Enhancing Density-Based Data Reduction Using Entropy

D. Huang
dihuang@ee.cityu.edu.hk
Tommy W. S. Chow
eetchow@cityu.edu.hk
Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong

Data reduction algorithms determine a small data subset from a given large data set. In this article, new types of data reduction criteria, based on the concept of entropy, are first presented. These criteria can evaluate the data reduction performance in a sophisticated and comprehensive way. As a result, new data reduction procedures are developed. Using the newly introduced criteria, the proposed data reduction scheme is shown to be efficient and effective. In addition, an outlier-filtering strategy, which is computationally insignificant, is developed. In some instances, this strategy can substantially improve the performance of supervised data analysis. The proposed procedures are compared with related techniques in two types of application: density estimation and classification. Extensive comparative results are included to corroborate the contributions of the proposed algorithms.

1 Introduction

As computer technology grows at an unprecedented pace, the size of data sets increases to the extent that data analysis has become cumbersome. Theoretically, using more data samples generally leads to improved data analysis (Provost & Kolluri, 1999; Friedman, 1997). But directly mining a data set in the gigabytes is a formidable, or even impossible, task because of the computational burden. In order to handle this problem, data reduction techniques have been studied (Blum & Langley, 1997; Catlett, 1991; Hart, 1968; Mitra, Murthy, & Pal 2002; Wilson & Martinez, 2000). Data reduction algorithms have been designed to reduce a huge data set to a small representative, but informative, pattern set on which data analysis can be performed. The belief is that data reduction introduces no or only minimal negative effect on data analysis.

The simplest methods of data reduction are to sample data patterns in a random or a stratified way. These can be easily implemented and have negligible computational burden, so they are widely used as evaluation baseline. They are, however, unable to guarantee stable performance because the randomness adopted in these methods may cause a loss of useful information (Catlett, 1991). A number of more sophisticated data reduction techniques have also been developed. These techniques can be categorized into two main groups: a classification-based method and a distribution-based method (Mitra et al., 2002; Blum & Langley, 1997). The former identifies the patterns that are informative for constructing a classification model, and the latter determines the patterns so that the original data distribution can be preserved as much as possible.

Classification-based methods assume that all patterns are not equally important for certain classification learning algorithms. Through rejecting the "useless" patterns, this type of method can improve the scalability and the final results of a classification process (Wilson & Martinez, 2000). The condensed nearest neighbor rule (CNN) (Hart, 1968) was one of the first classification-based methods that was developed. Subsequently, many k-nearest-neighbor (kNN) rule-based data reduction schemes, such as PNN (Gates, 1972) and RNN (Chang, 1974) were proposed (Bezdek & Kuncheva, 2001; Dasarathy, 1991). There are other types of classifiers, for instance, neural networks and decision trees, that have been explored for data reduction (Plutowski & White, 1993; Quinlan, 1983; Schapire, 1990). Generally, the classification-based data reduction method works with a specified classification model. The reduced data set is iteratively adjusted according to classification accuracy. These steps repeat until the classification accuracy cannot be further improved. Because these data reduction methods may have distorted the original data distribution, they do not support other classification models and other pattern recognition tasks. Also, the classification-based methods are likely to fail in a way that they confuse outliers with the real useful samples because outliers always have relatively high uncertainty (Roy & McCallum, 2001). (For discussion on classification-based data reduction, see Provost & Kolluri, 1999; Blum & Langley, 1997; Wilson & Martinez, 2000.)

The distribution-based data reduction method (Mitra et al., 2002; Gray, 1984; Kohonen, 2001; Astrahan, 1970) determines the representative pattern sets, that is, the reduced data set, so that the data distribution can be preserved as much as possible. This is a versatile method that can support various pattern recognition models tackling various pattern recognition tasks. Vector quantization error (VQE) is a popular technique employed by different distribution-based data reduction schemes (Gersho & Gray, 1992; Kohonen, 2001; Chow & Wu, 2004). VQE measures the distance between each pattern and its corresponding representative. A small VQE implies a good data reduction result. Through optimizing VQE, a representative data set, which is a reduced data set and is referred as *codebook* in the context of clustering, can be obtained. A self-organizing map (SOM) (Kohonen, 2001; Chow & Wu, 2004) is a typical descent-based algorithm designed for minimizing VQE. In SOM, the learning parameters are problem dependent, and the determination of these parameters has never been a straightforward issue. Another type of distribution-based method uses density estimation

(Mitra et al., 2002; Astrahan, 1970). In this approach, the main difficulty lies in the strategy for estimating the underlying probability density. The maximum likelihood learning algorithm is employed for density estimation by Yang and Zwolinski (2001), although it may not be efficient with a data set with relatively complex data distribution. Alternatively, a much more efficient strategy has been employed to analyze probability density (Astrahan, 1970; Mitra et al., 2002). The rationale behind this strategy is simple. For a pattern x, the density of it must be inversely related to the distance between x and its kth nearest neighbor. Thus, Mitra et al. (2002) and Astrahan (1970) analyze the density of each pattern after calculating the distances between the patterns. Without involving a learning process, this strategy is rather efficient, but it still requires calculating the distances between all possible pairs of patterns. This is computationally and memory demanding when one is handling a large amount of patterns. The problem of outliers is also significant because they may degrade the performance of supervised pattern recognition (Han & Kamber, 2001). But the filtering of outliers is overlooked in most studies.

In this article, we focus on density-based data reduction. New types of data reduction criteria are introduced, based on the concept of entropy. They are thus called representative entropy (RE)/weighted representative entropy (WRE). In our proposed RE and WRE, all the relationships between a data point and all its representatives are fully considered. This makes RE and WRE more sophisticated than VQE, which considers only a single relationship between a point and one of its representatives. This issue is elaborated in a later section. We describe the design of a new data reduction algorithm using RE or WRE. The proposed algorithm, which begins with a randomly selected data set R_0, includes two sequential stages: a forward searching stage and an RE/WRE–based step-wise searching stage. Compared with other density-based methods (Astrahan, 1970; Mitra et al., 2002), the proposed method exhibits two main advantages. First, it is computationally efficient because the exhaustive task of calculating the distances between all possible pattern pairs is avoided. More important, because of the characteristics of RE/WRE, the improved efficiency has not traded off the quality of data reduction. Second, we propose an outlier-filtering strategy that can be implemented in a simple but efficient way. This scheme is particularly useful in handling classification problems.

In the next section, the concept of entropy is briefly presented. The proposed data reduction criteria are introduced and evaluated in section 3, the proposed data reduction method is detailed in section 4, and experimental results are presented and discussed in section 5.

2 Preliminaries on Entropy

The concept of entropy, which originated with thermodynamics, has been extended to statistical mechanics and information theory (Cover & Thomas, 1991). For a discrete distribution modeled by $X = \{x_1, x_2, \ldots, x_N\}$, entropy measures the "information" conveyed by X. "Information" means the uncertainty or the degree of surprise for a particular value of X being drawn.

