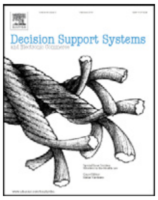
Decision Support Systems 50 (2010) 93–102

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss



The data complexity index to construct an efficient cross-validation method

Der-Chiang Li [a,*], Yao-Hwei Fang [b], Y.M. Frank Fang [c]

[a] *Department of Industrial and Information Management National Cheng Kung University, Taiwan*
[b] *Division of Biostatistics and Bioinformatics, National Health Research Institutes, Taiwan*
[c] *Geographic Information System Research Center, Feng Chia University, Taiwan*

ARTICLE INFO

Article history:
Received 20 January 2009
Received in revised form 31 March 2010
Accepted 18 July 2010
Available online 23 July 2010

Keywords:
Binary classification problem
Cross-validation
Data complexity

ABSTRACT

Cross-validation is a widely used model evaluation method in data mining applications. However, it usually takes a lot of effort to determine the appropriate parameter values, such as training data size and the number of experiment runs, to implement a validated evaluation. This study develops an efficient cross-validation method called Complexity-based Efficient (CBE) cross-validation for binary classification problems. CBE cross-validation establishes a complexity index, called the CBE index, by exploring the geometric structure and noise of data. The CBE index is used to calculate the optimal training data size and the number of experiment runs to reduce model evaluation time when dealing with computationally expensive classification data sets. A simulated and three real data sets are employed to validate the performance of the proposed method in the study, while the validation methods compared are repeated random sub-sampling validation and K-fold cross-validation. The results show that CBE cross-validation, repeated random sub-sampling validation and K-fold cross-validation have similar validation performance, except that the training time required for CBE cross-validation is indeed lower than that for the other two methods.

1. Introduction

In data mining applications, researchers generally use cross-validation to evaluate the learned classification model [11]. However, this usually requires considerable computational costs. With K-fold cross-validation, for example, the number of experiment runs must increase when parameter K increases, making the training computationally expensive [1]. Specifically, ((K − 1)/K)% training data are theoretically needed for learning a classification model, and when the data size is very large, ((K − 1)/K)% training data makes computation expensive [1].

In another common scenario, repeated random sub-sampling validation is usually repeated 30 or 50 times for model evaluation [23]. However, if the data structure is simple or uniform, the number of times sub-sampling validation is repeated is much more than what is needed, and thus the procedure is inefficient.

Our research develops an effective cross-validation procedure, called Complexity-based Efficient (CBE) cross-validation, for binary classification problems. The CBE cross-validation method can be used to calculate the optimal training data size and the number of experiment runs to reduce model validation time. The CBE cross-validation procedure systematically establishes a non-linear data complexity index (defined in Section 3) called CBE index by exploring the geometric structure and noise of data.

The density-based clustering algorithm (DBSCAN) is used to discover the geometric structure and noise, while the between-distance and within-distance of the clusters found are used as the factors of the CBE index. Based on this, this research develops an efficient CBE cross-validation procedure to calculate the optimal training data size and number of experiment runs.

The rest of this paper is organized as follows: The literature review is given in Section 2 while the detailed procedure of the proposed method is described in Section 3. One simulated and three real data sets are used to illustrate the CBE cross-validation model in Section 4, and Section 5 contains the conclusion and discussion of our research.

2. Literature review

In this section we review the concept of linear data complexity (the definition is explained in Section 3), the geometric structure and noise of data, and existing cross-validation methods.

2.1. Linear data complexity

For linear data complexity, the index used to detect the level of data complexity is Fisher's discriminant ratio f [1,10]:

$$f = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2} \tag{1}$$

where μ_1, μ_2, σ_1^2, and σ_2^2 are the means and variances of the two classes in a data set, respectively. f is specific for one feature dimension case.

* Corresponding author. Tel.: +886 6 2757575x53134.
E-mail addresses: lidc@mail.ncku.edu.tw (D.-C. Li), yhfang@nhri.org.tw (Y.-H. Fang), frankfang@gis.tw (Y.M.F. Fang).

0167-9236/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2010.07.005

For a multidimensional problem, the maximum *f* over all the feature dimensions is used to describe the problem. For problems with multidimensional features, Li and Fang proposed a Purity Level (PL) to measure linear data complexity [15]. The parameters of the index are defined as follows:

n: the number of data points. k: the number of dimensions of the data ($k \geq 2$).
A_{ij}^{+}, A_{ij}^{-}: the value of the j-th dimension of the i-th data point in the positive and negative classes, respectively.
$\bar{A}_j^{+}, \bar{A}_j^{-}$: the average value of the j-th dimension of the data in the positive and negative classes, respectively.
$A_{j\,\max}, A_{j\,\min}$: the maximum and the minimum values of the j-th dimension, respectively. Using the parameters listed above, the Purity Level is set as:

$$\text{Purity Level} = \frac{\sum_{i=1}^{n}\left(\sqrt{\frac{\sum_{j=1}^{k}\left(\frac{A_{ij}^{+}-\bar{A}_j^{-}}{A_{j\,\max}-A_{j\,\min}}\right)^2}{k-1}}+\sqrt{\frac{\sum_{j=1}^{k}\left(\frac{A_{ij}^{-}-\bar{A}_j^{+}}{A_{j\,\max}-A_{j\min}}\right)^2}{k-1}}\right)}{\sum_{i=1}^{n}\left(\sqrt{\frac{\sum_{j=1}^{k}\left(\frac{A_{ij}^{+}-\bar{A}_j^{+}}{A_{j\,\max}-A_{j\,\min}}\right)^2}{k-1}}+\sqrt{\frac{\sum_{j=1}^{k}\left(\frac{A_{ij}^{-}-\bar{A}_j^{-}}{A_{j\max}-A_{j\min}}\right)^2}{k-1}}\right)} \tag{2}$$

where the numerator is the sum of the between-class distance of the whole data set, and the denominator is the sum of the within-class distance of the whole data set. The results show that the smaller the PL value, the higher the linear data complexity, and vice versa. However, neither Fisher's discriminant ratio nor PL considers the geometric structure and noise of data.

2.2. The concept of geometric structure and noise of data

Rubinov [21] discussed the relationship between classes and clusters in data sets, and examined the distribution of classes within the obtained clusters. He found that some characteristics link data points more strongly than the classes they belong to. We thus believe that the geometric structure of data is an essential characteristic for classifying data sets.

In a study on the effect of noise in data processing, Lee et al. [14] combined the fuzzy adaptive resonance theory and the general regression neural network into a hybrid model, which assisted the removal of noise embedded in training data in order to improve the classification ability. Han et al. [9] proposed a revised Expectation-Maximization (EM) algorithm to discover and remove noise to improve the one-against-the-rest method in binary text classification. Cao et al. [2] proposed a data preprocessing method for training data to remove noise or outliers, and used the remaining data to obtain the decision function. However, the drawback of this method is that it is difficult to remove noise and outliers without the assistance of problem domain knowledge.

2.3. Common types of cross-validation method

Cross-validation is a model evaluation method that is better than residual analysis. The weakness of residual evaluation is that it does not give an indication of how well the learner will do when it is used to make predictions for unseen data. One way to overcome this problem is to leave out part of the data points from the data set when training a classifier, So that when training is finished the removed data are used to test the performance of the model. This is the basic idea for the model evaluation method called cross-validation [24]. Two widely used such methods, repeated random sub-sampling validation and K-fold cross-validation, are described below.

2.3.1. Repeated random sub-sampling validation

This method randomly splits a data set into training and validation data sets and then repeats this procedure several times. For each split, the classifier is trained with the training data and validated with the validation data. The results from each split can be averaged. This method is usually applied in small sample learning cases that use a small amount of training data to learn the model and large amount of validation data to validate it [16,17].

