Pattern Recognition 42 (2009) 1914 – 1921

Contents lists available at ScienceDirect

Pattern Recognition

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

Feature selection based on loss-margin of nearest neighbor classification

Yun Li[a,*], Bao-Liang Lu[b]

[a]Institute of Computer Technology, Nanjing University of Posts and Telecommunications, 66 Xinmofan Rd, Nanjing 210003, P.R. China
[b]Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Rd, Shanghai 200240, P.R. China

ARTICLE INFO

Article history:
Received 6 March 2008
Received in revised form 14 August 2008
Accepted 6 October 2008

Keywords:
Feature selection
Loss function
Margin
Energy-based model

ABSTRACT

The problem of selecting a subset of relevant features is classic and found in many branches of science including—examples in pattern recognition. In this paper, we propose a new feature selection criterion based on low-loss nearest neighbor classification and a novel feature selection algorithm that optimizes the margin of nearest neighbor classification through minimizing its loss function. At the same time, theoretical analysis based on energy-based model is presented, and some experiments are also conducted on several benchmark real-world data sets and facial data sets for gender classification to show that the proposed feature selection method outperforms other classic ones.

1. Introduction

In a great variety of fields, including pattern recognition and machine learning, the input data are represented by a very large number of features, but only few of them are relevant for predicting the label. In addition, many algorithms become computationally intractable when the dimension is high. On the other hand, once a good small set of features has been chosen, even the most basic classifiers (e.g. 1-nearest neighbor, 1NN) can achieve desirable performance. Therefore feature selection, i.e. the task of choosing a small subset of features which is sufficient to predict the target labels well, is critical to minimize the classification error. At the same time, feature selection also reduces training and inference time and leads to better data visualization, reduction of measurement and storage requirements.

Roughly speaking, feature selection algorithms have two key problems: search strategy and evaluation criterion. According to the criterion, feature selection algorithms can be categorized into filter model and wrapper model [1,2]. In the wrapper model, the feature selection method tries to directly optimize the performance of a specific predictor (classification or clustering algorithm). The main drawback of this method is its computational deficiency. In the filter model, the feature selection is done as a preprocessing, without trying to optimize the performance of any specific predictor directly. This is usually achieved through an evaluation function and a search strategy is used to select a feature subset that maximizes this evaluation function. Refs. [1,3,4] have given a comprehensive discussion of feature selection methodologies.

Specifically, in this paper we introduce a feature selection algorithm for nearest neighbor classification using Euclidean distance. The proposed algorithm takes advantage of the performance increasing of wrapper model whilst avoiding its computational complexity. *k*-nearest neighbor (*k*NN) rule [5], which classifies each unlabelled example by the majority label among its *k*NN in the training set, is one of the oldest and simplest methods for pattern classification and it is one of the top 10 algorithms in data mining [6]. Nevertheless, it often yields competitive results. On the other hand, margin [7,8] is a geometric measure for evaluating the confidence of a classifier with respect to its decision. Margin already plays a crucial role in machine learning research, and it is used both for theoretic analysis of generalization bounds and as guidelines for algorithm designs. In the paper, along with the guidelines of margin, a feature selection evaluation criterion based on loss function of nearest neighbor classification is proposed, and it usually can guarantee good performance for any feature search strategy. Although we focus on the nearest neighbor classification, however, most of results are relevant to other distance-based classifiers (e.g. support vector machine, SVM [8]) as well.

The novelties of this paper are the use of large margin principle together with loss function to rank the features, and the presentation of theoretic proof based on energy-based model (EBM) [12–14]. Feature ranking is a filter method: it is a preprocessing step, independent of the choice of the predictor. Still, under certain independence or orthogonal assumptions, it may be optimal with respect to a given predictor. Even when feature ranking is not optimal, it may be preferable to other feature subset selection methods because of its computational and statistical scalability: In terms of computation, it is efficient since it requires only the computation of n feature scores and sorting the scores. With respect to statistic, it is robust against

* Corresponding author. Tel.: +86 25 83492450; fax: +86 25 83492450.
E-mail addresses: yuncloudlee@gmail.com, liyun@njupt.edu.cn (Y. Li).

0031-3203/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2008.10.011

over-fitting because it introduces bias but it may have considerably less variance [3].

The paper is organized as follows: new evaluation criterion based on loss function of kNN rule are introduced and analyzed in Section 2. In Section 3, we propose a new feature selection method with theoretic analysis based on EBM. Experimental results and analysis are shown in Section 4. The paper ends with conclusions and discussions in Section 5.

2. Evaluation criterion

Assuming the training set S contains N samples, $\{x_i, y_i\}_{i=1}^N$, and each sample x_i is represented by an n-dimensional feature vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \in \mathscr{R}^n$ and discrete class labels y_i. We first introduce some definitions as follows.

Definition 1. *Binary matrix* $\boldsymbol{B}$, whose element $b_{ij} \in \{0,1\}$ $(i = 1,2,\ldots,N,\ j=1,2,\ldots,N)$ indicates whether the labels y_i and y_j match. In other words, 1 denotes x_i and x_j have the same label and belong to the same class. 0 shows they are not in the same class.

Definition 2. *Target neighbors*, kNN with the same label to sample x_i are named as target neighbors of x_i [9].

Definition 3. *Target matrix* $\boldsymbol{T}$, whose element $t_{ij} \in \{0,1\}$ $(i = 1,2,\ldots,N,\ j=1,2,\ldots,N)$ indicates whether x_j is a target neighbor of x_i. $t_{ij}=1$ denotes x_j is a target neighbor of x_i.

The target neighbors can be simply determined by distance between samples. In this paper, the Euclidean distance is used. Noted that a sample is not a target neighbor of itself and then $t_{ii}=0$.

