i and x_j, and α is a parameter. If we plot similarity vs. distance, then the curve will have a bigger curvature for a larger α. The experiments with various values for α suggest that it should be robust for all kinds of data sets, and not just for certain data sets. In this work, α is calculated automatically by assigning the maximum value of similarity (i.e., 0.5) in formula (2). This, in fact, produces good results for the tested data sets. Mathematically, it is given as: $\alpha = \frac{-\ln 0.5}{\overline{D}}$ where $\overline{D}$ is the average distance among the instances in a hyper-space. Hence, α is determined by the data and can be calculated easily. Euclidean distance measure is used to calculate the distance D_{ij} between two instances x_i and x_j. In a multi-dimensional space, it is defined as: $D_{ij} = [\sum_{k=1}^{M} (\frac{x_{ik}-x_{jk}}{max_k - min_k})^2]^{1/2}$ where max_k and min_k are the maximum and minimum values for the k^{th} dimension. The interval in the k^{th} dimension is normalized by dividing it by the maximum interval $max_k - min_k$ before calculating the distance.

Similarity for *nominal* variables is measured using the Hamming distance concept. The similarity value of two instances is given as:

$$S_{ij} = \frac{\sum_{k=1}^{M} |x_{ik} = x_{jk}|}{M} \tag{3}$$

where $|x_{ik} = x_{jk}|$ is 1 if x_{ik} equals x_{jk} and 0 otherwise, and M is the number of variables in the subset under consideration.

For mixed data (i.e., both numeric and nominal variables), we can discretize numeric values first before applying our measure [8].

3 Algorithm to Find Important Variables

We use a **S**equential Backward Selection algorithm [2] to determine the relative importance of variables for **U**nsupervised **D**ata (SUD). In the algorithm M is the number of variables originally present, and D is the data set.

SUD(D)
    T = Original Variable Set
    For $k = 1$ to $M-1$ { /* Iteratively remove variables one at a time */
        For every variable v in T { /* Determine which variable to be removed */
            $T_v = T - v$
            Calculate E_{T_v} on D using eqn. 1}
        Let v_k be the variable that minimizes E_{T_v}
        $T = T - v_k$ /* Remove v_k as the least important variable */
        Output v_k }

In each iteration, entropy **E** is calculated after removing one variable from the set of remaining variables. A variable is removed as the least important variable if the set without the variable gives the least entropy. This continues till the importance of all variables is determined. We obtain an ordered list of variables by reversing the order in which they are removed. That is the first variable is the most important and the last one is the least important.

As we are interested in dimensionality reduction, naturally we want to know how many variables we should keep for a task. If we know that an application only needs d variables, we can simply choose the first d variables. The automated selection of d is,

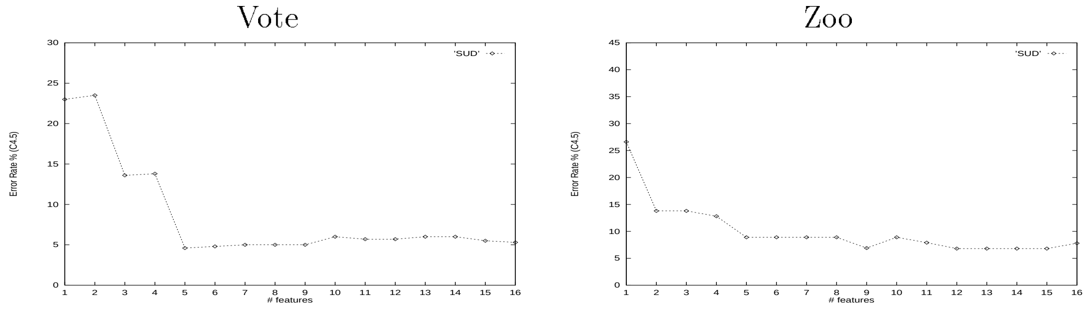

Figure 2: The typical trends of improved performance with increasing number of variables.