2.3.2. K-fold cross-validation

In K-fold cross-validation, the original sample is partitioned into K partitions. A partition is then used as the validation data for testing the model, and the remaining K − 1 partitions are used as the training data. The cross-validation process is then repeated K times, with each of the K partitions used as the validation data exactly once. The K results from the folds can be averaged to produce a single estimation [24]. The advantage of this method over the repeated random sub-sampling validation method is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used by researchers.

3. Proposed method

With binary classification problems, data complexity is defined as the level of complexity for separating data into classes. When the data complexity is high this means it is hard to classify. Complexities can be subdivided into linear and non-linear cases: linear data complexity means a complex level for separating the data using a linear hyperplane; while non-linear data complexity means a complex level for separating the data using a non-linear hyperplane. Taking the XOR problem as an example, we usually use a non-linear hyperplane to separate the data rather than a linear one.

This research focuses on finding an effective way to classify data by calculating the non-linear data complexity for high dimensional classification problems. We develop the CBE index by improving the Purity Level (PL) method [15], and consider the geometric structure and noise of data to precisely measure the level of non-linear separability. We then use the CBE index to form a sample size determination method to develop an efficient CBE cross-validation method to improve computational efficiency. The proposed Complexity-based Efficient (CBE) index is described in detail in subsection 3.1, and the proposed CBE cross-validation is described in subsection 3.2.

3.1. CBE index

Research on pattern recognition suffers from the uncertainty concerning the match between knowledge and a problem due to the strong dependence of classifying performance on available data. In other words, the accuracy of a classifier is highly dependent on the data characteristics [10]. Unfortunately, this uncertainty often remains because of a lack of understanding of the full data characteristics [18], and this situation also occurs in model validation. Therefore, in this work we consider more descriptors, such as the geometric structure and noise of data, to further understand the data characteristics with the goal of improving validation efficiency.

The CBE index relies heavily on the realization of the data's geometric structure, because, in our experience, when the center of the data belonging to a class is not located in the data cluster (such as with the XOR problem in Fig. 1), it is not reasonable to use a linear index, such as an F-test statistic or purity level, to measure the data complexity. We thus develop the non-linear CBE index to find multiple centers according to the geometric structures of data. In

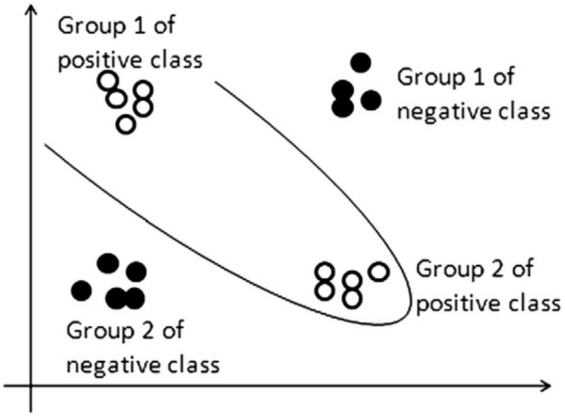


Fig. 1. The structure of a XOR problem.

that we calculate the centers of data clusters and let the centers be located in the data. Note that the linear index concept is a special case of the non-linear one when it has only one cluster in each class.

To discover the geometric structure and noise of data, researchers usually rely on prior knowledge, although this is experience oriented and inconclusive [22]. This research thus proposes a non-linear data complexity index, the CBE index, to systematically reflect the geometric structure and noise of data precisely. This study uses the density-based clustering (DBSCAN) algorithm to discover the geometric structure and noise of data to find the complexity level to separate data into classes, as explained below.

3.1.1. DBSCAN algorithm

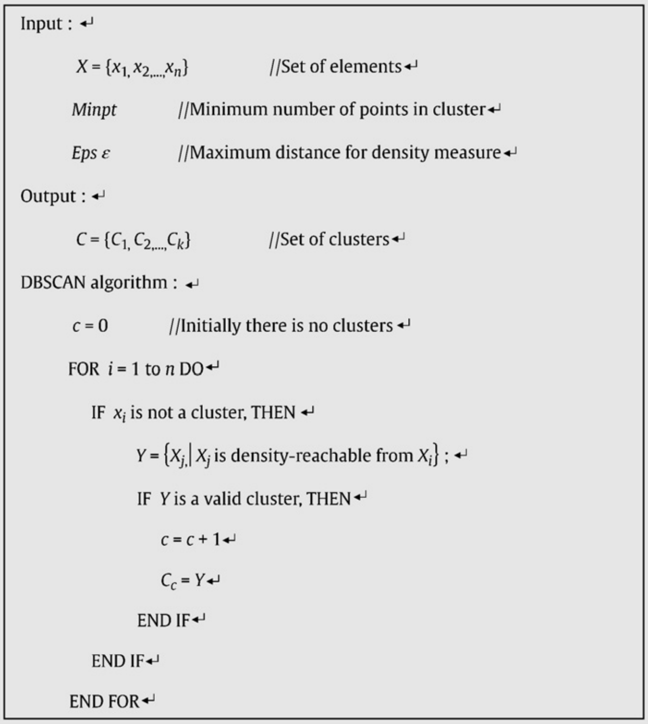
DBSCAN is a clustering algorithm suitable for a data set with a large amount of data with high dimensionality [7]. DBSCAN gathers together high density data as clusters and the shape of each cluster are arbitrary. The algorithm finds the clusters and then deletes data that does not belong to any of them. It searches for clusters by checking the surroundings of each data point within a scope called the ε-neighborhood. If the ε-neighborhood of a data point contains other data which has a data size that is more than a certain pre-defined number (MinPts), a cluster with this data (called the core object) is created; otherwise, the data is treated as noise which will be eventually deleted. DBSCAN iteratively collects directly density-reachable data (data within the ε-neighborhood of a core object) until no new data can be added to any cluster, and this may involve merging some clusters. We apply the DBSCAN algorithm to each class to detect the geometric structure and noise of data in binary classification. Table 1 shows the DASCAN algorithm pseudo code.

Consider the radius of a default ε, obtained by considering the fraction of objects to be selected (k/m) and the volume V [6]. We extend this concept to binary classification and suppose that n is the dimension of the data, k is the number of MinPts, Γ is the gamma function, m_+ and m_- are the amounts of data in the positive and negative classes, repectively, and $V_+ = \prod_j \text{range}\left(x_j^+\right)$ and $V_- = \prod_j \text{range}\left(x_j^-\right) \text{for} j = 1,..,k$ are the data ranges in the positive and negative classes, respectively. The following are the formula sets for ε_+, and ε_- for positive and negative classes, respectively:

$$\varepsilon_+ = \sqrt[n]{\frac{(k/m_+)\, V_+\, \Gamma(k/2+1)}{\sqrt{\pi^n}}} \tag{3}$$

$$\varepsilon_- = \sqrt[n]{\frac{(k/m_-)\, V_-\, \Gamma(k/2+1)}{\sqrt{\pi^n}}} \tag{4}$$

Table 1
The pseudo code of the DBSCAN algorithm.



```
Input : ↵
    $X = \{x_1, x_2, ..., x_n\}$        //Set of elements↵
    Minpt        //Minimum number of points in cluster↵
    Eps $\varepsilon$        //Maximum distance for density measure↵
Output : ↵
    $C = \{C_1, C_2, ..., C_k\}$        //Set of clusters↵
DBSCAN algorithm : ↵
    $c = 0$        //Initially there is no clusters↵
    FOR $i = 1$ to $n$ DO↵
      IF $x_i$ is not a cluster, THEN ↵
        $Y = \{X_j \mid X_j$ is density-reachable from $X_i\}$ ; ↵
        IF $Y$ is a valid cluster, THEN↵
          $c = c + 1$↵
          $C_c = Y$↵
        END IF↵
      END IF↵
    END FOR↵
```

Daszykowski et al. proposed a default MinPts calculation formula [5]. We extend this formula to binary classification and define:

$$\text{MinPts}_+ = \mathit{integer}\left(\frac{m_+}{25}\right), \text{ for a positiveclass} \tag{5}$$

$$\text{MinPts}_- = \mathit{integer}\left(\frac{m_-}{25}\right), \text{ for a negative class} \tag{6}$$

For a data set with numerous data points of positive and negative classes (m_+ or m_-), we suggest that MinPts_+ or MinPts_- be equal to 20.