2.1. Loss-based evaluation function

Based on kNN classification rule, which classifies each unlabelled sample by the majority label among its kNN in the training set. Then in order to achieve desirable classification performance, feature subset is selected with the aim that the kNN of any sample always belong to the same class while samples from different classes are separated by large margin in the selected feature space. Loss function is a common technique in machine learning for finding the right balance between small margin-error and large margin [10]. Once the loss function is chosen, the goal of learning algorithm is to minimize it and get large margin. Then feature selection criterion based on loss function is a natural choice. For a sample x_i, the loss function for nearest neighbor classification can be defined as follows.

Definition 4. Let S be a training set, x_i be a sample, the loss function of x_i is

$$L_S(x_i) = \sum_j t_{ij}\|x_i - x_j\|^2 + c\sum_{jp} t_{ij}(1-b_{ip})h_{jp}(x_i)$$
$$h_{jp}(x_i) = [\theta_i + \|x_i - x_j\|^2 - \|x_i - x_p\|^2]_+ \quad (1)$$

where x_p $(p=1,2,\ldots,N)$ and x_j $(j=1,2,\ldots,N)$ are samples in S. In $h_{jp}(x_i)$, $[z]_+ = \max(z,0)$ denotes the hinge loss, $c>0$ is some positive scaling factor (typically set by cross validation). t_{ij} and b_{ip} are the elements in target matrix $\boldsymbol{T}$ and binary matrix $\boldsymbol{B}$, respectively.

It should be noted that the loss function has two competing terms, the first term penalizes large distance between each sample and its target neighbors, not between all similarly labelled samples, while the second term penalizes small distance between each sample and other samples that do not share the same label. In the second term, we especially concern the different labelled samples located in distance from x_i to any of its target neighbors plus a predefined margin θ_i, which is a positive constant and defined as

$$\theta_i = |\|x_i - nearmiss(x_i)\|^2 - \|x_i - nearhit(x_i)\|^2| \quad (2)$$

where $nearhit(x_i)$ and $nearmiss(x_i)$ denote the nearest samples to x_i with the same and different label, respectively, [19]. They are easy to be obtained according to matrix $\boldsymbol{B}$ and $\boldsymbol{T}$, and the definition of θ_i assures at least one sample with different label to x_i (e.g., $nearmiss(x_i)$) will be computed in the loss function.

Note that the chosen feature subset affects the loss of nearest neighbor classification through the influence on distance measure. Then we introduce an evaluation criterion for feature selection based on low-loss nearest neighbor classification which assigns a weight to feature. When selecting a set of features, we can identify them by their weights. Firstly we formulate the loss as a function of the weighted features based on Definition 4.

Definition 5. Let S be a training set, x_i be a sample and w be a weight vector over the feature set, then the loss function of x_i is

$$L_S(w,x_i) = \sum_j t_{ij}\|x_i - x_j\|_w^2 + c\sum_{jp} t_{ij}(1-b_{ip})h_{jp}(w,x_i)$$
$$h_{jp}(w,x_i) = [\theta_i + \|x_i - x_j\|_w^2 - \|x_i - x_p\|_w^2]_+ \quad (3)$$

where $\|z\|_w = \sqrt{\sum_f w_f^2 z_f^2}$, $w_f \in [0,1]$ $(f=1,2,\ldots,n)$. Definition 5 considers the weights over features in the distance calculation.

Now we turn to define the evaluation function. A good generalization can be guaranteed if many samples have low loss and large margin [15]. When a training set is available, we sum the loss over the samples and then:

Definition 6. Given a training set S and weight vector w, the evaluation function is

$$e(w) = \sum_i L_S(w,x_i) \quad (4)$$

Formally, the evaluation function is well defined for any w and we utilizes it in our method.

2.2. Evaluation function analysis

Loss function incorporates the idea of margin. Especially, in the second term of Definition 4, the hinge loss h is incurred by differently labelled samples whose distances to x_i do not exceed the distance from x_i to any of its target neighbors plus the margin θ_i. Therefore the evaluation function supports the feature space in which differently labelled samples maintain a large margin of distance and do not threaten to "invade" each other's neighborhoods. The samples behavior induced by this evaluation function are illustrated in Fig. 1 for a sample x_i with $k=3$ target neighbors. The target neighbors move closer to x_i, while the different labelled samples located in specific domain move farther to x_i. This leads to large margin nearest neighbor classification and achieve good classification performance. The target neighbors are represented by circle and the solid square denotes the differently labelled samples whose distance to x_i does not exceed the distance from x_i to any of its target neighbors plus the predefined margin θ_i represented by green solid lines. Arrows indicate the gradient directions on distance arising from the optimization of evaluation function. In addition, the definition of θ_i contains the idea of the hypothesis-margin of sample x_i for 1-NN classification [10].

Moreover, the two terms in Eq. (3) are analogous to those in the loss function for SVMs [9]. In both loss function, one term penalizes the parameter (such as weight) vector of the maximum margin

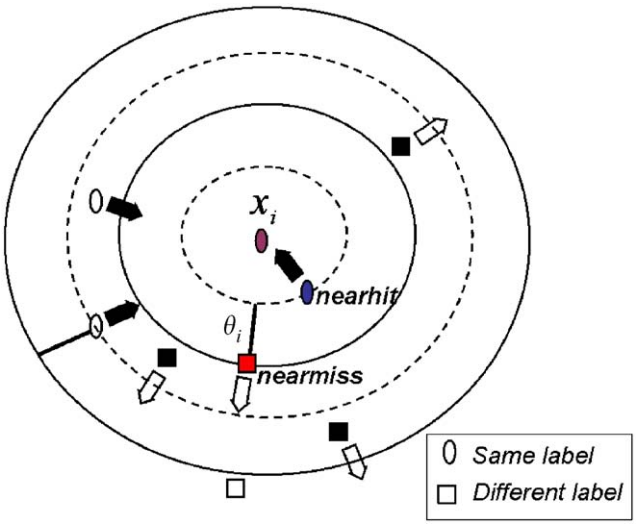

Fig. 1. Illustration for optimization of evaluation function.