Suppose that x is a value drawn from X, the event $x = x_k$ occurs with probability p_k, and the sum of the probabilities for $x = x_k$ $(k = 1, 2, \ldots, N)$ is 1, that is, $\sum_{k=1}^{N} p_k = 1$. In the case of $p_k = 1$, there is no uncertainty or surprise for $x = x_k$. A lower value of p_k increases the uncertainty or the "information" when it is known that $x = x_k$ occur. Thus, this "information" is generally measured by $I(x_k) = -\log(p_k)$. The "information" contained by the whole event set X is called entropy and is enumerated by the expected value of $-\log(p_k)$, that is,

$$H(X) = E(I(x_k)) = -\sum_{i=1}^{N} p_i \log p_i. \tag{2.1}$$

A large value of entropy $H(X)$ indicates high uncertainty about X. When all the probabilities (i.e., p_k for all k,) are equal to each other, we have the maximal uncertainty that the value in X is taken, and the entropy $H(X)$ achieves its maximum $\log(1/N)$. Conversely, when all the p_i except one are 0, there is no uncertainty about X, that is, $H(X) = 0$.

3 Data Reduction Criteria

3.1 Prior Work–Vector Quantization Error. Vector quantization, first developed in the community of signal process, is a technique that exploits the underlying data distribution to compress data (Gersho & Gray, 1992). This technique partitions a given data domain into a number of distinct regions and then generates a representative point in each region. Suppose that $R = \{r_1, r_2, \ldots, r_K\}$ is a representative set, that is, a codebook, of X. The VQE is defined as

$$VQE(R; X) = \frac{1}{N} \sum_{all\ x \in X} VQE(R; x) = \frac{1}{N} \sum_{all\ x \in X} \arg\min_j d(r_j, x)^2, \tag{3.1}$$

where $d(r_j, x)$ is the distance (dissimilarity) between a pattern x and a representative r_j. Intuitively, VQE measures the similarity between R and X. Until now, VQE has been the most popular objective function for data clustering or data reduction. Equation 3.1 shows that for a pattern, VQE considers the relationship of that pattern with one representative—the one closest to it. VQE explicitly ignores the relationships with other representatives.

3.2 Entropy-Based Data Reduction Criteria: Representative Entropy and Weighted Representative Entropy

3.2.1 Definitions of RE and WRE. Assuming that R is the result of a data reduction process, then the probability of x ($x \in X$) being represented by a representative r_j (i.e.,$r_j \in R$) is $p(r_j|x)$. Also the probability of x being represented by all representatives is 1 (i.e., $\sum_{j=1}^{K} p(r_j|x) = 1$). It is ideal that each pattern in X is close to one and only one representative as much as possible. In terms of probability, it is expected that $p(r_i|x)$ is zero for all i except one. The more uneven the distribution of the representation information of$R$ to X is, the better is the representative set R. This motivates us to explore the concept of entropy for evaluating the quality of R. The proposed criterion, called the representative entropy for data reduction (RE), is defined by

$$RE(R;\, x) = \frac{\sum_{j=1}^{K} -p(r_j|x)\log(p(r_j|x))}{\log(K)}.$$

For a pattern x, a smaller value of RE(R;x) indicates that x is more likely to be represented by only one object in R. That is, a small RE(R;x) indicates a good representation performance at$x$. For X, the representation entropy of R is

$$RE(R;\, X) = \frac{1}{N}\sum_{all\ x \in X} RE(R;\, x) = \frac{1}{N}\sum_{all\ x \in X} \frac{\sum_{j=1}^{K} -p(r_j|x)\log\left(p(r_j|x)\right)}{\log(K)}. \tag{3.2}$$

The denominator log(K) of RE is used for normalization, with which RE is limited to the range of [0, 1]. In this case, it is reasonable to compare the representative sets with different sizes.

RE can be rewritten as

$$\begin{aligned} RE(R;\, X) &= \frac{1}{\log(K)} \sum_{j=1}^{K} \underbrace{\frac{1}{N}\sum\nolimits_{all\ x \in X} -p(r_j|x)\log(p(r_j|x))}_{RE(r_j, X)} \\ &= \frac{1}{\log(K)} \sum_{j=1}^{K} RE(r_j,\, X). \end{aligned} \tag{3.3}$$

RE(r_j, X) in equation 3.3 evaluates the representation ability of a single representative r_j. Suppose that r_j covers or represents part of the original data space A_j. According to equation 3.3, RE(r_j, X) will achieve the minimum only when $p(r_j|x) = 0$ or 1 for all $x \in X$; that is, A_j has no overlap with the areas covered by other representatives. Under this condition, r_j can be considered a respectable representative. As the overlap of A_j with other areas increases, the representational ability of r_j decreases, whereas the value of RE(r_j, X) increases. Clearly, the value of RE(r_j, X) reflects the representational ability of r_j, for instance, a low value of RE(r_j, X) corresponds to a good r_j. This characteristic plays an important role in the proposed data reduction scheme, which is detailed in section 4.2.

To evaluate r_j, RE considers A_j, the data area around r_j. It is reasonable to assume that the patterns lying in A_j may have different contributions to this evaluation task. See x_1 and x_2 illustrated in Figure 1. x_1 must be more important than x_2 with respect to r_1, whereas x_2 is more influential for r_3 than x_1. RE(R, X) (see equation 3.3), however, does not incorporate this idea. Thus, RE(R; X) is modified with a weighting operation. The weighted RE (WRE) is defined as

$$
\begin{aligned}
WRE(R;\,X) &= \frac{1}{\log(K)} \sum_{j=1}^{K} WRE(r_j,\,X) \\
&= \frac{1}{\log(K)} \sum_{j=1}^{K} \frac{1}{N} \sum_{all\;x\in X} \underbrace{(-p(r_j|x)\log(p(r_j|x)))}_{A} \cdot \underbrace{p(x|r_j)}_{B}.
\end{aligned}
\tag{3.4}
$$

Part A of WRE, inherited from RE, measures the distribution of representational information, while part B, measuring the relationship of a data pattern to a representative, is a weighting factor. With part B, when a representative is evaluated, patterns close to that representative have a greater effect than those far from it. In section 5, the effect of this weighting operation is evaluated.

3.2.2 Calculation of RE and WRE. To calculate RE or WRE, $p(r_j|x)(j = 1, \ldots, K)$ must be known. Popular schemes used for estimating the probabilities include the maximum likelihood algorithm (Yang & Zwolinski, 2001) and the Bayesian-based algorithm (Duda, Hart, & Stork, 2001). However, these methods are not computationally simple and efficient enough for an iterative data reduction process, in which probability estimation must also be estimated many times. We thus adopt a distance-based probability estimation method.