3.1.2. The calculation of the CBE index

This research uses the CBE index to depict the level of non-linear data complexity. The CBE index of binary classification can be regarded as the relative distance of clusters discovered by the DBSCAN algorithm for each class, and it is found as follows:

Let $\mathbf{X}=\{\mathbf{X}_1,...,\mathbf{X}_N\}$ be a data set that includes positive samples ${}_+\mathbf{X} = \left\{{}_+\mathbf{X}_1, ..., {}_+\mathbf{X}_{n_+}\right\}$and negative samples${}_-\mathbf{X} = \left\{{}_-\mathbf{X}_1, ..., {}_-\mathbf{X}_{n_-}\right\}$, where $n_+ + n_- = N$. Let ${}_+C = \left\{{}_+\mathrm{C}_1, ..., {}_+\mathrm{C}_{|{}_+\mathrm{C}|}\right\}$ be a set that consists of $|{}_+C|$ positive clusters, ${}_-C = \left\{{}_-\mathrm{C}_1, ..., {}_-\mathrm{C}_{|{}_-\mathrm{C}|}\right\}$ be a set consisting of $|{}_-C|$ negative clusters, $\mathrm{d}(\mathbf{X}_i,\mathbf{X}_j)$ be the distance between $\mathbf{X}_i$ and $\mathbf{X}_j$, and ${}_+C_i = \left\{{}_+\mathbf{X}_1^{\mathrm{i}}, ..., {}_+\mathbf{X}^{\mathrm{i}}{}_{{}_+\mathrm{m}_\mathrm{i}}\right\}$ be the i-th positive cluster, where ${}_+m_i$ is the number of positive samples in the i-th cluster, and $i = 1, ..., |{}_+C|$. Similarly, let ${}_-C_i = \left\{{}_-\mathbf{X}_1^{\mathrm{i}}, ..., {}_-\mathbf{X}^{\mathrm{i}}{}_{{}_-\mathrm{m}_\mathrm{i}}\right\}$ be the i-th negative cluster, where ${}_-m_i$ is the number of negative samples in i-th cluster, and $i = 1, ..., |{}_-C|$. We first calculate the minimum average distance between a pair of clusters which belong to different classes as Min_Bet:

$$\text{Min\_Bet} = \underset{\substack{k=1,...,|{}_+C| \\ l=1,...,|{}_-C|}}{Min} \left\{ \frac{\sum_{i=1}^{{}_+m_k} \sum_{j=1}^{{}_-m_l} d\left({}_+\mathbf{X}_i^k, {}_-\mathbf{X}_j^l\right)}{{}_+m_k \cdot {}_-m_l} \right\} \tag{7}$$

A large value of Min_Bet indicates that the data are widely scattered and easy to classify.

We then calculate the average distance within all clusters of the positive class as:

$$\mathrm{Within}_{+} = \sum_{\mathrm{k}=1}^{|_{+}\mathrm{C}|} \frac{\sum_{i=1}^{_{+}\mathbf{m}_k}\sum_{j=1}^{_{-}\mathbf{m}_k} d\left({}_{+}\mathbf{X}_i^k, {}_{-}\ \mathbf{X}_j^l\right)}{{}_{+}\mathbf{m}_k({}_{+}\mathbf{m}_{\mathrm{k}}-1)} \tag{8}$$

and for all clusters of the negative class as:

$$\mathrm{Within}_{-} = \sum_{k=1}^{|_{+}\mathrm{C}|} \frac{\sum_{i=1}^{_{+}m_k}\sum_{i=1}^{_{-}m_k} d\left({}_{-}\mathbf{X}_i^k, {}_{-}\ \mathbf{X}_j^l\right)}{{}_{-}m_k({}_{-}m_k-1)} \tag{9}$$

If the value of the average distance within all clusters of a class (Within_{+} and Within_{-}) is small, it means that these clusters congregate with each other.

The calculation of the CBE index is defined as follows:

$$\text{CBE index} = \frac{\text{MinBet}}{\frac{\mathrm{Within}_{+} + \mathrm{Within}_{-}}{|_{+}C| + |_{-}C|}} \tag{10}$$

The determination of the CBE index takes three steps:

Step 1: Normalize the data
For different units of dimensions, the data is normalized before calculating the CBE index.
Step 2: Discover the geometric structure and noise of data
Use the DBSCAN algorithm in the binary classes with the suggested parameter settings: ε_{+},ε_{-}, MinPts_{+},and MinPts_{-}, to detect the geometric structure and remove the data noise.
Step 3: Calculate the CBE index
Calculate Min_Bet, Within_{+}, and Within_{-} to obtain the CBE index.

The CBE index has the following properties:

(1) $0 \le$ CBE index$<\infty$.
(2) The smaller the CBE index is, the higher the data complexity is.
(3) The larger the CBE index is, the lower the data complexity is.

3.2. CBE cross-validation method

We apply the CBE index to develop the CBE cross-validation method, where we first randomly select a certain small proportion (for example, 5%) of samples as the training data and calculate the CBE index. This process is repeated 30 times to calculate the averages $\overline{\mathrm{X}}_{\mathrm{CBE}}$ and the standard deviations $\mathrm{S}_{\mathrm{CBE}}$. In order to achieve a stable CBE index for the optimal training data size N, this process is iterated while increasing the proportion of the training data and checking the difference of $\overline{\mathrm{X}}_{\mathrm{CBE}}$ as:

$$\begin{aligned} &\text{When } \overline{X}_{\mathrm{CBE}}^{n\%} - \overline{X}_{\mathrm{CBE}}^{n+1\%} < 0.01, \text{THEN} \\ &N = \mathrm{Max}\{\text{no.of } n\% \text{ samples, no. of } 10\% \text{ samples}\}\cdot \text{data size} \end{aligned} \tag{11}$$

When the difference decreases by a level smaller than 0.01, we consider the structure of the training data to be stable, and use this training data size as the optimal one. Where 0.01 is only an empirical suggestion and 10% is also an empirical save low sample size limit.

For the number of experiment runs, we repeat the process 30 times to calculate the average and standard deviation of CBE. Note that the sample distribution of the CBE index will converge to a normal distribution according to the Central Limit Theorem (CLT) [3], and the optimal training data size (average $\overline{\mathrm{X}}_{\mathrm{CBE}}^{n\%}$ and standard deviations $\mathrm{S}_{\mathrm{CBE}}^{n\%}$) is used to calculate the number of experiment runs. The number of experiment runs K is determined as:

$$\begin{aligned} &\text{CALCULATE} \frac{\left(\mathrm{Z}_{\alpha/2}\right)^2 {\mathrm{S}_{\mathrm{CBE}}^{n\%}}^{2}}{\left(0.05\cdot\overline{\mathrm{X}}_{\mathrm{CBE}}n\%\right)^2} = k, \text{THEN} \\ &\mathrm{Max}\ \{k, 5\} = \mathrm{K} \end{aligned} \tag{12}$$

where α is the significance level, $\mathrm{Z}_{\alpha/2}$ is the value with $\alpha/2\%$ in the tail of the cumulative standard Normal distribution, and $\left(0.05\cdot\overline{\mathrm{X}}_{\mathrm{CBE}}^{n\%}\right)$ is set as the desired margin of error, where 5 is again our suggestion.