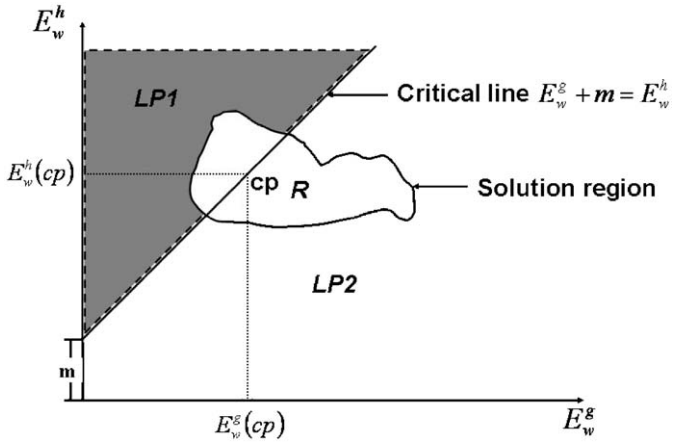

Fig. 2. The plane formed by energy model.

hyperplane, while the other incurs the hinge loss for samples that violate the condition of predefined margin. On the other hand, just as the hinge loss in SVMs is only triggered by samples near the decision boundary, the hinge loss h in the proposed evaluation function is also only triggered by differently labelled samples that invade each other's neighborhoods, i.e., the samples near the decision hyperplane.

Finally, on the one hand, the large margin intuitively yields maximal robust to perturbation (such as noise and outlier). On the other hand, in SVMs, the risk of outliers is normally mitigated by using a soft margin criterion, such as hinge loss is utilized to reduce outlier sensitivity [16]. In our case, the propose evaluation function is also a soft margin criterion because of the use of hinge loss, and then it can efficiently reduce the effect of outliers. At the same time, the number of target neighbors k is assigned as $k > 1$, then it filters noise better for nearest neighbor classification [10].

3. Feature selection algorithm

3.1. Loss-margin based algorithm (Lmba)

In order to find the feature subset that minimizes the evaluation function, many search strategies can be used, such as sequential forward and backward search, plus-l-take-r, sequential floating search, genetic algorithm, branch and bound, etc. [1,11]. However, they assign the feature weight w_f as 1 or 0 to indicate whether the fth feature is selected or not, and these search strategies at least have time complexity $O(N^2n^2)$, where N is the size of training set and n is the number of features. However, we like to let the feature weight w_f take values [0,1] and rank features based on their weights for the reasons described in Section 1. Now the question is raised: does the minimization of the proposed evaluation criterion obtain a weight vector w that leads to the behavior depicted in Fig. 1? We will give answer to this question based on the EBM [12–14].

Let x_i and x_j be a pair of samples, w be the shared parameter vector, and the energy between samples is defined as

$$E(w, x_i, x_j) = \|x_i - x_j\|_w \tag{5}$$

which is used to measure the compatibility between x_i and x_j, and it is a weighted Euclidean distance.

Given a genuine pair samples (x_i, x_j) on which x_j is a target neighbor of x_i, and a heterogeneous pair samples (x_i, x_p) on which x_p is a differently labelled sample whose distance to x_i does not exceed the distance from x_i to any of its target neighbors plus the a margin m. The samples behave in a desirable manner if the following condition holds:

Condition 1. $\exists m > 0$, *such that* $E(w, x_i, x_j) + m < E(w, x_i, x_p)$.

For simplicity, notation $E(w, x_i, x_j)$ is written as E_w^g and $E(w, x_i, x_p)$ as E_w^h for the remainder of the paper.

We will consider a training set consists of one genuine pair (x_i, x_j) with energy E_w^g and one heterogeneous pair (x_i, x_p) with energy E_w^h. Let us define

$$L(E_w^g, E_w^h) = L_g(E_w^g) + L_h(E_w^h) \tag{6}$$

$$L_g = (E_w^g)^2 \tag{7}$$

$$L_h = h[\theta_i + (E_w^g)^2 - (E_w^h)^2]_+ \tag{8}$$

L is the total loss function for the two pairs. L_g and L_h are the partial loss function for a genuine pair and heterogeneous pair, respectively. Then the following proposition is given:

Proposition 1. *Minimizing L with respect to w would lead to finding a w that satisfies Condition* 1.

Proof. As depicted in Fig. 2, there exists two half planes $E_w^g + m < E_w^h$ and $E_w^g + m \geqslant E_w^h$, and they are denoted by LP1 and LP2, respectively. We like to minimize L over E_w^g and E_w^h for all values of w in its domain. Let R be the region inside the plane formed by E_w^g and E_w^h which correspond to all values in the range of w. In the most common setting, R could be lied anywhere in the plane. However, in our case, m is assigned as $\|x_i - nearmiss(x_i)\| - \|x_i - nearhit(x_i)\|$. We surely can find a sample x_t, which is not in the same class with x_i, has larger distance than $nearmiss(x_i)$ to x_i, and then $\|x_i - nearhit(x_i)\|_w + m < \|x_i - x_t\|_w$. In other words, there exists a w for a sample such that Condition 1 is satisfied, and then we can draw a conclusion that a part of R intersects the LP1. Now we have to show that there exists at least one point in the intersection of R and LP1, such that the loss L at this point is less than any points in the intersection of R and LP2.