however, more complicated. We investigated this issue through experiments on a number of data sets with class labels. We ran SUD by removing class labels, and determined the importance levels of all variables. Then we performed 10-fold cross-validation of C4.5 [13] using the subset consisting of the d most important variables, d varies from 1 to M, to obtain average error rates. Two results are shown in Figure 2 and more in Figure 3. They clearly show that increasing d may not improve the performance of the chosen classifier (C4.5) much. But in most cases the performance ceases to improve after a certain point which varies from one data set to another (in case of the Parity3+3 data, in Figure 3, the performance falls sharply as more variables are added). In order to determine d for any data having class information, one may perform such experiments, and choose d to be the number of variables beyond which the error rate does not fall. If this is not possible, one may use a windowing technique to find d. That is, stop including more variables if the performance does not improve for a few more extra variables.

4 Experimental Studies

We want to show: (1) SUD can find important variables for continuous and nominal data; and (2) its performance is comparable to some method that requires class information unlike SUD.
Method: for part (1) we test SUD over a number of data sets having continuous or nominal variables whose important variables are known. For part (2) we select Relief-F [7], and (a) run SUD (without using class variable) and Relief-F (using class variable) over a total of 15 data sets having class variables (class variables are required to run a classifier) to find variables ordered by their importance, and (b) compare SUD and Relief-F by performing 10-fold cross-validation of C4.5 and OC1 [11] using the subset consisting of the d most important variables, d varies from 1 to M.

Data Sets: a summary of 17 data sets is given in Table 1. Parity3+3 data is a modified version of the original Parity3 data with 6 extra variables in which 3 are redundant, and the other 3 are irrelevant. In the last column of the Table 1 the important variables are shown for the data sets: Iris, Chemical Plant, Non-linear, CorrAL, Monk3, and Parity3+3. For CorrAL, Monk3, and Parity3+3 data sets the target concepts are also

Data Set	Data Type	M	N	# Classes	I.V.
Parity3+3	Nominal	12	128	2	{1,7},{2,8},{3,9}
CorrAL	Nominal	6	64	2	1,2,3,4
Monk3	Nominal	6	122	2	2,4,5
Vote	Nominal	16	135	2	NK
Mushroom	Nominal	22	7125	2	NK
Breast Cancer	Nominal	9	699	2	NK
Lymphography	Nominal	18	148	4	NK
Zoo	Nominal	16	74	7	NK
Iris	Continuous	4	150	3	3,4
Ionosphere	Continuous	34	351	2	NK
Bupa	Continuous	6	345	2	NK
Glass	Continuous	10	214	7	NK
Pima-Diabetes	Continuous	8	768	2	NK
New-Thyroid	Continuous	5	215	3	NK
Wine	Continuous	13	178	3	NK
Chemical Plant	Continuous	5	70	∞	1,2,3
Non-linear	Continuous	4	50	∞	1,2

Table 1: Summary of data sets. I.V. – Important Variables, ∞ – Class variable is continuous, NK – Not Known, N and M are defined earlier.

Data set	Important variables	Decreasing order of importance
Iris	3,4	**3,4**,1,2
Plant	1,2,3	**3,1,2**,4,5
Non-linear	1,2	**2,1**,3,4
Monk3	2,4,5	**5,2,4**,1,6,3
CorrAL	1,2,3,4	**3**,6,**1,2,4**,5
Parity3+3	{1,7},{2,8},{3,9}	**9,3,8,2,7,1**,4,10,6,5,11,12

Table 2: SUD is able to find important variables for data sets with known important variables.