4. Experiment

In this section, we use one simulated and three real data sets to verify the performance of the Complexity-based Efficient (CBE) cross-validation method. In the simulation experiments, a support vector machine (SVM) [12], a Back-propagation Network (BPN) [8,20], and a Naive Bayes Classifier (NBC) [24] are used as the classification tools, while in the three real data sets, only SVM is used.

To find the relationship between CBE index and classification accuracy, we randomly select 10% of the total samples and calculate the CBE index with the suggested ε_{+},ε_{-},MinPt_{+},and MinPt_{-} in Section 3 to measure the relationship for all data sets. This process is repeated 10 times, where SVM, BPN, and NBC are used as the classifiers with the resubstitution method (all available data are used for training and testing) [13].

To implement the CBE cross-validation, we randomly select a small proportion of the data as the training set (such as 5%), and calculate the CBE indexes. This procedure is repeated 30 times. The training data size is gradually increased, where we calculate the average and the standard deviation of the CBE index in order to find the optimal training data size and the number of experiment runs.

4.1. Simulated data experiments

This research uses the Parametric Equation of a Hypersphere [16], briefly introduced below, to generate simulated data. The n-hypersphere (often simply called the n-sphere) is a generalization of an object with n dimensions in $\mathbb{R}^{\mathrm{n}}$ (the circle and sphere are called the two-sphere and three-sphere, respectively). The n-sphere centered at the origin can therefore be defined as a set of points $(x_1, x_2, \ldots, x_k)$ such that:

$$x_1^2 + x_2^2 + \ldots + x_n^2 = r^2 \tag{13}$$

Table 2
The CBE index and classification accuracies of the three classifiers for the simulated data sets with 1% noise where "Average no. of noise samples found" means for the average number of noise found, "Average no. of clusters found" means for the average number of cluster found by using DBSCAN algorithm.

CBE index	2.964	2.447	2.743	2.594	3.628	3.264	3.896	3.889	4.168	4.357
Average no. of noisy samples found	0.46	0.32	0.68	0.72	0.63	0.84	0.91	0.54	0.42	0.86
Average no. of clusters found (pos., neg.)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)	(2, 2)
Accuracy of SVM	70.25	70.56	71.11	71.45	72.42	72.56	72.44	75.35	76.55	77.89
Accuracy of BPN	70.75	71.02	71.84	71.76	72.12	72.84	75.32	75.98	76.35	78.52
Accuracy of NBC	68.14	69.45	70.12	70.85	71.34	72.81	73.56	74.38	75.92	75.45

The hypersphere can be specified in a parametric equations as:

$$\begin{cases} x_1 = r\,sin\theta_1 sin\theta_2 \cdots sin\theta_{n-1} \\ x_2 = r\,sin\theta_1 sin\theta_2 \cdots cos\theta_{n-1} \\ x_3 = r\,sin\,\theta_1 sin\theta_2 \cdots cos\theta_{n-2} \\ x_4 = r\,sin\theta_1 sin\theta_2 \cdots cos\theta_{n-3} \\ \quad\vdots \\ x_{n-1} = r\,sin\theta_1 cos\theta_2 \\ x_n = r\,cos\theta_1 \end{cases} \tag{14}$$

where r is the radius and $\theta_1, \theta_2, \ldots, \theta_{n-1} \in [0, 2\pi]$ are the angles of the hypersphere. The formula of parametric equations is not unique, but must satisfy the identity $x_1^2 + x_2^2 + \ldots + x_n^2 = 1$.

We consider the two-cluster condition in each class and insert noise into the data. We generate 808 five-dimension data (404 positive and 404 negative samples) following the Parametric Equation of a Hypersphere [16]. In the positive class, the data is generated into two clusters. One is:

$$\begin{cases} x_1 = -0.7 + sin\theta_1 \sin\theta_2 \cdots \sin\theta_4 \\ x_2 = -0.7 + \sin\theta_1 \sin\theta_2 \cdots \cos\theta_4 \\ x_3 = -0.7 + sin\theta_1 \sin\,\theta_2 \cdots \cos\theta_3 \\ x_4 = -0.7 + \sin\theta_1 \sin\theta_2 \cos\theta_2 \\ x_5 = -0.7 + \cos\theta_1 \end{cases}, 0 \le \theta \le 2\pi \tag{15}$$

and the other is:

$$\begin{cases} x_1 = -0.7 + sin\theta_1 \sin\theta_2 \cdots \sin\theta_4 \\ x_2 = 0.7 + \sin\theta_1 \sin\theta_2 \cdots \cos\theta_4 \\ x_3 = -0.7 + sin\theta_1 \sin\theta_2 \cdots \cos\theta_3 \\ x_4 = -0.7 + \sin\theta_1 \sin\theta_2 \cos\theta_2 \\ x_5 = -0.7 + \cos\theta_1 \end{cases}, 0 \le \theta \le 2\pi \tag{16}$$

In the negative class, the data is generated into two clusters too. One is

$$\begin{cases} x_1 = 0.7 + sin\theta_1 \sin\theta_2 \cdots \sin\theta_4 \\ x_2 = -0.7 + sin\theta_1 \sin\theta_2 \cdots \cos\theta_4 \\ x_3 = -0.7 + sin\theta_1 \sin\theta_2 \cdots \cos\theta_3 \\ x_4 = -0.7 + \sin\theta_1 \sin\theta_2 \cos\theta_2 \\ x_5 = -0.7 + \cos\theta_1 \end{cases}, 0 \le \theta \le 2\pi \tag{17}$$

and the other is:

$$\begin{cases} x_1 = 0.7 + sin\theta_1 \sin\theta_2 \cdots \sin\theta_4 \\ x_2 = 0.7 + \sin\theta_1 \sin\theta_2 \cdots \cos\theta_4 \\ x_3 = -0.7 + sin\theta_1 \sin\theta_2 \cdots \cos\theta_3 \\ x_4 = -0.7 + \sin\theta_1 \sin\theta_2 \cos\theta_2 \\ x_5 = -0.7 + \cos\theta_1 \end{cases}, 0 \le \theta \le 2\pi \tag{18}$$

We then add 1% noise to each class by randomly selecting 4 samples to change class label. Table 2 and Fig. 2 show the results of using the CBE index with the simulated data sets.

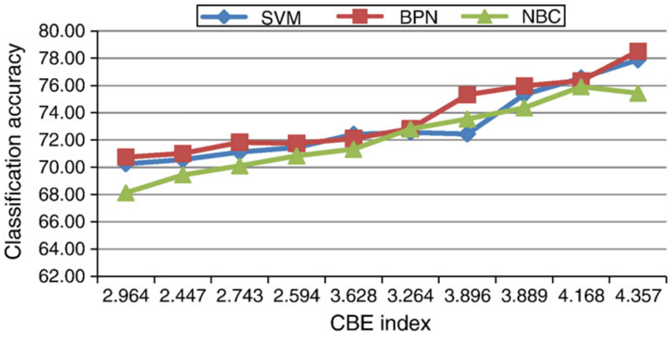


Fig. 2. The relationship between classification accuracy and the CBE index with 1% noise.

Table 3
The averages and standard deviations (SDs) of CBE indexes with increasing size of the training data sets for the simulated data set. (Bold value means the optimal data size).