Let cp be the point on the critical line $E_w^g + m = E_w^h$, for which L is minimum. If the energy of cp are $E_w^g(cp)$ and $E_w^h(cp)$, then $E_w^h(cp) = E_w^g(cp) + m$ and

$$L(E_w^g(cp), E_w^h(cp)) = argmin\{L(E_w^g, E_w^g + m)\} \tag{9}$$

On the one hand, the gradient of L is

$$\frac{\partial L}{\partial E_w^g} = \begin{cases} 2E_w^g & \text{if } (\theta_i + (E_w^g)^2) < (E_w^h)^2 \\ 4E_w^g & \text{otherwise} \end{cases} \tag{10}$$

$$\frac{\partial L}{\partial E_w^h} = \begin{cases} 0 & \text{if } (\theta_i + (E_w^g)^2) < (E_w^h)^2 \\ -2E_w^h & \text{otherwise} \end{cases} \tag{11}$$

and then the negative of gradient of L at all points on the critical line is in the direction which is inside the half plane LP1. On the other hand, it is apparent that L is convex with respect to E_w^g and E_w^h. So we can conclude that

$$L(E_w^g(cp), E_w^h(cp)) \leqslant L(E_w^g, E_w^h) \tag{12}$$

when $E_w^g + m > E_w^h$.

Now consider a point at a distance ζ away from cp, and inside the half plane LP1. The loss of this point can be denoted as

$$L(E_w^g(cp) - \zeta, E_w^h(cp) + \zeta) \tag{13}$$

and it can be described as follows using a first-order Taylor expansion:

$$\begin{aligned} &L(E_w^g(cp) - \zeta, E_w^h(cp) + \zeta) \\ &\quad = L(E_w^g(cp), E_w^h(cp)) - \zeta \frac{\partial L}{\partial E_w^g} + \zeta \frac{\partial L}{\partial E_w^h} + O(\zeta^2) \end{aligned} \tag{14}$$

Based on Eqs. (10) and (11), the second and third terms on the right side of above equation are negative. Then for sufficient small ζ,

$$L(E_w^g(cp) - \zeta, E_w^h(cp) + \zeta) \leqslant L(E_w^g(cp), E_w^h(cp)) \tag{15}$$

Thus there exists a point in the intersection of R and LP1 that the loss of this point is less than any points in the intersection of R and LP2. Then the proof is completed. □

L is derived from Definition 5 and L is equal to Definition 5 when only one genuine pair and one heterogeneous pair is considered. Then there exists a w that satisfies Condition 1, and we have answered the question described above.

The proof above is based on gradient analysis and $e(w)$ is smooth almost everywhere, then the gradient descent is a natural choice to find the weight vector w that minimizes $e(w)$ as defined in Eq. (4). When a training set S is given, the gradient of $e(w)$ is

$$\begin{aligned} \frac{\partial e(w)}{\partial w_f} &= \sum_i \frac{\partial L_S(w, x_i)}{\partial w_f} \\ &= \sum_i \left(2w_f \sum_j t_{ij} \, (x_{if} - x_{jf})^2 + c \sum_{jp} t_{ij} (1 - b_{ip}) \frac{\partial h_{jp}(w, x_i)}{\partial w_f} \right) \end{aligned} \tag{16}$$

and the gradient of hinge loss function $h_{jp}(w, x_i)$ with respect to feature weight w_f is defined as follows:

$$\begin{aligned} &\frac{\partial h_{jp}(w, x_i)}{\partial w_f} = \begin{cases} 0 & \text{if } (\theta_i + \|x_i - x_j\|^2) < \|x_i - x_p\|^2 \\ g(w_f) & \text{otherwise} \end{cases} \\ &g(w_f) = 2w_f((x_{if} - x_{jf})^2 - (x_{if} - x_{pf})^2) \end{aligned} \tag{17}$$

Now, the steps of the proposed feature selection algorithm Lmba are described as follows.

Step 1: Initialize $w = (1, 1, \ldots, 1)$.

Step 2: Construct matrix $\boldsymbol{B}$ and $\boldsymbol{T}$ for training set S.

Step 3: For $i = 1, 2, \ldots, I$.

(a) Pick an instance x_i.
(b) Find the *nearmiss*(x_i) and *nearhit*(x_i) for sample x_i, and get the value of θ_i.
(c) For $f = 1, 2, \ldots, n$ calculate

$$\nabla_f = 2w_f \sum_j t_{ij}(x_{if} - x_{jf})^2 + c \sum_{jp} t_{ij}(1 - b_{ip}) \frac{\partial h_{jp}(w, x_i)}{\partial w_f}$$

(d) $w = w - \beta_i \nabla / \|\nabla\|$, where β_i is a decay factor.

Step 4: Rank features based on the value of w.

In each iteration we update w only with respect to one sample x_i and it is one term in the sum in Eq. (16). On the one hand, the weights of feature increase, then the relative effect of the correction term ∇ decreases. On the other hand, we have proved there exists a weight vector w that can cause the desired behavior based on the EBM and gradient analysis, so the algorithm is typical convergent.

The parameters of the algorithm are k (number of target neighbors), c (scaling factor), I (number of iterations) and $\{\beta_i\}_{i=1}^I$ (step size decay factor). The value of k and c can be tuned by cross validation. I is usually equal to the number of training samples. It always makes sense to use β_i that decay over time to ensure convergence and regulate the convergence rate. However, on our data, convergence was also achieved with $\beta_i = 1$ and the number of iterations equals to the size of training sample.

The computational time of Lmba mainly contains the calculation of $\boldsymbol{B}$, $\boldsymbol{T}$ and w, they have time complexity $O(N^2)$, $O(N^2)$ and $O(N^2kn)$, respectively, where k is the number of target neighbors. In general, k is a small constant number. Then the total time complexity is $2O(N^2) + O(N^2n) \approx O(N^2n)$.

3.2. Comparison to algorithms based on NN rule

Simba [15,17] is also a feature selection algorithm based on NN rule, which was shown to be very efficient for estimating features quality. The evaluation criterion is to directly maximize the hypothesis-margin of 1-NN. The complexity of Simba is $O(N^2n)$. There are two major drawbacks for Simba: first, the nearest neighbors are defined in the original feature space, which may not be true in the weighted feature space; second, it cannot deal with noise and outlier data. However, note that the proposed algorithm Lmba can alleviate first issue by considering all the samples whose distances to x_i less than the distance between x_i to its k target neighbors plus the margin θ_i, not just one *nearmiss*(x_i) and *nearhit*(x_i) as in Simba. In addition, the value of k in Lmba is more than 1 and the hinge loss is used, then it filters noise and outlier better. And the definition of θ_i in Lmba incorporates the idea of hypothesis margin of 1-NN. In a word, Lmba is more robust and powerful than Simba.