A representative covers the L original patterns nearest to it. That is, for a representative (say, r_j), if x is one of the L patterns nearest to r_j, $p(x|r_j) > 0$; otherwise, $p(x|r_j) = 0$. Based on this idea, we estimate $p(x|r_j)$ by using

$$
p(x|r_j) = \begin{cases} 1 - \frac{d(x,r_j)}{Radius(r_j)}, & d(x, r_j) \le Radius(r_j) \\ 0, & \text{otherwise} \end{cases},
$$

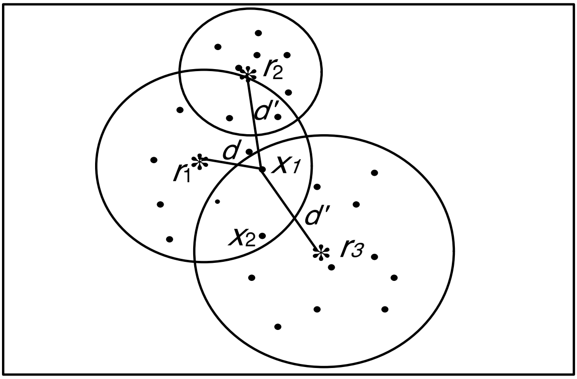


Figure 1: The relation between the representatives and the original data patterns. The original data and the representatives (i.e., the reduced data) are marked by "." and "*," respectively.

where $Radius(r_j)$ is the distance of r_j with the $(L+1)$th pattern nearest to it. Also, $p(r_j)$ can be calculated with $p(r_j) = 1/K$ (K is the size of R). According to the Bayes formula (Duda et al., 2001), $p(r_j|x)$ is estimated by

$$p(r_j|x) = \frac{p(x|r_j)p(r_j)}{p(x)} = \frac{p(x|r_j)p(r_j)}{\sum_{j=1}^{K} p(x|r_j)p(r_j)} = \frac{p(x|r_j)}{\sum_{j=1}^{K} p(x|r_j)}.$$

3.3 Comparison Between the Vector Quantization Error and RE/WRE. Referring to the definitions, the main advantage of RE (see equation 3.3) and WRE (see equation 3.4) over VQE (see equation 3.1) is evident. Let us detail the goal of data reduction first. As illustrated in Figure 1, suppose that d is the distance between a pattern and the representative closest to that pattern, and d' denotes the distance of that pattern to any representative but the closest one. The goal of data reduction is to decrease all d and to increase all d' at the same time. Explicitly considering only d, VQE can be minimized through reducing d. On the other hand, both d and d' are included in RE or WRE.

Below, VQE and RE/WRE are briefly evaluated in synthetic scenarios. A data set with 100 data points (say, A) was generated from the normal distribution

$$N\left([0, \quad 0], \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right).$$

Then a representative set having 10 points (say, B) was drawn from A under the constraint that a point in B must be in the disc with the center [0, 0] and

Table 1: Evaluation of Data Reduction Criteria Under Synthetic Circumstances.

a. Results on the Data with a Normal Distribution.

g^2	Vector Quantization Error	RE	WRE
0.5	2.20 ± 1.98	0.068 ± 0.004	0.075 ± 0.004
1.5	2.68 ± 1.91	0.051 ± 0.006	0.058 ± 0.007
2.5	2.98 ± 1.95	0.046 ± 0.007	0.051 ± 0.007
3.5	3.34 ± 2.41	0.042 ± 0.006	0.047 ± 0.007
4.5	3.27 ± 2.19	0.040 ± 0.007	0.045 ± 0.008

b. Results on the Data with a Uniform Distribution.

g^2	Vector Quantization Error	RE	WRE
0.5	0.91 ± 0.45	0.054 ± 0.006	0.060 ± 0.007
0.8	1.01 ± 0.48	0.044 ± 0.007	0.050 ± 0.007
1	1.13 ± 0.51	0.042 ± 0.007	0.047 ± 0.008

the radius g. Basically, as g increases, this constraint becomes loose, and the likelihood that B represents A increases accordingly. It is expected that the variation of a data reduction criterion is able to roughly reflect this fact. In this study, g^2 changes in a range of [0.5, 4.5]. The statistical results over 500 independent trials are listed in Table 1a. It shows that RE or WRE decreases with as g increases, which is in accord with the expectation. In contrast, VQE cannot perform well because it changes in a way opposite to the theoretical expectations. Also, these data reduction criteria are compared on data generated from a uniform distribution. One hundred points of A were uniformly distributed in the square area from $[-1, -1]$ to $[1, 1]$. The results obtained on this uniform distribution are presented in Table 1b. It is shown that RE or WRE correctly reflects the representation ability of B, whereas VQE does not.

4 Density-based Data Reduction Method

4.1 Multiscale Method. The multiscale method (Mitra et al., 2002) is a typical density-based data reduction scheme. In this method, the density of a pattern is analyzed according to the distance of that pattern to its neighbor. All patterns are then ranked in order of density. With this ranking list, the representatives are recursively determined. Given data X, this method can be briefly stated as follows:

Step 0. Determine the parameter k. This parameter is closely related to the size of the data region covered by a representative.

Step 1. Calculate the distance between all possible pattern pairs in X.

Step 2. Repeat the following operation until there is no remaining pattern in X.

For a pattern in X, determine the distance of it to the kth neighbor. According to these distance values, identify the densest pattern, say, x_{d}, and mark it as representative. Finally draw the disc with the center at x_{d} and the radius of $2rad_{\mathrm{d}}$, where rad_{d} is the distance between x_{d} and its kth neighbor. Delete the patterns lying in this disc.

In this process, the computational and memory requirement of step 2 is $O(N^2)$, where N is the size of X. Apparently, this step will be computationally expensive when a large data set is given.

4.2 RE/WRE–Based Data Reduction Method

4.2.1 Procedure. The proposed method has two sequential stages: a forward searching stage and an RE/WRE–based stepwise searching stage. At the beginning, a set of data points, R_0, is randomly drawn from a given data set, and the representative set R is empty. Then the forward search is conducted on R_0 to recursively place the appropriate representatives into R. This process stops when R_0 has been scanned through, or when the representatives selected in R have been in a desired number. Following the forward process is an RE/WRE–based stepwise process, in which a pattern is first identified as representative from the area that is not yet covered well by R. When R has been in the desired size, the "worst" representative (i.e., the one having the lowest representation ability) will be deleted after a new representative is determined. The "worst" representative is identified according to $\mathrm{RE}(r_j, X)$ or $\mathrm{WRE}(r_j, X)$. In this sense, the proposed data reduction method is called REDR or WREDR. For a given data set X containing N patterns, REDR or WREDR can be stated as follows:

Step 0. Randomly select a pattern set R_0. Set the representative set R empty. Determine K, the desired size of R. Naturally, a representative will represent L $(L = N/K)$ patterns of X.

Step 1. This step repeats until R_0 is scanned through or R has contained K elements.

In X, determine the top Lpatterns nearest to r_j (any$r_j \in R_0$). Based on the sum of the distances of r_j with these patterns, the densest element of R_0 (say, r_d) is identified and placed into R. The top Lpatterns nearest to r_d (including r_d itself) will be rejected in the further forward searching stage.

Step 2. In X, sort out the patterns having $\max_{r_j \in R} p_{outer}(x|r_j) > 0$. Among these patterns, identify the one with $\min(\max_{r_j \in R} p_{outer}(x|r_j))$, and put it into R. $p_{outer}(x|r_j)$ is defined by

$$p_{outer}(x|r_j) = \begin{cases} 1 - \frac{d(x,r_j)}{Rad\_outer(r_j)}, & d(x,r_j) \leq Rad\_outer(r_j) \\ 0, & \text{otherwise} \end{cases},$$

where $Rad\_outer(r_j)$ is the distance of r_j with the $2L$th pattern nearest to it.