Training data	40 (5%)	81 (10%)	121 (15%)	161 (20%)	202 (25%)
Average	3.836	3.474	3.408	3.071	2.684
SD	0.368	0.393	0.424	0.393	0.382
Training data	242 (30%)	283 (35%)	291 (36%)	**299 (37%)**	307 (38%)
Average	2.224	2.167	2.140	**2.129**	2.124
SD	0.236	0.287	0.296	**0.138**	0.161

From the table and figure above we can see that when the value of CBE increases, the classification accuracies of SVM, BPN, and NBC also rise. There is thus a highly positive correlation between the CBE index and classification accuracy for the simulated data sets.

To find the optimal training data size, we calculate various CBE indexes by increasing the training set size. Table 3 and Fig. 3 show the results of using the CBE index with various simulated data sets.

$$\begin{aligned} &\text{WHEN } \overline{X}_{\text{CBE}}^{37\%} - \overline{X}_{\text{CBE}}^{38\%} < 0.01, \text{ THEN} \\ &\text{Max}\{\text{no. of 37\% samples, no.of 10\% samples}\} \cdot \text{ data size} \\ &\quad = 37\% \cdot 808 \\ &\quad = 299 \end{aligned} \tag{19}$$

We determine that the optimal training data size is 299 when $CBE_{\overline{X}}$ decreases by less than 0.01, and consider that the geometric structure of the optimal training data is stable.

To find the optimal number of experiment runs for the simulated data set. We use the optimal sample size to measure the optimal experiment runs as:

$$\begin{aligned} &\text{CALCULATE} \frac{\left(Z_{\alpha/2}\right)^2 0.138^2}{(0.05 \cdot 2.129)^2} = 6.426, \text{THEN} \\ &\text{Max}\{6.456, 5\} \\ &\quad = 6 \end{aligned} \tag{20}$$

In the simulated data set, with a significance level $\alpha = 0.05$ and a margin of error of 0.1065, the optimal number of training data is 299, and the optimal number of experiment runs is six.

We use repeated random sub-sampling validation (with 533 (66%) training data points, 275 (34%) testing data points, experiment repeated 30 times) to validate that our CBE cross-validation (with 299 (37%) training data points, 509 (63%) testing data points, experiment repeated six times) is efficient. The average and standard deviations of the SVM with the repeated random sub-sampling validation are 78.326 and 1.044, respectively; and of the CBE cross-validation are 77.779 and 1.145. The performances of the two cross-validation methods thus have insignificant differences (the P-value is 0.125,

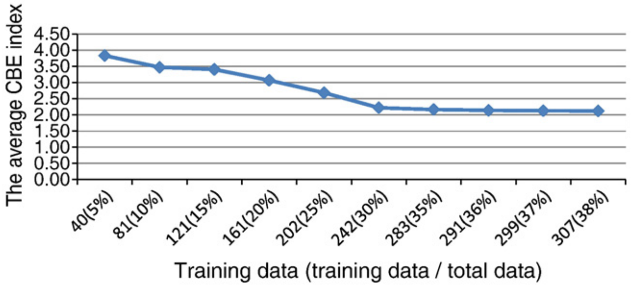


Fig. 3. Relationship between training size and the CBE index with the simulated data set.

Table 4
Properties of the three data sets.

Data set	No. of dimensions	No. of samples	No. of classes
Pima Indians diabetes	8	768	2
Haberman's survival	3	306	2
Australian credit approval	14	690	2

using the independent *t*-test). The average training time of the repeated random sub-sampling validation is 0.89*30 = 26.7 s, and that of the CBE cross-validation is 0.51*6 = 3.1 s. We also use five-fold cross-validation to validate that our CBE cross-validation is efficient. The average and standard deviations of the SVM with five-fold cross-validation are 78.454 and 1.141, respectively. The performances of the cross-validation methods have insignificant differences (the *P*-value is 0.138, using the independent *t*-test) and the average training time of the five-fold cross-validation is 0.99*6 = 5.94 s.

In addition, when we use 10% of the total data (the lower bound of the training data size) as the training data, and five experiments runs (the lower bound of the experiment runs), the average and standard deviations of SVM are 73.779 and 2.563, with a significant difference (lower) compared to CBE cross-validation (the *P*-value<<0.01, using the independent *t*-test). The average training data is 0.32*5 = 1.68 s. Since validation effectiveness is the bssic concern of researchers, the CBE cross-validation is thus considered to be better than the cross-validation using the lower bound of the training data size and experiment runs, and so it is an efficient and effective method.

4.2. Real data experiment

This research uses two medical data sets, Pima Indians Diabetes and and Haberman's Survival, and one business data set, Australian Credit Approval, in the experiment. The Pima Indians diabetes data set consists of 768 data with eight numeric dimensions (attributes), and it is a two-class data set with target values denoted by 0 and 1. The class value 1 means tested positive for diabetes, and the class value 0 means tested negative. The Haberman's Survival data set consists of 306 data with three numeric dimensions, and it is a two-class data set to record the survival status for breast cancer patients. The Australian Credit Approval data set consists of 690 data with 14 dimensions that include six numerical and eight categorical data, and it is a two-class data set. Table 4 shows the summary of the sample characteristics of the three data sets, which are all downloaded from the UCI repository, available at http://www.ics.uci.edu. The results of the experiment for the three data sets are shown in the following subsection.

4.2.1. The Pima data set

The relationship between the CBE indexes and classification accuracies is shown in Table 5 and Fig. 4.

From the table and figure above we can see that when the value of CBE decreases, the classification accuracy of the SVM also falls. There is thus a highly positive correlation between the CBE index and classification accuracy for the Pima data set. Table 6 and Fig. 5 show the experimental results of CBE cross-validation for the Pima data set.

$$\begin{aligned} &\text{WHEN } \overline{X}_{\text{CBE}}^{13\%} - \overline{X}_{\text{CBE}}^{14\%} < 0.01, \text{ THEN} \\ &\text{Max}\{\text{no. of 13\% samples, no. of 10\% samples}\} \cdot \text{ data size} \\ &\quad = 13\% \cdot 768 \\ &\quad = 100 \end{aligned} \tag{21}$$

Table 5
Pima data set with 77 selected samples as the training data (default MinPt = 3).

Accuracy	63.75	71.25	68.75	75	76.25	78.75	82.5	85	81.25	86.25
CBE index	1.063	1.089	1.098	1.123	1.124	1.163	1.176	1.215	1.221	1.481

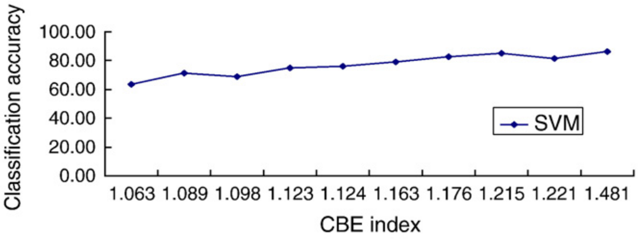


Fig. 4. Relationship between CBE indexes and accuracies with the Pima data set (correlation coefficient = 0.773).

Table 6
The averages and standard deviations (SDs) of CBE indexes with increasing size of the training data sets for the Pima data set. (Bold value means the optimal data size).

Training data	38 (5%)	46 (6%)	54 (7%)	61 (8%)	70 (9%)
Average	1.536	1.474	1.428	1.481	1.444
SD	0.260	0.292	0.321	0.293	0.287
Training data	77 (10%)	84 (11%)	92 (12%)	**100 (13%)**	108 (14%)
Average	1.202	1.177	1.170	**1.128**	1.123
SD	0.137	0.107	0.106	**0.026**	0.061

We determine this size as the optimal training data size to be 100, and thus consider that the geometric structure of the optimal training data is stable.