Mitra's [18] is to find the feature subset that is highly correlated based on the kNN rule. The time complexity of Mitra's is $O(n^2N)$. Although Mitra's is based on kNN rule, however, it is an unsupervised feature selection method without using the label information in training and it focuses on features instead of samples. It only utilizes the kNN rule to cluster the features to find feature having the most compact subset, i.e. having the largest similarity to its kth nearest neighbor feature, and discard its k neighbors. The process is repeated for the remaining features until all of them are considered. At the same time, k may be changing over iterations and k controls the size of selected feature subset. However, it only can eliminate redundant features. All of these are different from Lmba.

In addition, Relief [19] is another well-known feature selection algorithm based on nearest neighbor rule, and it was shown to be

very efficient for estimating features quality. The algorithm holds a weight vector over all features and updates this vector according to the sample points presented. Relief was extended to deal with multi-class problems, noise and missing data in Refs. [20,21]. The update rule in a single step of Relief is similar to the one performed by Simba. The time complexity is $O(N^2n)$. However, previous work in Ref. [15] have shown Simba is superior to Relief, and Simba can be considered as the improvement version of Relief. So in the following experiments, we only compare the performance among Lmba, Simba and Mitra's and ignore Relief.

4. Experiments

We empirically evaluate our feature selection method Lmba on different data sets. First, experiments are conducted on both real-world and synthetic data sets to check the correctness of the evaluation function and to see whether the proposed method can rank the important features at the top ranking position. Iris is a well-known real-world benchmark data set and popularly used for testing the performance of clustering and classification algorithms, is taken from UCI ML repository [22]. It contains 150 examples, and they are classified into three classes with 50 examples in every class. Each example is characterized by four numerical features. Out of the four features, it is known that No. 3 (petal length) and No. 4 (petal width) features are more important. Two synthetic data sets, S_1 and multi-class, are generated with different numbers of classes and features, and they all have 100 samples. For S_1, the first six features are chosen as important features and these features follow Gaussian distribution. Unimportant features are added which take uniformly random values. For multi-class, the value of data set $X=\{x_1,x_2,\ldots,x_{100}\}$ are random, and the labels $Y=\{y_1,y_2,\ldots,y_{100}\}$ are generated using following Matlab function: $Y=bin2dec(num2str(X(:,\ 1{:}2)>0))$, then first two features are important. The serial number (SN) of important features for these data sets is shown in Table 1 in second column (from left to right).

Other three real-world benchmark data sets taken from UCI ML repository [22] are also used to evaluate the performance of Lmba, Simba and Mitra's.

Multi-features: The data set consists of features of handwritten numerals ("0"–"9") extracted from a collection of Dutch utility maps. There are total 2000 patterns, 649 features and 10 classes.

Pima Indian diabetes (DIAB): The data set contains 768 samples from two classes, where 500 samples are from class 1 and the remaining 268 samples are from class 2. Each sample is represented by eight features. The problem posed is to predict whether a patient would test positive for diabetes according to World Health Organization criteria.

Wisconsin diagnostic breast cancer (WDBC): The data set consists of 357 benign samples and 212 malignant samples, with 30 real-valued features. The task here is to predict diagnosis results (benign or malignant).

In addition, some facial data sets are used to show the gender classification performance of selected feature subsets for different feature selection algorithms. For gender classification, the gallery sets used for training include 500 male samples and 500 female samples, which have the same vector dimension of 1584 gabor filter. The probe sets used for testing include 15 kinds of facial images, which consists of various backgrounds, poses, expressions and occlusions, such as front with blue background (front 1), front with nature background (front 2), down 10°, down 20°, down 30°, smiling, closed eyes, opened mouth, front with glasses, right 10°, right 20°, right 30°, up 10°, up 20° and up 30°. The number of testing face images in these probe sets are 1278, 1066, 820, 819, 816, 805, 815, 805, 813, 814, 815, 805, 819, 816 and 816, respectively, and these probe sets are numbered as 1–15, i.e., the SN of 15 probe sets are $1,2,3,4,\ldots,15$,

Table 1
Description of benchmark and synthesis data sets with ranking results

Data sets	SN of important features	Lmba ranking (descending)	Simba ranking
Iris	3,4	$\{3,4\},2,1$	$\{3,4\},2,1$
Multi-class	1,2	$\{2,1\},9,6,\ldots$	$\{2,1\},9,10,\ldots$
S_1	1,2,3,4,5,6	$\{2,3,6,1,4,5\},8,\ldots$	$\{2,1,3,6,4,5\},9,\ldots$

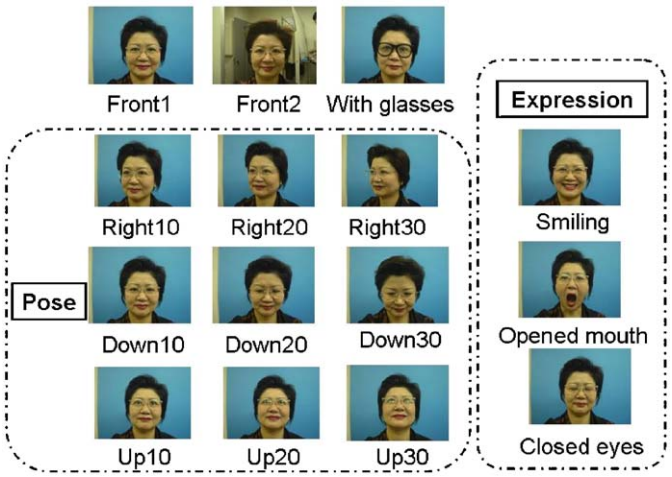

Fig. 3. Examples of facial images in probe sets.

respectively. Examples of facial images from these probes sets are shown in Fig. 3, and these facial sets have been used in [23,24].