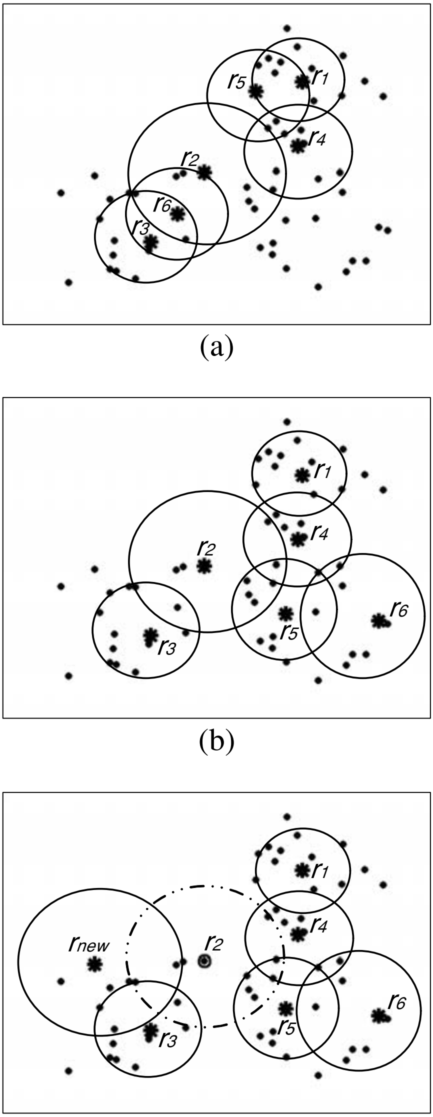
Step 3. When R consists of $K+1$ representatives, delete the worst representative—the one having the largest RE(r_j, X) or WRE(r_j, X).

Step 4. Calculate RE(R, X) or WRE(R, X) for the newly constructed R.

Step 5. Repeat from step 2 to step 4 until RE(R, X) or WRE(R, X) cannot be reduced for five consecutive iterations.

Below, an example is given to demonstrate the procedure of WREDR. In this example, WREDR identifies 6 representatives from 60 original patterns. In Figure 2, certain intermediate results are illustrated. The original patterns and representatives are marked by "." and "*", respectively. The disc around a representative roughly illustrates the region covered by that representative. Figure 2a shows that 6 patterns are selected into R_0 during initialization. The representative ability of these 6 patterns is poor because the area at the bottom right is uncovered, and the regions covered by r_1 andr_5 overlap each other. The forward searching process tackles the problem of overlapping. As a result, r_1 is marked as representative, and r_5 is eliminated. Also, r_6 is eliminated because it is redundant to r_2 and r_3. Obviously, in this forward searching stage, qualified representatives are selected from R_0, and redundant ones are deleted. However, the area uncovered by R_0—for instance, the bottom right in this example—has not been explored yet. This task will be fulfilled in the following WRE-based stepwise searching process in which new representatives r_5 and r_6, illustrated in Figure 2b, are determined consecutively in the order of r_5 to r_6. Apparently, in this course, the bottom right of the given data space is gradually explored. After r_5 and r_6 are added, the size of R is 6. This is the desirable value. Thus, in the following, the WRE-based process substitutes the "worst" representative with a new one. In this example, r_2 is determined as the "worst" and replaced by *rnew*. As suggested in Figure 2c, the region covered by r_2 has much overlap with the regions covered by other representatives. In this sense, deleting r_2 is reasonable. Figure 2c also shows that substituting r_2 with *rnew* improves the representative ability of R because more data patterns are covered, and the overlap between representatives is further reduced.

4.2.2 Remarks. In the first stage of REDR or WREDR, the density of the patterns in R_0 is analyzed, whereas the density of only one data pattern is required at each iteration in the second stage. Assume that there are k_0



(a)

(b)

(c)

Figure 2: Demonstration of the proposed data reduction procedure. The original data patterns are marked with "." The representatives are described using "*."

patterns selected into R_0 and that the second stage runs n_i iteration. The computational complexity of REDR is then $O(N(n_i + k_0))$, where N is the size of X. Recall that the computational complexity of the multiscale method is $O(N^2)$. Generally, k_0 is much less than N. The proposed method is rapidly convergent, as suggested by our experimental results. Thus, we have

$$N(n_i + k_0) \ll N^2.$$

That is, the proposed methods have a substantially lower computational requirement compared to the multiscale method. Also, the proposed methods have a significantly lower memory requirement compared with the multiscale method because the proposed methods do not need to remember a large number of distance values.

In the above process, the condition of $\max_{r_j \in R} p_{outer}(x|r_j) > 0$ in step 2 guarantees a stable data reduction process. Apparently the patterns with $\max_{r_j \in R} p_{outer}(x|r_j) = 0$ must still be uncovered by R. Without a priori knowledge on the density distribution of these patterns, determining the representative from them is not recommended. With the constraint of $\max_{r_j \in R} p_{outer}(x|r_j) > 0$, the newly determined representative must be around the boundary of the area that has already been covered by R. In this way, the proposed method can gradually and reliably explore the whole data space. This can be seen in the above example in which r_5 and r_6 are marked as representative in turn.

In a supervised task, the proposed methods are conducted in a stratified way. Given a value of L in step 0, which determines the ratio between the sizes of the reduced data and the original data, the pattern set of each class is reduced separately. The collection of these reduced data sets is the final data reduction result. Apart from this stratified way, a new strategy is developed to filter out outliers.

An outlier is an object that behaves in a different way from the others. For instance, in a classification data set, an outlier generally exhibits a different class label from those having similar attributes. Theoretically, outliers cause noise to the supervised learning algorithms and degrade the performance of these algorithms (Han & Kamber, 2001). It is always desirable to eliminate outliers. In our proposed outlier-filtering strategy, a representative candidate is checked before being added to R, which is described in step 1 or 2. Only those that are not outliers can be placed into R.

To determine if an object is an outlier, the area around that object is considered. Assume that A_o is the area around a representative candidate r_o. For any class (say, c_k), the conditional probability $p(A_o|c_k)$ can be calculated using

$$p(A_o|c_k) = \frac{1}{N}\sum_{x \in X} p(x|r_o)p(x|c_k) \ = \frac{1}{N} \sum_{x \in \text{ class } c_k} p(x|r_o),$$

Table 2: Comparisons Between WREDR and WREDR-FO.

Number of Outliers	WREDR	WREDR-FO	No Data Reduction
10	0.98 ± 0.02	1.00 ± 0	0.97
50	0.89 ± 0.05	1.00 ± 0	0.90
80	0.81 ± 0.05	0.99 ± 0	0.83
120	0.79 ± 0.07	0.99 ± 0	0.78

Note: These results highlight the contributions of the filtering-outlier strategy.

where N is the number of patterns in X. The class having the maximum of $p(A_o|c_k)$ is the dominant class of A_o. Apparently, if the class of r_o is consistent with the dominant class of A_o, $r_o$is not an outlier; otherwise, r_o is an outlier and cannot be included in R. In the proposed method, the computational burden of $p(A_o|c_k)$ is very small because $p(x|r_o)$ has been estimated during the calculation of RE/WRE. Below, the proposed algorithm, together with this outlier-filtering strategy, is called REDR-FO or WREDR-FO.