Data	Decreasing order of importance	
	SUD	Relief-F
Par3+3	9,3,8,2,7,1,4,10,6,5,11,12	3,9,7,1,8,2,6,11,12,10,4,5
CorrAL	3,6,1,2,4,5	1,2,6,3,4,5
Monk3	5,2,4,1,6,3	2,5,4,3,6,1
Vote	14,6,5,13,4,12,16,8,7,3,9,15,2,1,10,11	4,3,5,14,11,12,13,8,15,9,10,2,1,6,16,7
Mush	9,3,20,22,5,11,21,2,19,15,14, 1,13,12,4,10,8,7,18,6,17,16	5,20,11,8,19,4,10,22,9,12,13, 21,7,3,2,15,14,18,6,17,16,1
Breast	1,7,3,2,5,6,4,8,9	6,2,3,7,5,1,4,8,9
Lym	18,12,10,13,1,15,11,6,2,16,8,5,17,3,7,4,9,14	2,13,15,10,17,18,5,7,14,11,9,1,3,16,12,6,8,4
Zoo	13,3,4,1,8,16,6,5,2,14,9,10,12,11,15,7	4,3,8,1,13,2,9,10,14,16,12,5,6,7,11,15
Iris	3,4,1,2	4,3,1,2
Iono-sphere	13,15,11,9,7,17,19,21,5,3,23,25,27,29,31,33,10 4,6,12,14,8,16,20,18,22,28,26,24,30,32,34,2,1	34,22,33,6,4,8,16,14,21,9,27,15,30,20,29,24,32, 7,12,18,10,11,3,5,28,25,26,19,23,1,31,13,17,2
Bupa	5,3,4,6,1,2	6,5,3,2,4,1
Glass	9,7,4,8,3,2,5,6,10,1	7,4,9,3,8,5,2,6,10,1
Pima	1,3,6,4,7,2	8,1,2,5,6,4,7,3
Thyroid	4,5,3,2,1	4,3,1,2,5
Wine	7,6,12,9,11,10,5,13,1,4,3,8,2	6,9,1,11,5,7,10,4,12,2,13,3,8

Table 3: Order of variables by SUD and Relief-F.

known. The important variables for Iris and Chemical Plant are based on the results of other researchers. For Iris data, Chiu [1] and Liu and Setiono [8] conclude that v_3 (petal-length) and v_4 (petal-width) are the most important variables. Both Chemical Plant and Non-linear data sets are taken from Yasukawa and Sugeno's paper [9]. For Chemical Plant data, they conclude that the first 3 variables are important. The Non-linear data has an output or class variable y and two input variables v_1 and v_2, and is defined as: $y = (1 + v_1^2 + v_2^{-1.5})^2, 1.0 \leq v_1, v_2 \leq 5.0$. The other two variables in this data set are irrelevant. Most data sets (unless specified) can be found in the machine learning repository in the University of California at Irvine.

Results: we produce the experimental results in Tables 2 and 3, and Figures 3 and 4. Table 2 shows that SUD is able to find the important variables for data sets. This is verified by comparing the list with known one from literature. The class variable is not included for any data set. SUD also assigns high importance to the redundant variables in Parity3+3 and CorrAL. Table 3 shows the order of importance by SUD and Relief-F for 15 data sets with discrete class variables (Plant and Non-linear data sets have continuous class variables, hence, left out). Figure 3 and 4 compares SUD and Relief-F by performing 10-fold cross-validation of C4.5 (inducing axis-parallel decision trees) and OC1 (producing oblique decision trees) using the subset consisting of the d most important variables, $d = 1...M$. We run OC1 only on continuous data as it is suggested that it run on such data. In each of these graphs average error rates are plotted in the y-axis and the number of variables (starting with 1 for the most important variable) in the x-axis. It shows that the performance of SUD (without using class variable) is close