We use the optimal sample size to calculate the optimal number of experiment runs with the Pima data set as:

$$\begin{aligned} &\text{CALCULATE} \frac{\left(Z_{\alpha/2}\right)^2 0.026^2}{(0.05 \cdot 1.128)^2} = 0.815, \text{THEN} \\ &\text{Max}\{0.815, 5\} \\ &\quad = 5 \end{aligned} \tag{22}$$

where $\alpha = 0.05$ is the significance level, and $(0.05 \cdot 1.128) =$ 0.0564 is the desired margin of error. We thus determine that the optimal number of experiment runs to be five.

We then use repeated random sub-sampling validation (with 507 (66%) training data points, 261 (34%) testing data points, experiment repeated 30 times) to validate that our CBE cross-validation (with 100 (13%) training data points, 668 (87) testing data points, experiment repeated five times) is efficient. The average and standard deviations of the SVM with the repeated random sub-sampling validation are 76.578 and 1.743, respectively; and of the CBE cross-validation are 74.192 and 2.044. The performances of the two cross-validation methods have insignificant differences (the *P*-value is 0.043, using the independent *t*-test). The average training time of the repeated random sub-sampling validation is 0.91*30 = 27.3 s and the average training time of the CBE cross-validation is 1.9*5 = 6.2 s.

We also use five-fold cross-validation to validate that our CBE cross-validation is efficient. The average and standard deviations of the SVM with the five-fold cross-validation are 75.824 and 1.874, respectively. The performances of the two cross-validation methods have insignificant differences (the *P*-value is 0.052, using the independent *t*-test). The average training time of the five-fold cross-validation is 3.16*5 = 15.8 s.

In addition, when we use 10% of the total data (the lower bound of the training data size) as the training data with five experiment runs

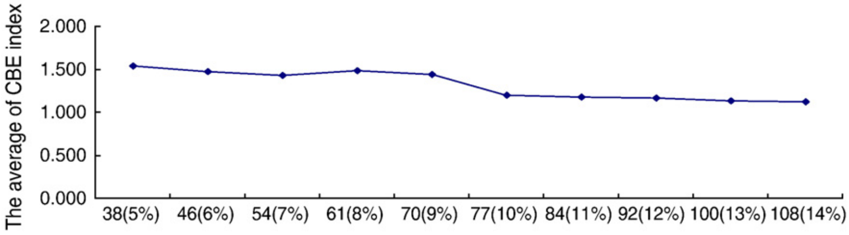


Fig. 5. Relationship between training size and CBE index with the Pima data set.

(the lower bound of the experiment runs), the average and standard deviations of SVM are 72.731 and 2.942, and it has significant differences with CBE cross-validation (the *P*-value = 0.055, using the independent *t*-test). The average training data is 1.69*5 = 8.4 s. The CBE cross-validation is better than the cross-validation using the lower bounds of the training data size and experiment runs. Therefore, CBE cross-validation is considered an efficient and effective method.

4.2.2. The Haberman data set

The relationship between the CBE indexes and classification accuracies is shown in Table 7 and Fig. 6.

From the table and figure above we can see that when the value of CBE decreases, the classification accuracy of the SVM also falls. There is thus a highly positive correlation between the CBE index and classification accuracy for this data set. Table 8 and Fig. 7 show the results of CBE cross-validation for the Haberman data set.

$$\begin{aligned}&\text{WHEN}\overline{X}^{33\%}_{\text{CBE}} - \overline{X}^{34\%}_{\text{CBE}} < 0.01,\ \text{THEN}\\&\text{Max}\{\text{no.of 33\% samples, no.of10\% samples}\}\cdot\ \text{data size}\\&\quad = 33\%\cdot 306\\&\quad = 101\end{aligned} \tag{23}$$

Using the above equation, we determine the optimal training data size to be 101. With that, we consider the geometric structure of the optimal training data is stable.

By a similar procedure, the optimal number of experiment runs is:

$$\begin{aligned}&\text{CALCULATE}\frac{\left(Z_{\alpha/2}\right)^2 0.019^2}{(0.05\cdot 1.132)^2} = 9.973,\text{THEN}\\&\text{Max}\{9.973,\ 5\}\\&\quad \approx 10\end{aligned} \tag{24}$$

where $\alpha = 0.05$ is the significance level, and $(0.05\cdot 1.132) = 0.0566$ is the desired margin of error. We thus determine the optimal number of experiment runs to be 10.

We then use repeated random sub-sampling validation (with 204 (66%) training data points, 104 (34%) testing data points, experiment repeated 30 times) to validate that our CBE cross-validation (with 101 (33%) training data points, 205 (67%) testing data points, experiment repeated 10 times) is efficient. The average and standard deviations of the SVM with the repeated random sub-sampling validation are 74.027 and 3.219, respectively, and the average and standard deviations of the SVM with the CBE cross-validation are 73.058 and 2.024. The performances of the two cross-validations have insignificant differences (the *P*-value is 0.379, using the independent t-test). The average training time of the repeated random sub-sampling validation is 0.33*30 = 9.9 s, while that of the CBE cross-validation is 0.23*10 = 2.3 s.

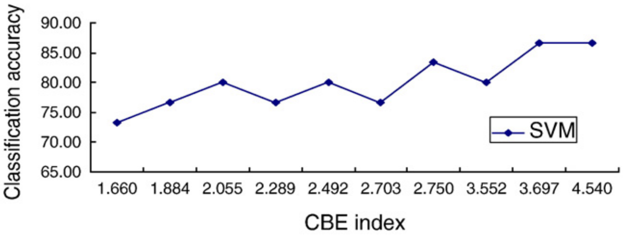


Fig. 6. Relationship between CBE indexes and accuracies with the Haberman data set (correlation coefficient = 0.827).

We then use 10-fold cross-validation to validate that our CBE cross-validation is efficient. The average and standard deviations of SVM with the five-fold cross-validation are 75.124 and 2.168, respectively. The performances of the two cross-validation methods have insignificant differences (the *P*-value is 0.075, using the independent t-test). The average training time of the 10-fold cross-validation is 0.512*10 = 5.12 s.

In addition, when we use 10% of the total data (the lower bound of the training data size) as the training data with five experiment runs, the average and standard deviations of the SVM are 72.913 and 3.641, and it has significant differences with the CBE cross-validation (the *P*-value << 0.01, using the independent t-test). The average training data is 0.18*5 = 0.9 s. By considering validation effectiveness, the CBE cross-validation is thus again considered better than the cross-validation using the lower bounds of training data size and experiment runs. Therefore, CBE cross-validation is an efficient and effective method.

4.2.3. The Australian credit approval

First, for numerical independent variables analysis, we delete the categorical independent variables $\mathbf{X}_1, \mathbf{X}_4, \mathbf{X}_8, \mathbf{X}_9, \mathbf{X}_{11}, \text{and}\mathbf{X}_{12}$ and delete the data that have missing value. The relationship between the CBE indexes and classification accuracies is shown in Table 9 and Fig. 8.

From the table and figure above we can see a highly positive correlation between the CBE index and classification accuracy.

Table 7
Haberman data set with 31 samples selected as the training data (Default MinPt = 2).

Accuracy	73.3	76.67	80	76.67	80	76.67	83.33	80	86.67	86.67
CBE index	1.66	1.884	2.055	2.289	2.492	2.703	2.75	3.552	3.697	4.54

Table 8
The averages and standard deviations (SD) of CBE indexes with increasing the size of the training data set for the Haberman data set. (Bold value means the optimal data size).