To highlight the benefits of this outlier-filtering strategy, WREDR-FO is compared to WREDR. A classification data set was generated from two normal distributions that consist of two classes {0, 1}:

$$\text{Class 1 (class} = 0\text{) (500 data points)} X \sim N\left([1, 1], \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}\right),$$

$$\text{Class 2 (class} = 1\text{) (500 data points)} X \sim N\left([-1, -1], \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}\right).$$

A thousand patterns of this data set were evenly split into two groups of equal size for training and testing. Certain patterns were randomly chosen in the training data and were mislabeled to generate outliers. Classification accuracy is used to evaluate the quality of a reduced data set. Obviously, high classification accuracy indicates a good reduced data set. In this section, the kNN rule with $k = 1$ is used as the evaluation classifier. WREDR and WREDR-FO are required to reduce the original 500 training patterns to 50. As the performance of these data reduction methods may be affected by different initializations, statistical results of 10 independent trials are presented.

Table 2 lists the comparative results. It is noted that as the number of outliers increases, the contribution of the outlier-filtering strategy becomes more significant. It is worth noting that because of the proposed outlier-filtering strategy, WREDR-FO is able to improve the final classification performance.

5 Experimental Results

Here, *reduction ratio* denotes the ratio between the sizes of reduced data and given data. For a given training data set, a testing data set; and a reduction ratio, a data reduction method is first used to reduce the training data set. Then, based on the reduced data set, certain density estimation and classification models are built. With the performance of these models on the testing data set, the employed data reduction method is evaluated. Throughout our investigation, the following strategies are adopted unless stated otherwise:

1. Each input variable is preprocessed to have zero mean and unit variance through a linear transformation.
2. In WREDR or REDR, the size of R_0 is the same as that of R, the final data reduction result.
3. In the sampling type schemes, SOM and the proposed methods, performance is affected by initialization. Thus, in each case, these methods are independently run 10 times. The statistical results over the 10 trials are presented here.
4. Unlike other methods, Mitra's multiscale method (Mitra et al., 2002) delivers only one set of result in each case. The results delivered by this method are by no mean statistical ones. Also, an exact desired reduction ratio, 0.1 or 0.3, may not be obtained; trials with different values of k may deliver only a close but not the exact reduction ratio. Thus, the reduction ratios provided later in this article are simply the closest ones. For a given reduction ratio, we choose the trial in which the actual reduction ratio is the closest to that given value and present the results of that trial here.
5. Our investigations are conducted using Matlab 6.1 on a Sun Ultra Enterprise 10000 workstation with 100 MHz clock frequency and 4 GB memory.

5.1 Density Estimation. The study presented in this section was conducted on synthetic data, in which the real density functions are known. More important, a large number of testing data patterns can be generated to guarantee the accuracy of evaluation results.

Five data reduction methods—random sampling scheme, SOM, the density-based multiscale method, REDR, and WREDR—are compared from the perspectives of efficiency and effectiveness. The running time is recorded for efficiency evaluation. The effectiveness is measured by using the difference between the real density function ($g(x)$) and a density estimation function obtained based on reduced data ($f(x)$). A small density difference indicates good reduced data.

In this study, $g(x)$ is known, whereas $f(x)$ is modeled with a Parzen window (Parzen, 1962). Given an M-dimensional pattern set $Q = \{q_1, q_2, q_3, \ldots, q_{Nq}\}$, the Parzen window estimator is

$$p(x) = \frac{1}{Nq}\sum_{i=1}^{Nq} p(x|q_i) = \frac{1}{Nq}\sum_{i=1}^{Nq} \kappa(x - q_i, h_i), \tag{5.1}$$

where $\kappa(\bullet)$ is the kernel function and h_i is the parameter to determine the width of the window. With the proper selection of $\kappa(\bullet)$ and h_i, the Parzen window estimator can converge to the real probability density (Parzen, 1962). In this study, κ is a gaussian function. The M-dimensional gaussian function is

$$\kappa(x - q_{\mathrm{i}}, h_i) = G(x - q_i, h_i) = \frac{1}{(2\pi h_i^2)^{M/2}} \exp\left(-\frac{1}{2h_i}(x - q_i)(x - q_i)^T\right).$$

For the pattern q_i, the window width h_i is determined with $h_i = d(q_i, q_j)^2$ where $d(q_i,\ q_j)$ is the Euclidean distance, that is, $d(q_i, q_j) = \sqrt{(q_i - q_j)(q_i - q_j)^T}$, and q_j is the pattern that is the kth nearest to q_i. We use two settings, $k = 2$ and $k = 3$, and the results indicate that the latter one performs better than the former one. Thus, we set k with 3. The difference between $g(x)$, the real density function, and $f(x)$, the density estimation function on a reduced data using equation 5.1, is measured with two indices: the absolute distance D_{ab} and the Kullback-Leibler distance (divergence) D_{KL}. D_{ab} and D_{KL} are, respectively, defined as

$$D_{ab}(f(x), g(x)) = \int_x |f(x) - g(x)| dx \text{ and}$$

$$D_{KL}(f(x), g(x)) = \int_x f(x) \log \frac{f(x)}{g(x)} dx.$$

The integrals of the above equations are calculated by using a numerical approach. After a large set of patterns, *TX*, is evenly sampled in a given data space, D_{ab} and D_{KL} are approximated on *TX* by using

$$D_{ab}(f(x), g(x)) \approx \sum_{tx_i \in TX} |f(tx_i) - g(tx_i)| \Delta tx_i,$$

$$D_{KL}(f(x), g(x)) \approx \sum_{tx_i \in TX} f(tx_i) \log \frac{f(tx_i)}{g(tx_i)} \Delta tx_i.$$

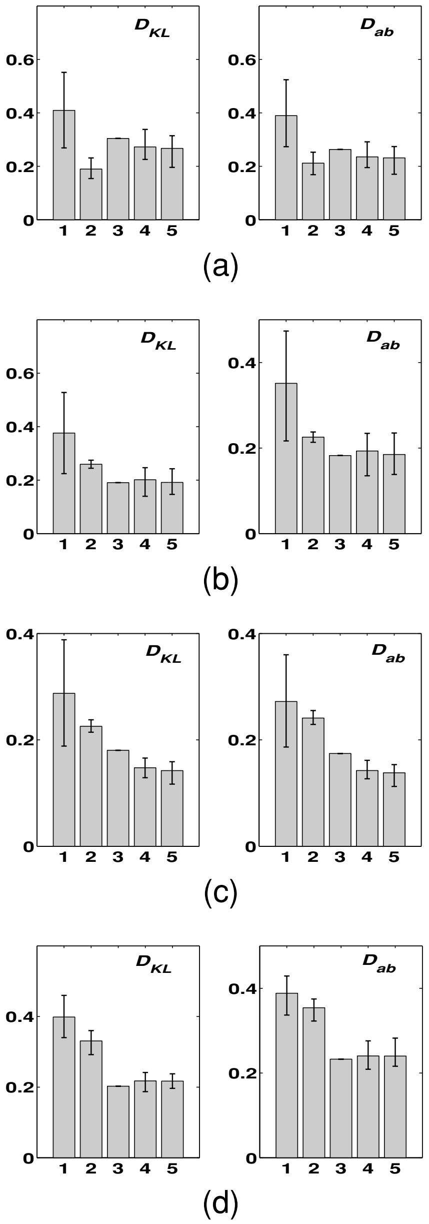
Table 3: Data Sets Used in Density Estimation Application.