4.1. Experimental results on benchmark data sets

Lmba and Simba are used to rank the features, and the results are listed in the third and fourth column of Table 1, respectively (from left to right). From this table, we can see that our method Lmba and Simba are able to rank the relevant features in the top positions. However, Mitra's cannot rank the features, so the results of Mitra's are not shown in Table 1.

For other real-world data sets, we compare the classification performance of selected feature subset instead of listing the concrete selected features. We use the classification accuracy of 1-NN classifier to evaluate the selected feature subsets and fivefold cross-validation is adopted. In Lmba, the parameter c and the number of target neighbors k are set to 1 and 3, respectively, and the same sets are used in other experiments. The results are shown in Figs. 4, 5 and 6, which are corresponding to Multi-features, WDBC and DIAB, respectively.

4.2. Gender classification

In this subsection, we present experimental results of gender classification for different numbers of selected features and various feature selection algorithms, which are Lmba, Simba and Mitra's. The number of selected features is chosen as 127, 254, 508 and 1016. Traditional SVM [25] is adopted, and the parameter C is set to 1. Detailed gender classification rates of different algorithms on 15 probe sets are displayed in Figs. 7–10. The figures are corresponding to different numbers of selected features, i.e. 127, 254, 508 and 1016, respectively. The X-axis is the SN of probe sets and Y-axis is the accuracy rate on these probe sets for gender classification. The average accuracy rates on 15 probe sets for different numbers of selected features are shown in Table 2.

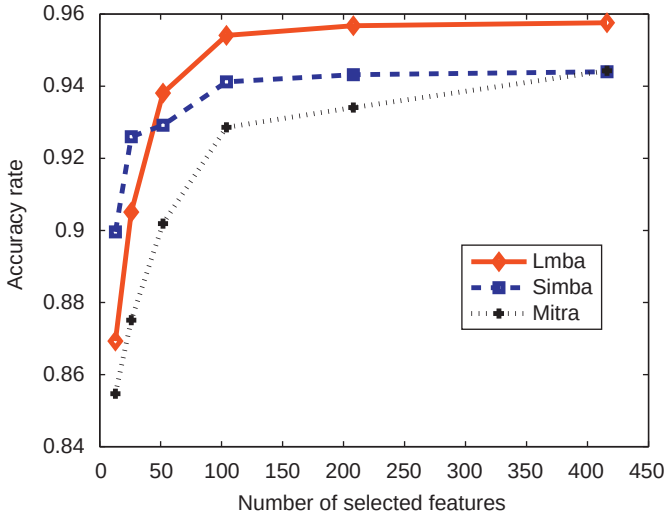

Fig. 4. Experimental results for Multi-features.

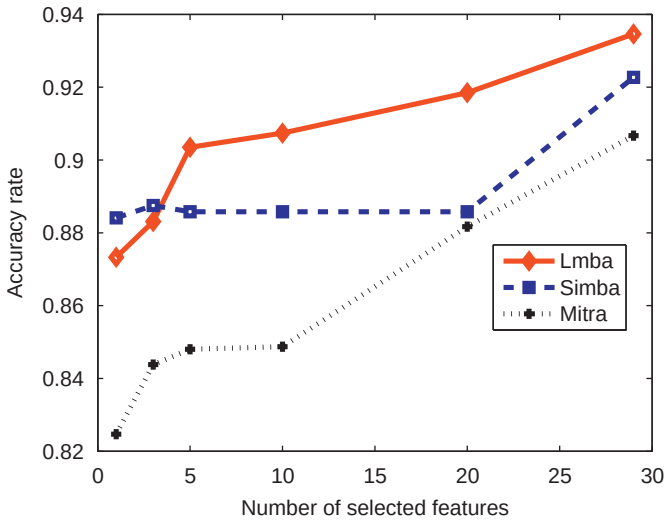

Fig. 5. Experimental results for WDBC.

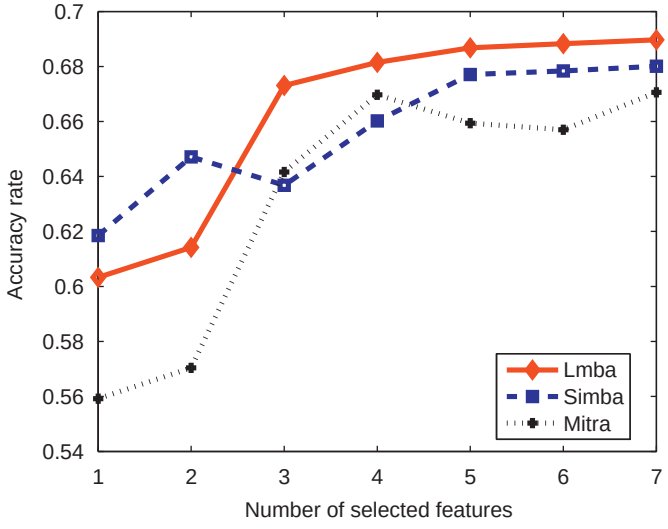

Fig. 6. Experimental results for DIAB.

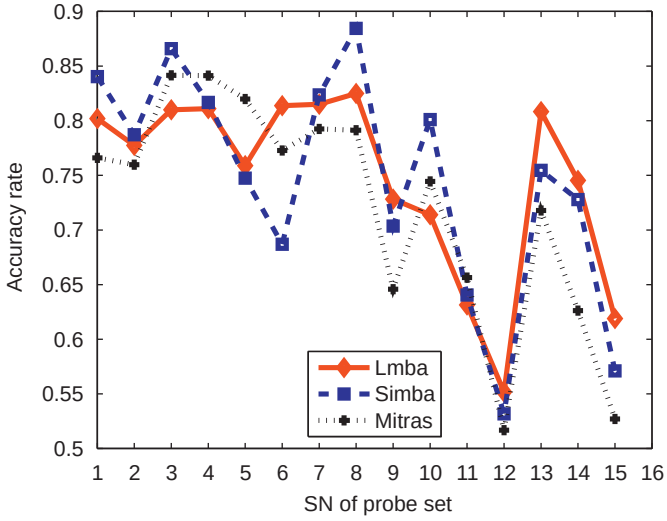

Fig. 7. Experimental results for the number of features 127.