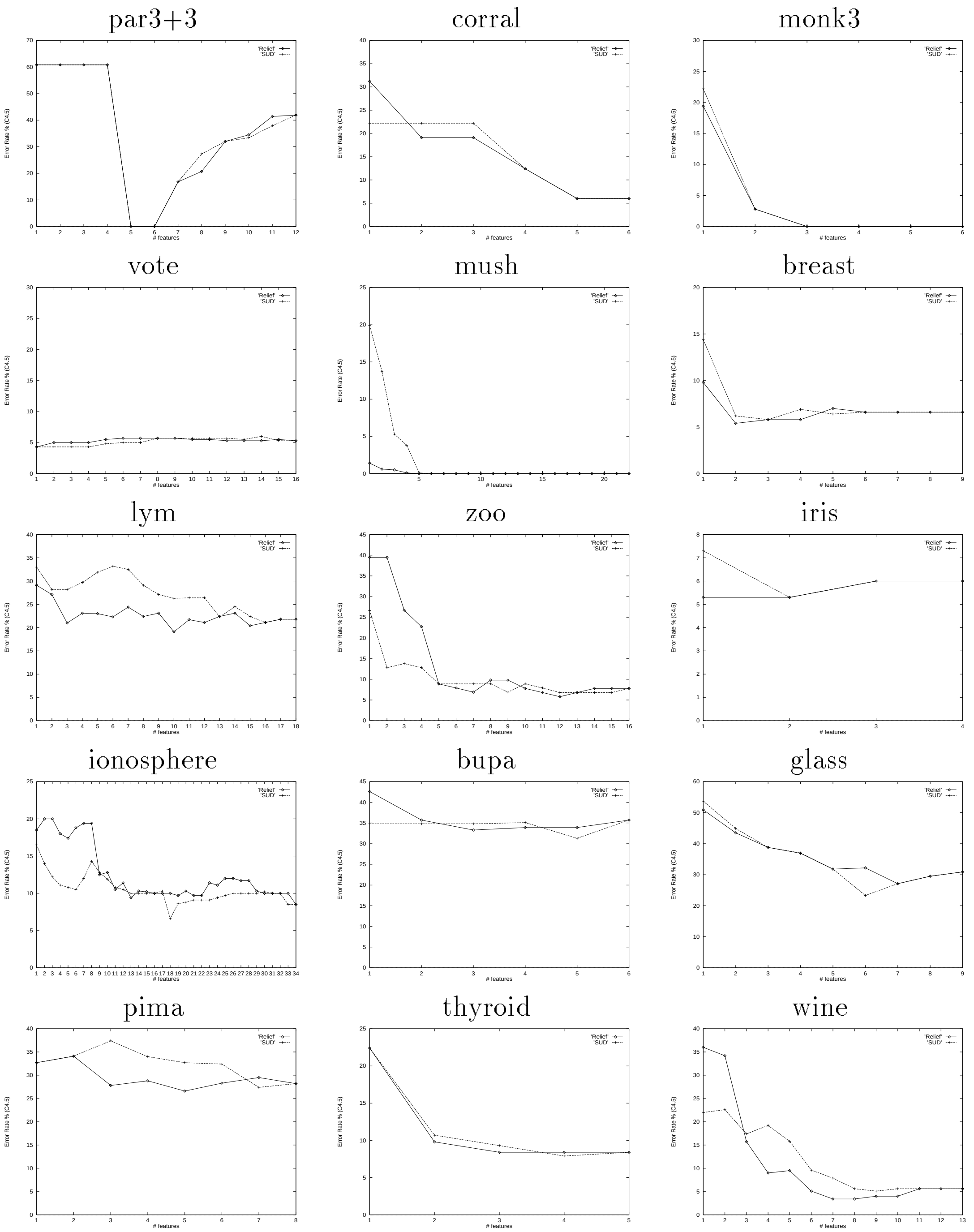

Figure 3: The error rates of 10-fold cross validation of C4.5 using the variables selected by SUD (without class variable) is very close to and sometimes even lower than that using the variables selected by Relief-F (using class variable).