Training data	42 (14%)	52 (17%)	61 (20%)	70 (23%)	73 (24%)	77 (25%)	80 (26%)	83 (27%)
Average	2.532	2.298	2.417	2.165	2.092	2.105	2.066	2.019
SD	1.104	0.706	0.874	0.361	0.692	0.518	0.726	0.512
Training data	86 (28%)	89 (29%)	92 (30%)	95 (31%)	97 (32%)	**101 (33%)**	104 (34%)	
Average	1.452	1.349	1.342	1.362	1.287	**1.131**	1.122	
SD	0.395	0.292	0.288	0.315	0.255	**0.091**	0.057	

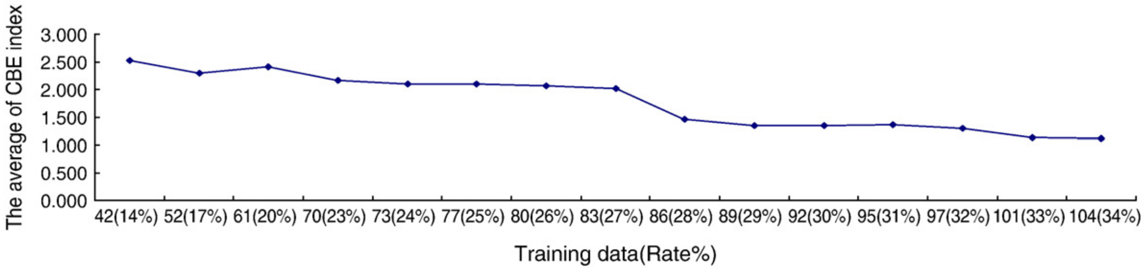


Fig. 7. Relationship between training size and the CBE index with the Haberman data set.

Table 9
Australian data set with 1,902 samples selected as training data (Default MinPt = 3).

Accuracy	68.12	68.44	69.4	70.37	70.05	71.01	71.18	71.82	71.66	72.62
CBE index	2.868	2.998	3.042	3.059	3.572	3.868	4.467	4.867	4.96	5.021

Table 10 and Fig. 9 show the results of CBE cross-validation for the Australian data set.

$$\begin{aligned}&\text{WHEN}\overline{X}_{\text{CBE}}^{42\%}-\overline{X}_{\text{CBE}}^{43\%}<0.01,\ \text{THEN}\\&\text{Max}\{\text{no.of 37\% samples, no.of 10\% samples}\}\cdot\text{data size}\\&\quad=42\%\cdot 690\\&\quad=290\end{aligned}\tag{25}$$

By a similar procedure, we determine the optimal number of training data points to be 290, and measure the optimal number of experiment runs as:

$$\begin{aligned}&\text{CALCULATE}\frac{\left(Z_{\alpha/2}\right)^{2}0.083^{2}}{(0.05\cdot 2.231)^{2}}=2.127,\ \text{THEN}\\&\text{Max}\{2.127,5\}\\&\quad=5\end{aligned}\tag{26}$$

where $\alpha = 0.05$ is the significance level, and $(0.05\cdot 2.231) = 0.1116$ is the desired margin of error. We determine the optimal number of experiment runs to be five.

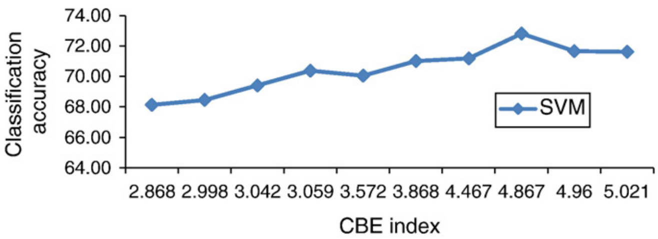


Fig. 8. Relationship between CBE indexes and accuracies with the Australian data set (correlation coefficient = 0.892).

Again, when we use repeated random sub-sampling validation (with 455 (66%) training data points, 235 (34%) testing data points, experiment repeated 30 times) to validate that our CBE cross-validation (with 290 (42%) training data points, 400 (58%) testing data points, experiment repeated 5 times) is efficient. The average and standard deviations of the SVM with the repeated random sub-sampling validation are 79.17 and 1.302, respectively, and the average and standard deviations of the SVM with the CBE cross-validation are 77.870 and 1.504. The performances of the two cross-validations have insignificant differences (the *P*-value is 0.305, using the independent *t*-test). The average training time of the repeated random sub-sampling validation is 1.83*30 = 54.9 s, and that of the CBE cross-validation is 1.84*5 = 9.2 s.

When we use five-fold cross-validation to validate CBE cross-validation, the average and standard deviations of SVM with the five-fold cross-validation are 79.2 and 1.351, respectively. Thus, the performance of the two cross-validation methods has insignificant differences (the *P*-value is 0.333, using the independent *t*-test). The average training time of the five-fold cross-validation is 1.98*5 = 9.9 s.

In addition, using 10% of the total data (the lower bound of the training data size) as training data with five experiment runs, the average and standard deviations of SVM are 74.124 and 2.169, showing significant differences with the CBE cross-validation (the *P*-value << 0.01, using the independent t-test). The average training data is 1.49*5 = 7.5 s. Similarly, the CBE cross-validation is better than the cross-validation using the lower bounds of the training data size and experiment runs. Therefore, CBE cross-validation is an efficient and effective method.

4.3. Discussion of CBE index for various data characteristics

In this subsection, we apply sensitivity analysis to the calculation of the CBE index using unbalanced classes, dimensions, and sample sizes of a data set as the attributes.

4.3.1. Unbalanced class

Nguyen and Yonggwan proposed that the accuracy of classifiers goes down as the unbalanced level increases. Specifically, they used

Table 10
The averages and standard deviations (SD) of CBE indexes with increasing the size of the training data set for the Australian data set. (Bold value means the optimal data size).

Training data	23 (10%)	138 (20%)	173 (25%)	207 (30%)	242 (35%)
Average	3.051	2.873	2.651	2.501	2.371
SD	0.397	0.185	0.146	0.2	0.139
Training data	276 (40%)	283 (41%)	**290 (42%)**	296 (43%)	
Average	2.283	2.256	**2.231**	2.225	
SD	0.097	0.089	**0.083**	0.063	

SVM as the classification tool and found that it was affected by the unbalanced effect [19]. In our experiments, we first consider the unbalanced class characteristic of a data set with the same data structure. We generate data sets by fixing the positive sample size and increasing the negative sample size, and the results are shown in Table 11. Table 11 shows that the higher the unbalanced level, the higher the data complexity and the lower the CBE index.

4.3.2. Dimensions

For a fixed sample size, adding dimensions will degrade the performance (high data complexity) of a classifier if the number of training data points is small relative to the number of dimensions [4]. For the second characteristic, a fixed sample size of 50 is used. When increasing the number of dimensions with the same data structure, given that the number of training data is smaller than the number of dimensions in the experiments, the results are obtained and shown in Table 12. Table 12 shows that when the dimensions are high, the data complexity is also high, while the CBE index is low.

4.3.3. Sample size

For the third characteristic in our experiments, we use the same sample sizes for both classes, and these are increasing with the same structure. The results are shown in Table 13.

Table 13 shows that when the samples of both classes increase, the data complexity stays the same, as does the CBE index.

Table 11
Sensitivity analysis of the CBE index for unbalanced data sets.

	Positive samples	Negative samples	MinPts	CBE index
Case1	100	100	8	2.122
Case2	100	150	10	2.087
Case3	100	200	12	1.887
Case4	100	250	14	1.653
Case5	100	300	16	1.481

Table 12
Sensitivity analysis of the CBE index for various data dimensions.

	No. of dimensions	MinPts	CBE index
Case 1	50	2	1.501
Case 2	60	2	1.366
Case 3	70	2	1.318
Case 4	80	2	1.302
Case 5	90	2	1.296

Table 13
Sensitivity analysis of the CBE index for various sample sizes of both classes.