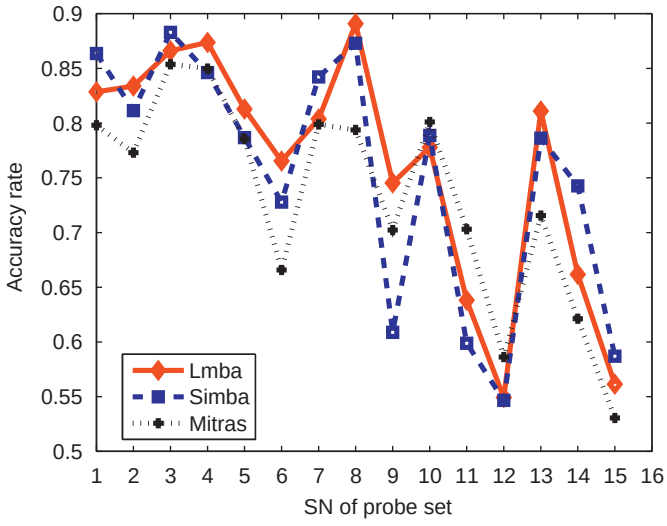

Fig. 8. Experimental results for the number of features 254.

4.3. Observations

From the experiments on benchmark and facial data sets, we can obtain the following two observations.

- The proposed evaluation criterion based on loss-margin of nearest neighbor classification can correctly rank the important features.
- For different data sets, Lmba and Simba often obtain higher performance than Mitra's for different classifiers (e.g. 1-NN and SVM). At the same time, Lmba gets highest performance in most cases. For gender classification, the number of probe sets on which Lmba obtains highest accuracy is more than Simba and gets highest average accuracy rates for different number of selected features.

For the experimental results, Simba, which is based on large margin 1-NN classification, works well for 1-NN classifier and data set without noise. So we conduct experiments to compare Lmba with Simba under these conditions. Of course, if we adopt kNN classifier ($k>1$) and experiments are conducted on data set with many noise, the Lmba will surely get much higher performance than Simba

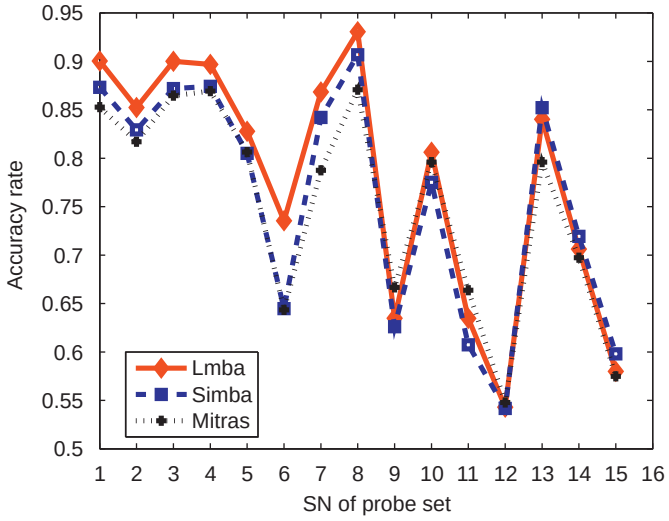

Fig. 9. Experimental results for the number of features 508.

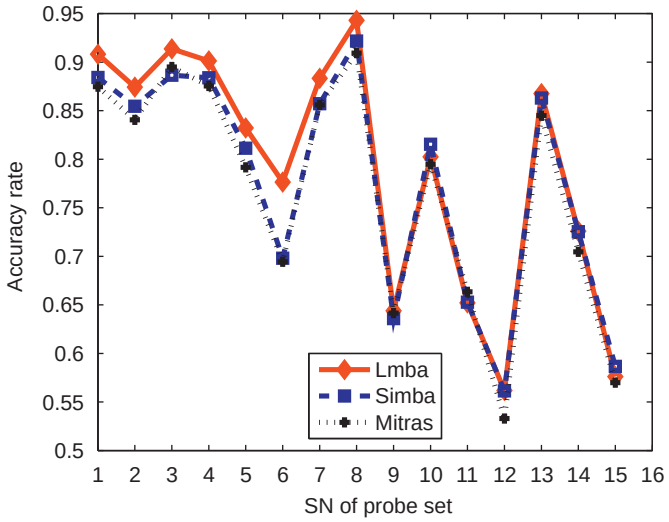

Fig. 10. Experimental results for the number of features 1016.

because *k*NN rule performs better than 1-NN rule as it filters noise better [10] and the proposed evaluation function is based on large-margin of *k*NN classification. It is well-known if an algorithm selects a small set of features with large margin, the bound guarantees it generalizes well [15].

5. Conclusions and discussions

In the paper we used the maximal margin principle together with loss function to derive algorithm for feature selection. An evaluation criterion based on the loss function of nearest neighbor classification is proposed. At the same time, theoretic analysis of the evaluation function based on margin and EBM is presented. We derive feature selection algorithm Lmba by using gradient descent to minimize the evaluation criterion, which indirectly maximizes the margin of nearest neighbor classification. We have shown that Lmba outperforms Simba and Mitra's, which are based on hypothesis margin of 1-NN classification and *k*NN rule clustering, respectively, on benchmark data sets and a gender classification task. Although the criterion is focus on large margin nearest neighbor classification, it is also can get better results for other distance-based classifiers (such as SVM) than Simba and Mitra's. Note that the *k*NN rule usually performs better than 1-NN rule as it filters noise better, then Lmba will be more powerful when applied to data set with noise. Of course, other optimization algorithm for minimizing our loss-margin based criterion can be utilized and more attention should be paid to reduce the effects of outlier and noise data.