Name of Data	Distribution of Data	TX
Data1	600 from$N([0, 0],\ I_2)$.	1681 patterns in [3, 3]~[−3, −3]
Data2	800 from$N([0, 0], 0.5I_2)$, 800 from $N([1, 1], 0.3I_2)$, 800 from $N([-1, -1], 0.3I_2)$.	1681 patterns in [3, 3]~[−3, −3]
Data3	500 from$N([0, 0], 0.5I_2)$, 500 from $N([1, 1], 0.3I_2)$, 500 from $N([-1, -1], 0.3I_2)$, 500 from $N([-1, 1], 0.3I_2)$, 500 from $N([1, -1], 0.3I_2)$.	1681 patterns in [3, 3]~[−3, −3]
Data4	800 from$N(0, 0.5)$, 800 from $N(-0.5, 1)$, 800 from $N(1, 0.3)$.	1201 patterns in [3] ~ [−3]

In order to guarantee precise approximation, the range for sampling TX is determined in a way that the range can cover virtually the whole data region where the probability is more than zero. That is, for a given probability function $g(x)$, TX covers almost all areas where $g(x)$ is not zero. In this case, we have $1 = \int_x g(x)dx \approx \sum_{tx_i \in TX} g(tx_i)\Delta tx_i$.

The data sets used in this section are shown in Table 3. In this study, these data sets were all generated in low-dimensional data domains because high dimensionality is well known for its adverse effect of reducing the reliability of Parzen window. First, these methods are compared in terms of D_{ab} and D_{KL}. The comparative results are presented in Figure 3, with a reduction ratio of 0.05, and Figure 4, with a reduction ratio of 0.1. The results in different examples and with different reduction ratios lead clearly to similar conclusions. From the perspective of the quality of data reduction results, the density-based methods—that is, the multiscale methods, REDR and WREDR–deliver similar performance. These density-based methods outperform SOM and the random sampling scheme. In Table 4, the data reduction methods are compared in terms of computational efficiency. It is noted that both REDR and WREDR are more efficient than the multiscale method. This is because the exhaustive computation on pattern-pair distance is avoided in REDR and WREDR. To sum up, among the compared methods, REDR and WREDR are clearly the best data reduction methods because they can deliver the best or nearly the best data reduction results with greater computational efficiency.

Besides, the very small deviations illustrated in Figures 3 and 4 indicate that initialization has little effect on the performance of either REDR or WREDR. Also, it can be noted that WREDR outperforms REDR in most cases. Clearly, it is due to the weighting strategy of WRE. Furthermore, REDR and WREDR are compared through t-tests in which the p-values can reflect the significance of difference between the results of REDR and



(a)

(b)

(c)

(d)

Figure 3: Comparisons on effectiveness in terms of D_{ab} and D_{KL} for the reduction ratio = 0.05. (a) Results on Data1. (b) Results on Data2. (c) Results on Data3. (d) Results on Data4. In each image, from left to right, the bars represent the results of the random sampling scheme, SOM, the multiscale method, REDR, and WREDR, respectively.

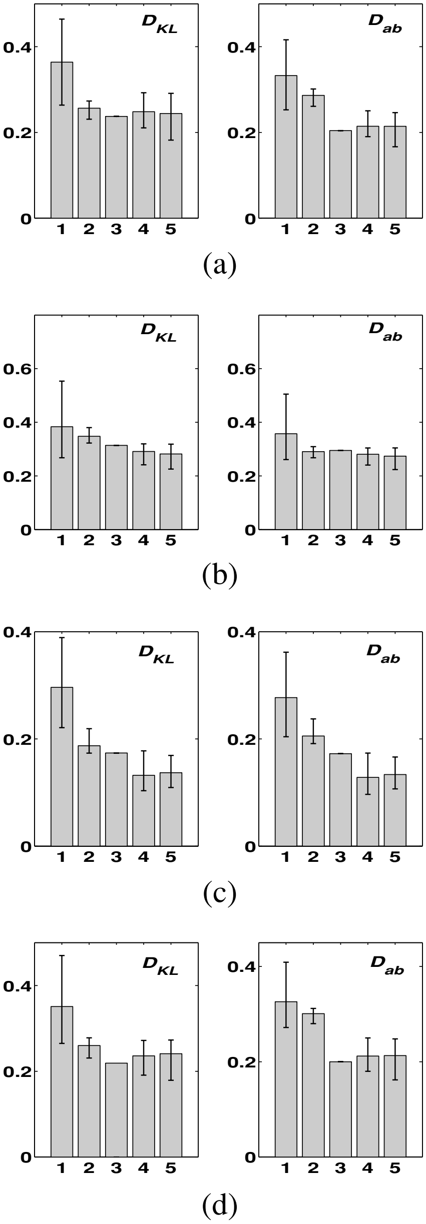


Figure 4: Comparisons on the effectiveness in terms of D_{ab} and D_{KL} for a reduction ratio $= 0.1$. (a) Results on Data1. (b) Results on Data2. (c) Results on Data3. (d) Results on Data4. In each image, from left to right, the bars represent the results of the random sampling scheme, SOM, the multiscale method, REDR, and WREDR, respectively.

Table 4: Comparisons in Terms of Running Time (in seconds).

Name of Data Set	SOM	Multiscale Method	REDR	WREDR
Reduction ratio = 0.05				
Data1	1	2	1	1
Data2	15	271	59	64
Data3	27	452	79	82
Data4	7	177	21	24
Reduction ratio = 0.1				
Data1	2	5	3	3
Data2	19	364	125	109
Data3	23	520	147	152
Data4	13	209	26	30

Table 5: Comparisons Between WREDR and REDR in Terms of D_{ab}.

	Reduction Ratio = 0.05			Reduction Ratio = 0.1		
Name of Data Set	Average of REDR	Average of WREDR	p-Value	Average of REDR	Average of WREDR	p-Value
Data1	0.235	0.235	0.61	0.161	0.161	0.89
Data2	0.193	0.184	0.35	0.280	0.273	0.34
Data3	0.142	0.138	0.16	0.128	0.130	0.76
Data4	0.195	0.184	0.34	0.305	0.299	0.45

WREDE. A small p-value means a large difference. In Table 5, the comparative results of REDR and WREDR are presented. These results show that the advantage of WRE over RE becomes significant as the reduction ratio decreases. Also, this advantage basically increases along the direction from the simple distribution, such as Data1, to a relatively complex data distribution, such as Data3.

D_{KL} and D_{ab} are known as a straightforward and accurate way to measure the representation ability of a reduced data set. Using them as reference, we evaluate the reliability of the proposed criteria RE and WRE. The values of RE (WRE), D_{KL}, and D_{ab} in each iteration of the second stage of REDR (WREDR) are recorded. The variations of D_{KL} and, D_{ab} are compared with those of REDR (WREDR). In Figures 5 and 6, the typical results on two data are illustrated. It can be seen that RE and WRE vary in a similar fashion with D_{KL}, and D_{ab}. We can thus assert that RE and WRE are reliable enough to measure the representation ability of a reduced data set.