	Positive samples	Negative samples	MinPts	CBE index
Case 1	100	100	8	2.122
Case 2	150	150	12	2.116
Case 3	200	200	16	2.113
Case 4	250	250	20	2.112
Case 5	300	300	20	2.112

5. Conclusion and discussions

Our research develops an efficient and effective cross-validation method called Complexity-based Efficient (CBE) cross-validation. The CBE cross-validation uses the CBE index (calculated by exploring the data's geometric structure and noise) to precisely discover the data's characteristics and its non-linear complexity, in order to help understand the data set. We also employ the CBE index to calculate the optimal training data size and number of experiment runs. CBE cross-validation aims to reduce model evaluation time when a complex and computationally expensive classifier is used.

We expect that when we apply CBE cross-validation to real binary data sets, we can use the proposed method to find the optimal training data and the number of experiment runs, to help researchers to develop more precise classification tools with less evaluation time. Thus this work can assist researchers in developing new classification tools.

The threshold criterion of $\overline{X}^{n\%}_{\mathrm{CBE}} - \overline{X}^{n+1\%}_{\mathrm{CBE}}$, the lower bound sample size of 0.01, and the lower bound of experiment runs of five are empirical values, that we hope to find theoretical values in future studies. With regard to the setting of the threshold criterion of the lower bound, we consider that when the number of data is large, we do not want to use too few data for the analysis, even though the data is easy to classify, because the information lost could be significant, and thus it is very difficult to convince decision makers intuitively. Besides, when we use these low limits, we are indicating that there are about 40% of the whole data that have the chance to be selected as the training data $\left(1-\left(1-{}^{1}/_{10}\right)^{5}\approx 40\%\right)$.

As to the experiment being repeated 30 times, we consider that the CBE distribution will normally converge to a normal distribution when n is large. As a matter of convenience, we thus use 30 times to approximate a normal distribution. In fact, one may need to use Q-Q plot to check if the statistics (accuracy) does in fact follow a normal distribution.

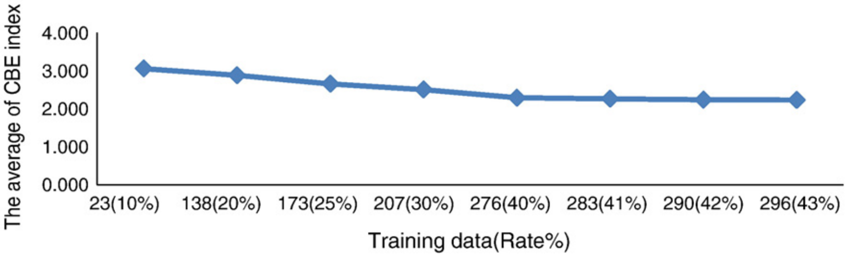


Fig. 9. Relationship between training size and CBE index of Australian data set.

CBE cross-validation is a binary classification validation method. However, multi-class classification problems are very common in both studies and real-world applications. Therefore, the study of CBE cross-validation with multiple classes is also considered as one direction for future research.

References

[1] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[2] L.J. Cao, H.P. Lee, W.K. Chong, Modified support vector novelty detector using training data with outliers, Pattern Recognition Letters 24 (2003) 2479–2487.

[3] G. Casella, R.L. Berger, Statistical Inference, second edition, Duxbury, 2002.

[4] R. Clarke, H.W. Ressom, A. Wang, J. Xuan, M.C. Liu, E.A. Gehan, Y. Wang, The properties of high-dimensional data spaces: implications for exploring gene and protein expression data, Nature Reviews. Cancer 8 (1) (2008) 37–49.

[5] M. Daszykowski, B. Walczak, D.L. Massart, Looking for natural patterns in data part 1. density-based approach, Chemometrics and Intelligent Laboratory Systems 56 (2) (2001) 83–92.

[6] M. Daszykowski, B. Walczak, D.L. Massart, Representative subset selection, Analytica Chimica Acta 468 (2002) 91–103.

[7] M. Ester, H.P. Kriegel, J. Sander, X. Xu.,, A density-based algorithm for discovering clusters in large spatial databases with noisy, Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining, Portland, 1996, pp. 226–231.

[8] M.T. Hagan, H.B. Demuth, M. Beale, Neural Network Design, Thomson, Singapore, 1996.

[9] H. Han, Y. Ko, J. Seo, Using the revised EM algorithm to remove noisy for improving the one-against-the-rest method in binary text classification, Information Processing and Management 43 (5) (2007) 1281–1293.

[10] T.K. Ho, A data complexity analysis of comparative advantages of decision forest constructors, Pattern Analysis and Applications 5 (2002) 102–112.

[11] M.Y. Hu, M. Shanker, G.P. Zhang, M.S. Hung, Modeling consumer situational choice of long distance communication with neural networks, Decision Support Systems 44 (4) (2008) 899–908.

[12] V.N. Vapnik, The Nature of Statistical Learning Theory, second editionSpringer, New York, 2000.

[13] M. Kantardzic, Data Mining: Concept, Model, Method, and Algorithms, Wiley-Interscience, 2003.

[14] E.W.M. Lee, Y.Y. Lee, C.P. Lim, C.Y. Tang, Application of a noisy classification technique to determine the occurrence of flashover in compartment fires, Advanced Engineering Informatics 20 (2006) 213–222.

[15] D.C. Li, Y.H. Fang, An algorithm to cluster data for efficient classification of support vector machines, Expert Systems with Applications 34 (2008) 2013–2018.

[16] D.C. Li, Y.H. Fang, A non-linearly virtual sample generation technique using cluster discovery and parametric equations of hypersphere, Expert Systems with Applications 36 (2009) 844–851.

[17] D.C. Li, C.W. Yeh, T.I. Tsai, Y.H. Fang, Susan C. Hu, Acquiring knowledge with limited experience, Expert Systems 24 (3) (2007) 162–170.

[18] E.B. Mansilla, On classifier domains of competence, Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04), 2004.

[19] H.V. Nguyen, W. Yonggwan, Classification of unbalanced medical data with weighted Regularized Least Squares, Proceedings of the Frontiers in the Convergence of Bioscience and Information Technologies (IEEE), 2007, pp. 347–352.

[20] S. Piramuthu, M.J. Shaw, J.A. Gentry, A classification approach using multi-layered neural networks, Decision Support Systems 11 (5) (1994) 509–525.

[21] A.M. Rubinov, N.V. Soukhorkova, J. Ugon, Classes and clusters in data analysis, European Journal of Operational Research 173 (2006) 849–865.

[22] C. Schaffer, Technical note: selecting a classification method by cross-validation, Machine Learning 13 (1993) 135–143.

[23] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, 1st edition, Pearson Addison, Wesley, Boston, 2006.

[24] I.H. Witten, Eibe was presented as. first name and Frank as.surname. Please check if. appropriate.Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Second editionMorgan Kaufman, Amsterdam, 2005.

Der-Chiang Li is a Distinguished Professor in the Department of Industrial and Information Management, the National Cheng Kung University, Taiwan. He received his Ph.D. degree at the Department of Industrial Engineering at Lamar University Beaumont, Texas, USA, in 1985. As a research professor, his current interest concentrates on learning with small data sets.

Yao-Hwei Fang is a postdoctoral fellow in the Division of Biostatistics and Bioinformatics, National Health Research Institutes. He is working at the laboratory for statistical analysis of human genetic. He received his Ph.D. at the Department of Industrial and Information Management at National Cheng Kung University, Taiwan, in 2009.

Y.M. Frank Fang obtained his PhD degree from the Department of Civil and Hydraulic Engineering, Feng Chia University in 2006. Before he joined the Department of Civil and Hydraulic Engineering of Feng Chia University (FCU) in 2006, he worked as a post doctoral researcher in Geographic Information Systems Research Center, Feng Chia University. Currently, Assistant Professor Fang is Chief Researcher of Geographic Information Systems Research Center, FCU. His research interests include disaster Monitoring and civil engineering.