Table 2
The average accuracy rate on 15 probe sets for gender classification

No. selected features	Algorithms	Average accuracy rate (%)
127	Lmba	74.73
	Simba	74.55
	Mitra's	72.13
254	Lmba	76.23
	Simba	75.26
	Mitra's	73.19
508	Lmba	77.72
	Simba	75.78
	Mitra's	75.00
1016	Lmba	79.08
	Simba	77.58
	Mitra's	76.59

Acknowledgment

We gratefully thank OMRON Cooperation for supplying facial images and R.G. Bachral for the code of Simba. This work was done in part while the first author was a Postdoctoral fellow at Department of Computer Science and Engineering, Shanghai Jiao Tong University, P.R. China. This research was partially supported by the National Natural Science Foundation of China via the Grants NSFC 60773090, and Shanghai Jiao Tong University and Microsoft Research Asian Joint Laboratory for Intelligent Computing and Intelligent Systems, Natural science fund for colleges and universities in Jiangsu Province via the Grants 08KJB520007, and Scientific Research Foundation of Nanjing University of Posts and Telecommunications via the Grants NY207137.

References

[1] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (2005) 494–502.
[2] R. Kohavi, G. John, Wrapper for feature subset selection, Artif. Intell. 97 (1997) 234–273.
[3] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, J. Mach. Learn. Res. 3 (2003) 1157–1182.
[4] H. Liu, H. Motoda, Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic, Boston, 1998.
[5] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1967) 21–27.
[6] X.D. Wu, V. Kumar, et al., Top 10 algorithms in data ming, Knowl. Inf. Syst. 14 (1) (2008) 1–37.
[7] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (1995) 273–297.
[8] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Statist. 26 (1998) 1651–1686.
[9] K.Q. Weinberger, J. Blitzer, L.K. Saul, Distance metric learning for large margin nearest neighbor classification, in: Proceedings of Advances in Neural Information Processing Systems, vol. 18, Cambridge, MA, 2006.
[10] K. Crammer, R.G. Bachrach, A. Navot, N. Tishby, Margin analysis of the lvq algorithm, in: Proceedings of Advances in Neural Information Processing Systems, La Jolla, CA, 2002.
[11] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, Englewood Cliffs, 1982.
[12] Y. LeCun, F.J. Huang, Loss functions for discriminative training of energy-based models, in: Proceedings of International Workshop on Artificial Intelligence and Statistics, 2005.
[13] Y. LeCun, S. Chopra, R. Hadsell, F.J. Huang, M.A. Ranzato, A tutorial on energy-based learning, in: Bakir et al. (Eds.), Predicting Structured Outputs, MIT Press, 2006.

[14] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, 2005, pp. 539–546.
[15] R.G. Bachrach, A. Navot, N. Tishby, Margin based feature selection-theory and algorithm, in: Proceedings of International Conference on Machine Learning, Banff, Canada, 2004.
[16] L.L. Xu, K. Crammer, D. Schuurmans, Robust support vector machine training via convex outlier ablation, in: National Conference on Artificial Intelligence (AAAI), 2006, pp. 536–546.
[17] I. Guyon, S. Gunn, M. Nikravesh, L. Zadeh, Feature Extraction, Foundations and Applications, Springer, Physica-Verlag, New York, 2006.
[18] P. Mitra, C.A. Murthy, S.K. Pal, Unsupervised feature selection using feature similarity, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 301–312.
[19] K. Kira, L. Rendell, A practical approach to feature selection, in: Proceedings of International Conference on Machine Learning, 1992, pp. 249–256.
[20] I. Kononerko, Estimating attributes: analysis and extension of RELIEF, in: Proceedings of European Conference on Machine Learning, 1994, pp. 171–182.
[21] M.K. Sikonja, L. Kononenko, An adaptation of relief for attribute estimation in regression, in: Proceedings of International Conference on Machine Learning, 1997, pp. 296–304.
[22] C.J. Merz, P.M. Murphy, UCI repository of machine learning database ⟨http://www.ics.uci.edu/mlearn/MLRepository.html⟩, 1996.
[23] H.C. Lian, B.L. Lu, Gender recognition using a min-max modular SVM, Lecture Notes in Computer Science, vol. 3611, Springer, Berlin, 2005, pp. 433–436.
[24] Y. Li, B.L. Lu, Feature selection for identifying critical variables of principal components based on k-nearest neighbor rule, Lecture Notes in Computer Science, vol. 4781, Springer, Berlin, 2007, pp. 196–207.
[25] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines ⟨http://www.csie.ntu.edu.tw/∼cjlin/papers/libsvm.ps.gz⟩, 2002.

About the Author—YUN LI received Dr.Eng. degree in Computer Software and Theory from ChongQingUniversity, China in 2005, and from July 2005 to October 2007, he was a Post-doctoral fellow at Department of Computer Science and Engineering, Shanghai Jiao Tong University (SJTU), China. Now he is the associate professor at the College of Computer, Nanjing University of Posts and Telecommunications. His research interests are in the area of pattern recognition and data mining.

About the Author—BAO-LIANG LU received Dr.Eng. degree in electrical engineering from Kyoto University, Kyoto, Japan, in 1994. From April 1994 to March 1999, he was a Frontier Researcher at the Bio-Mimetic Control Research Center, the Institute of Physical and Chemical Research (RIKEN), Nagoya, Japan. From April 1999 to August 2002, he was a Research Scientist at the RIKEN Brain Science Institute, Wako, Japan. Since August 2002, he has been a full professor at the Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He has been an adjunct professor of the Laboratory for Computational Biology, Shanghai Center for Systems Biomedicine since 2005. His research interest includes brain-like computing, neural networks, machine learning, pattern recognition, computer vision, brain-computer interface, natural language processing, and computational biology.