5.2 Classification. In this section, five data reduction methods are compared: the stratified sampling scheme, the supervised SOM, the multiscale method, WREDR, and WREDR-FO. The results of the RE-based method

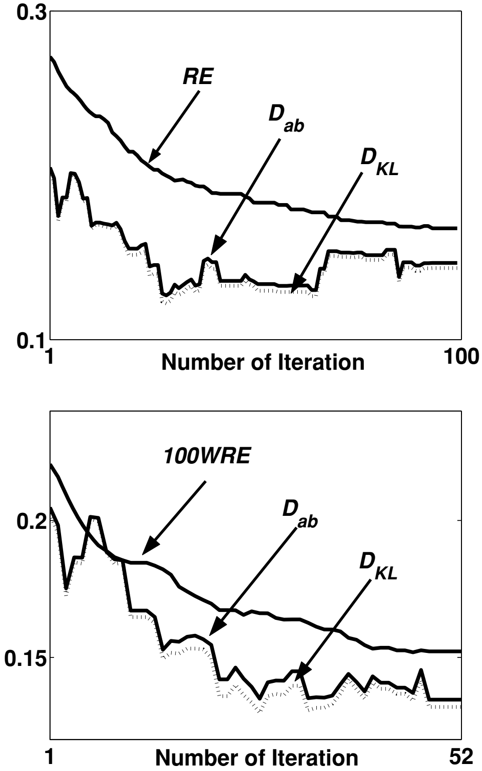


Figure 5: Typical comparison of variation between RE/WRE and density errors (i.e., D_{ab} and D_{KL}). These values are obtained in the second stage of REDR/WREDE on Data3 with reduction ratio = 0.05. For clear illustration, the values of WRE are timed by 100. Both RE and WRE are shown to vary in a similar fashion with density errors. It verifies that RE and WRE are reliable in evaluating the data reduction effectiveness.

are not presented here because they are similar to the results of the WRE-based ones. The stratified sampling scheme and the supervised SOM treat a classification data set in the same way as WREDR and WREDR-FO: with a predetermined reduction ratio, the pattern subsets of different classes are reduced separately, and the final data reduction result is the collection of the results on all the classes. The six data sets used in this section are described in Table 6. The synthetic data, which were detailed in section 4.2.2, contain 80 outliers. To evaluate a reduced data set, several popular classifiers are first built. According to the results of these classifiers on the testing data, the tested data set is evaluated. A high classification result indicates a good reduced data set. The classifiers used are the kNN

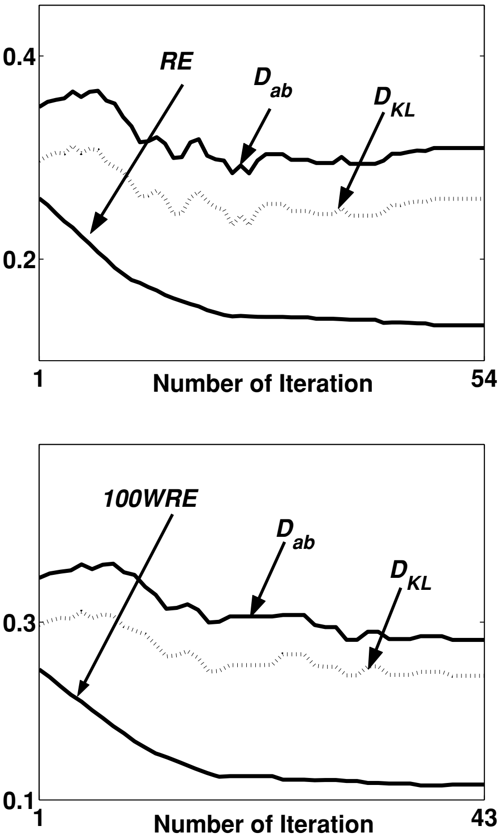


Figure 6: Typical comparison of variation between RE/WRE and density errors (i.e., D_{ab} and D_{KL}). These results are obtained in the second stage of REDR/WREDE on Data3 with a reduction ratio = 0.05.

rule with k =1 and the multilayer perceptron (MLP) (Haykin, 1999). MLP is provided in the Netlab toolbox (http://www.ncrg.aston.ac.uk/netlab). Throughout this investigation, six hidden neurons are used, and the number of output neurons is set with the number of classes so that one class distinctively corresponds to one output neuron. Also, in each example, the classification models are constructed on the entire training data set. The results of those models on the testing data are presented in Table 7.

Two reduction ratios, 0.1 and 0.3, are investigated in this study. Table 8 lists the comparative results. In the example of image segmentation, the maximal reduction ratio of the multiscale method is about 0.17, which is much less than 0.3. Thus, in this example, there is no result of the

Table 6: Data Sets used in Classification Application.

Name of Data Set	Number of Training Data Samples	Number of Testing Data Samples	Number of Features	Number of Classes
Synthetic Data	500	500	2	2
MUSK	3000	3598	166	2
Pima Indian Diabetes	500	268	8	2
Spambase	2000	2601	58	2
Statlog Image Segmentation	4435	2000	36	6
Forest Covertype[a]	50,000	35,871	54	5

[a] The original Forest Covertype data set has seven classes and more than 580,000 patterns. Under our computer environment, it is very hard to tackle the whole data set. Thus, the patterns belonging to class 1 and class 2 are omitted in our study.

Table 7: Classification Accuracy of the Models Built on the Training Data Sets.

Name of Data Set	*k*NN	MLP
Synthetic Data	0.83	0.99
MUSK	0.94	0.97
Pima	0.69	0.73
Spambase	0.90	0.92
Image Segmentation	0.89	0.85
Forest Covertype	0.95	0.85

multiscale method for a reduction ratio of 0.3. The presented results clearly indicate the advantage of WREDR-FO, which is due to the contribution of the criterion WRE and the outlier-filtering strategy. Also, referring to the results presented in Table 7, it is suggested that WREDR-FO has little effect on reducing classification accuracy. Even in the examples of the synthetic classification and the Pima Indian Diabetes classification, WREDR-FO can enhance the final classification performance, in contrast to the general argument that a reduced data set corresponds to degraded classification results (Provost & Kolluri, 1999). Clearly, using our proposed outlier-filtering strategy can compensate for the classification loss caused by data reduction to a certain degree. In addition, it is noted that WREDR-FO provides more classification enhancement to *k*NN than to MLP. This is mainly due to the fact that *k*NN may be more sensitive to noise than MLP (Khotanzad & Lu, 1990).

In Table 9, different methods are compared in terms of running time. WREDR and WREDR-FO are shown to be much more efficient than the multiscale method. Also, the computational effort required by the proposed outlier-filtering strategy is insignificant since WREDR-FO is almost as efficient as WREDR.

Table 8: Comparisons in Terms of Classification Accuracy.