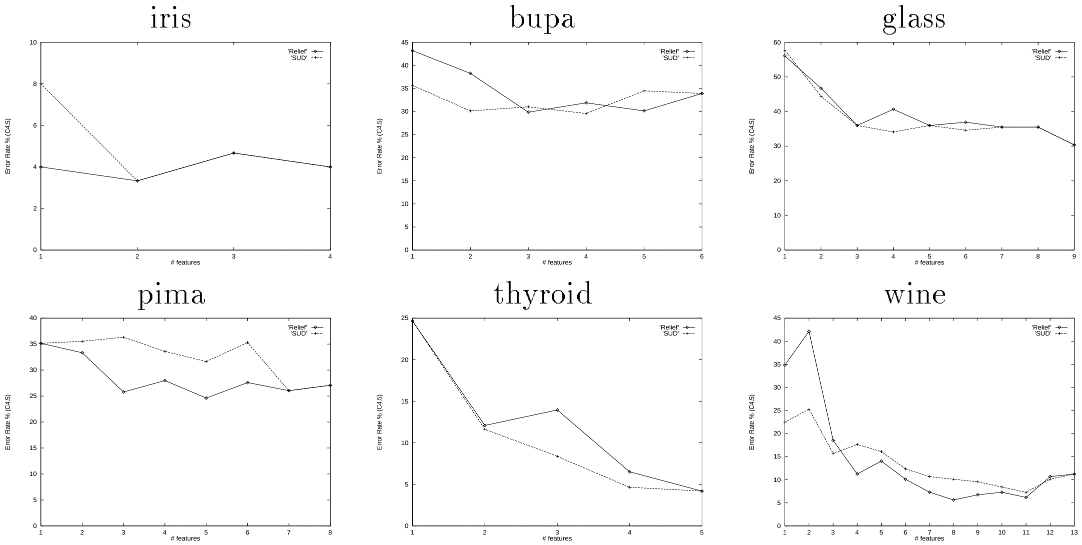

Figure 4: The error rate of 10-fold cross validation of OC1 using the variables selected by SUD (without class variable) is very close to and sometimes even lower than that using the variables selected by Relief-F (using class variable).

to and sometimes even better than that of Relief-F (using class variable).

5 Conclusion

In this paper we focus on solving the problem of reducing the number of original features for unsupervised data - a problem concerns data mining for large size data. Neither feature selection methods nor PCA solves this problem directly. An entropy measure is proposed to measure the importance of a variable. The measure is applicable to both numeric and nominal data types. A sequential backward selection algorithm, SUD, is implemented to determine the relative importance among the features. The issue of choosing d important features is also discussed. We carried out two sets of experiments to show that (a) SUD is able to find important features, and (b) it compares well with a feature ranking algorithm that requires class variables.

References

[1] S. L. Chiu. Method and software for extracting fuzzy classification rules by subtractive clustering. In *Proceedings of North American Fuzzy Information Processing Society Conf. (NAFIPS'96)*, June 1996.

[2] P. A. Devijver and J. Kittler. *Pattern Recognition : A Statistical Approach.* Prentice Hall, 1982.

[3] Johan Diedrich Fast. *Entropy : the significance of the concept of entropy and its applications in science and technology*, chapter 2: The Statistical Significance of the Entropy Concept. Eindhoven : Philips Technical Library, 1962.

[4] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, 1986.

[5] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of Ninth National Conference on AI*, 1992.

[6] George J. Klir and Tina A. Folger. *Fuzzy Sets, Uncertainty, and Information*, chapter Ch. 5: Uncertainty and Information. Prentice-Hall International Editions, 1988.

[7] I. Kononenko. Estimating attributes : Analysis and extension of RELI EF. In *Proceedings of European Conference on Machine Learn ing*, pages 171–182, 1994.

[8] H. Liu and R. Setiono. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of the 7th IEEE International Conference on Tools with Artificial Intelligence(TAT'95)*, pages 388–391, November 1995.

[9] T. Yasukawa M. Sugeno. A fuzzy-logic-based approach to qualitative modeling. In *IEEE Transactions on Fuzzy system Vol.1, No.1*, February 1993.

[10] A. N. Mucciardi and E. E. Gose. A comparison of seven techniques for choosing subsets of pattern recognition. *IEEE Transactions on Computers*, C-20:1023–1031, September 1971.

[11] Sreerama K. Murthy, Simon Kasif, and Steven Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[12] J. R. Quinlan. Induction of decision trees. In *Machine Learning*, pages 81–106. Morgan Kaufmann, 1986.

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

[14] R. Uthurusamy. From data mining to knoweldge discovery: Current challenges and future directions. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 561–569. AAAI Press / The MIT Press, 1996.

[15] N. Wyse, R. Dubes, and A.K. Jain. A critical evaluation of intrinsic dimensionality algorithms. In E.S. Gelsema and Kanal L.N., editors, *Pattern Recognition in Practice*, pages 415–425. Morgan Kaufmann Publishers, Inc., 1980.