Name of	Stratified Sampling		Supervised SOM		Multiscale Method		WREDR		WREDR-FO	
Data Set	*k*NN	MLP	*k*NN	MLP	*k*NN	MLP	*k*NN	MLP	*k*NN	MLP
Reduction ratio = 0.1										
Synthetic	0.83	0.93	0.92	0.89	0.86	0.93	0.91	0.98	**0.99**	**1.00**
Data	0.07	0.06	0.07	0.05			0.00	0.00	0.02	0.00
Musk	0.89	0.90	0.94	0.85	0.92	0.93	0.91	0.92	**0.94**	**0.93**
	0.02	0.02	0.01	0.00			0.01	0.01	0.01	0.01
Pima	0.66	0.68	0.70	0.71	0.68	0.73	0.68	0.71	**0.72**	**0.75**
	0.03	0.03	0.02	0.03			0.02	0.03	0.02	0.02
Spambase	0.82	0.88	0.83	0.84	0.83	0.88	0.84	0.89	**0.86**	**0.90**
	0.01	0.02	0.01	0.01			0.01	0.01	0.00	0.01
Image	0.86	0.81	0.86	0.81	0.87	0.82	0.86	0.81	**0.87**	**0.84**
segmentation	0.01	0.02	0.01	0.01			0.01	0.01	0.01	0.01
Forest	0.80	0.54	0.82	0.66	0.88	0.78	0.90	0.77	**0.91**	**0.80**
covertype	0.02	0.02	0.02	0.02			0.00	0.00	**0.01**	**0.00**
Reduction ratio = 0.3										
Synthetic	0.82	0.98	0.91	0.93	0.80	0.98	0.91	0.99	**0.98**	**1.00**
Data	0.04	0.03	0.02	0.01			0.02	0	0.01	0
Musk	0.92	0.94	0.94	0.86	0.93	0.94	0.93	0.94	**0.94**	**0.95**
	0.01	0.02	0.00	0.01			0.00	0.02	0.00	0.01
Pima	0.67	0.67	0.70	0.65	0.67	0.68	0.69	0.69	**0.72**	**0.73**
	0.04	0.04	0.03	0			0.02	0.02	0.01	0.02
Spambase	0.84	0.89	0.85	0.84	0.87	0.90	0.85	0.90	**0.88**	**0.91**
	0.01	0.01	0.01	0.00			0.01	0.00	0.01	0.00
Image	0.86	0.82	0.87	0.83	—	—	0.88	0.83	**0.88**	**0.84**
segmentation	0.01	0.01	0.01	0.01			0.01		0.01	0.01
Forest	0.84	0.70	0.85	0.71	0.92	0.79	0.92	0.84	**0.93**	**0.84**
covertype	0.01	0.00	0.00	0.00			0.00	0.00	0.00	0.00

Notes: In the cells of listing results, the upper and lower values are the mean and the standard deviation, respectively. The best result of each case is highlighted in boldface.

Table 9: Comparisons in Terms of Running Time (in seconds).

Name of Data Set	Supervised SOM	Multiscale Method	WREDR	WREDR-FO
Reduction ratio = 0.1				
Synthetic Data	0.8	2.7	0.8	0.7
Musk	153	1.1×10^3	410	479
Pima	1.2	3.3	1.4	1.6
Spam-base	15	285	35	35
Image Segmentation	43	1.7×10^3	99	107
Forest covertype	8.2×10^3	1.2×10^5	6.3×10^3	7.8×10^3
Reduction ratio = 0.3				
Synthetic Data	1.8	7.3	4.0	3.4
Musk	651	3.1×10^3	1.5×10^3	1.6×10^3
Pima	1.9	10.0	4.1	4.2
Spambase	29	760	131	133
Image Segmentation	73	—	414	459
Forest covertype	4.9×10^4	1.9×10^5	1.3×10^4	1.9×10^4

6 Conclusions

This article focuses on the study of density-based data reduction schemes because this type of data reduction technique can be widely used for tackling data analysis tasks and building data analysis model. In the conventional density-based methods, the probability density of each data point has to be estimated or analyzed. This makes these methods computationally expensive when huge data sets are given. To address this shortcoming, we propose a novel type of entropy-based data reduction criteria and a data reduction process based on these criteria. Compared with the existing density-based methods, our proposed methods exhibit higher efficiency and similar effectiveness. Also, the strategy for outlier filtering is designed. This simple and efficient strategy is immensely useful for classification tasks. Finally, it is important to note that the experimental results indicate that the proposed methods are robust to initializations.

Acknowledgments

We thank the anonymous reviewers for their useful comments. The work described in this article is fully supported by a grant from City University of Hong Kong of project, no. 7001701-570.

References

Astrahan, M. M. (1970). *Speech analysis by clustering, or the hyperphoneme method* (Stanford A I Project Memo). Palo Alto, CA: Stanford University.

Bezdek, J. C., & Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. *Int. J. Intell. Syst., 16*(12), 1445–1473.

Blum, A. L., & Langley, P. (1997). Selection of relevant feature and examples in machine learning. *Artificial Intelligence*, *97*(1–2), 245–271.

Catlett, J. (1991). *Megainduction: Machine learning on very large databases*. Unpublished doctoral dissertation, University of Sydney, Australia.

Chang, C. L. (1974). Finding prototypes for nearest neighbor classifiers. *IEEE Trans. Computers, 23*(11), 1179–1184.

Chow, T. W. S., & Wu, S. (2004). An online cellular probabilistic self-organizing map for static and dynamic data Sets. *IEEE. Trans. on Circuits and Systems I*, *51*(4), 732–747.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Dasarathy, B. V. (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques*. Los Alamitos, CA: IEEE Computer Society Press.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: Wiley.

Gates, G. W. (1972). The reduced nearest neighbor rule. *IEEE Trans. on Inform. Theory*, *IT-18*, 431–433.

Gersho, A., & Gray, R. M. (1992). *Vector quantization and signal compression*. Norwell, MA: Kluwer.

Gray, R. M. (1984). Vector quantization. *IEEE Assp Magazine, 1*, 4–29.

Friedman, J. H. (1997). Data mining and statistics: What's the connection? Available online at http://www.salford-systems.com/doc/dm-stat.pdf.

Han, J. W., Kamber, M. (2001). *Data mining: Concepts and techniques.* San Francisco: Morgan Kaufmann.

Hart, P. E. (1968). The condensed nearest neighbor rule. *IEEE. Trans. on Information Theory*, *14*, 515–516.

Haykin, S. (1999). *Neural networks: A comprehensive foundation*. Upper Saddle River, NJ: Prentice Hall.

Khotanzad, A., & Lu, J.-H. (1990). Classification of invariant image representations using a neural network. *IEEE Transactions on Signal Processing*, *38*(6), 1028–1038.

Kohonen, T. (2001). *Self-organizing maps*. London: Springer.

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Density-based multiscale data condensation. *IEEE. Trans. on PAMI*, *24*(6), 734–747.

Parzen, E. (1962). On the estimation of a probability density function and mode. *Ann. Math. Statist.*, *33*, 1064–1076.

Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery*, *2*, 131–169.

Plutowski, M., & White, H. (1993). Selecting concise training sets from clean data. *IEEE Trans. Neural Networks*, *4*(2), 305–318.

Quinlan, R. (1983). Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell, & T. M. Mitechell, (Eds.), *Machine Learning—an Artificial Intelligence Approach* (pp. 463–482) Palo Alto, CA: Tioga.

Roy, N., & McCallum, A. (2001). Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann. Available online at www.cs.umass.edu/∼mccallum/papers/active-icm/101.ps.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*, 197–227.

Scott, D. W. (1992). *Multivariate density estimation: Theory, practice, and visualization*. New York: Wiley.

Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. *Machine Learning*, *38*, 257–286.

Yang, Z. P., & Zwolinski, M. (2001). Mutual information theory for adaptive mixture models. *IEEE Trans. on PAMI*, *23*(4), 396–403.

Received December 29, 2004; accepted June 27, 2005.