Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 27 (2006) 1685–1691
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Multi-class feature selection for texture classification

Xue-wen Chen [a,*], Xiangyan Zeng [b], Deborah van Alphen [b]

[a] Information and Telecommunication Technology Center, Department of Electrical Engineering and Computer Science, The University of Kansas, Lawrence, KS 66045, United States
[b] Department of Electrical and Computer Engineering, California State University, Northridge, CA 91330, United States

Received 18 June 2005; received in revised form 22 February 2006
Available online 27 June 2006

Communicated by M.A.T. Figueiredo

Abstract

In this paper, a multi-class feature selection scheme based on recursive feature elimination (RFE) is proposed for texture classifications. The feature selection scheme is performed in the context of one-against-all least squares support vector machine classifiers (LS-SVM). The margin difference between binary classifiers with and without an associated feature is used to characterize the discriminating power of features for the binary classification. A new criterion of min–max is used to mix the ranked lists of binary classifiers for multi-class feature selection. When compared to the traditional multi-class feature selection methods, the proposed method produces better classification accuracy with fewer features, especially in the case of small training sets.

Keywords: Multi-class feature selection; Texture classification; Least squares support vector machine; Recursive feature elimination; Min–max value

1. Introduction

Texture analysis plays an important role in many computer vision systems. As the crucial steps in texture analysis, feature extraction and selection are receiving more attention (Mao and Jain, 1992; Unser, 1995; Jain and Farroknia, 1991; Zeng et al., 2004; Randen and Hakon Husoy, 1999). Among various feature extraction methods, filter bank methods, such as Gabor filters and wavelet transforms, are the most commonly used. Filter bank methods aim to enhance edges and lines of different orientations and scales in order to obtain different feature components. However, the design of suitable filter banks is not a trivial problem. In recent years, independent component analysis (ICA) has been applied to feature extraction of natural image data (Bell and Sejnowski, 1997; Olshausen and Field, 1996). The obtained ICA filters exhibit Gabor-like structures and provide an orthogonal basis for image coding. ICA filters have been used in image denoising as an adaptive option of wavelet basis (Hyvarinene et al., 1998). In this paper, an ICA filter bank is used to extract texture features and performs much better than the Gabor filters.

The large number of extracted features leads to expensive computation in classification. Additionally, noisy or redundant features may degrade the classification performance. Thus, it is necessary and critical to perform feature selection to identify a subset of features that are capable of characterizing the texture images. Feature selection has been explored in (Grigorescu et al., 2002), where the optimization is based on the intrinsic properties of data and is independent of any specific classifiers (this type of feature selection methods is called filter methods). For instance, the Fisher criterion utilizes the mean of data within classes and the variance of data between classes to select features. In the case of small training sets, these methods tend to be less effective. Alternatively, wrapper methods and embedded methods which involve learning processes in the feature selection can achieve higher accuracy (Kohavi and

* Corresponding author. Tel.: +1 785 864 8825.
E-mail address: xwchen@ku.edu (X.-w. Chen).

0167-8655/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2006.03.013

John, 1997; Guyon and Elisseeff, 2004; Mao, 2004). Wrappers take the learning machine as a black box and evaluate features by classification performance. They try to search for the optimal subset in combinatorial feature space, which leads to intensive computation. Embedded methods perform feature selection in the process of training and reach a solution faster by avoiding retraining the learning machine when every feature is selected. For instance, the recursive feature elimination method (RFE) uses the change in objective functions when a feature is removed as a ranking criterion. With a backward elimination strategy, the features that contribute least to the classification are removed iteratively. The RFE method is usually specific to the given learning machine. In this paper, we will adopt the RFE method to select texture features for multi-class classification.

Texture feature selection is typically a multi-class problem. For multi-class feature selection problems, embedded methods either consider one single criterion for all the classes or decompose multi-class into several two-class problems. It has been pointed out that in the case of uneven distribution across classes, using one single criterion for all the classes may over-represent easily separable classes (Forman, 2003). Alternatively, mixing the results of several binary classifiers may yield better performance. To address this issue, Sindhwani et al. (2004) use the summation of the margin differences of all the binary classifiers as a feature selection criterion for the support vector machines (SVMs) and multi-layer perceptrons, and Weston et al. (2003) use the summation criterion for the multi-class case in their zero-norm learning algorithm.

In this paper, we propose a new method to mix ranked features of several binary classifiers. We use the maximum value of the margin differences of binary classifiers to rank the features and omit ones with minimum values. Compared with the summation criterion, the maximum value is robust to oscillation. This is especially important for the cases with small training samples. Various classifiers have been used in texture classification such as Bayesian classifiers, nearest neighbor classifiers, neural networks, and support vector machines (Manian and Vasquez, 1998; Chitre and Dhawan, 1999; Laine and Fan, 1993; Li et al., 2003). They all may be integrated into embedded feature selection methods. Among these classifiers, SVMs are considered to have better performance for small training sample problems. They aim at minimizing a bound on the generalization error instead of minimizing the training error as do other supervised learning methods (Li et al., 2003; Burges Christopher, 1998). The RFE feature selection based on SVM has been applied to gene selection and was observed to be robust to data overfitting (Guyon et al., 2002). In this paper, the proposed method is performed in the context of least square version of SVM (LS-SVM) (Suykens and Vandewalle, 1999), which is computationally efficient for feature selection where training a large number of classifiers is needed.

The rest of the paper is organized as follows. Sections 2 and 3 briefly introduce the ICA texture features and the LS-SVM. The multi-class RFE algorithm is described in Section 4. The texture classification experiments and concluding remarks are given in Sections 5 and 6, respectively.

2. ICA filter banks

In filter bank methods, a texture image $\boldsymbol{I}(x,y)$ of size $M \times N$ is convolved with a bank of filters g_i

$$\boldsymbol{G}_i(x,y) = \boldsymbol{I}(x,y) \otimes g_i. \tag{1}$$

The energy distributions of the filtered images defined as

$$f_i = \sum_{y=1}^{N} \sum_{x=1}^{M} \boldsymbol{G}_i^2(x,y) \tag{2}$$

are used to represent the texture features. A number of filter banks have been used to extract texture features, including Laws filter masks, Gabor filter banks, wavelet transforms, and discrete cosine transforms.

As an adaptive option of Gabor filters and wavelet basis, the basis images obtained from the independent component analysis (ICA) of natural image patches have been used in image coding and denoising. In this paper, we use the ICA filter bank for extracting the texture features.

To obtain the filter bank, we apply ICA to train 8000 8×8 nature image patches. Each image patch is reshaped row-by-row into column $\boldsymbol{z} = (z_1, z_2, \ldots, z_{64})$. ICA is used to find a matrix $\boldsymbol{W}$, such that the elements of the resulting vector

$$\boldsymbol{x} = \boldsymbol{W}\boldsymbol{z} \tag{3}$$

are statistically as independent as possible over the 8,000 image patches. Each row of $\boldsymbol{W}$ is reshaped into a two-dimensional filter and 64 filters are obtained as in Fig. 1. Several ICA algorithms have been proposed. We use the FastICA algorithm proposed by Hyvarinen and Oja (1997). Compared with other adaptive algorithms, it quickly converges and is not affected by a learning rate.

3. LS-SVM

Given a training set of N data points $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_i \in R^d$ is a feature vector and $y_i \in \{\pm 1\}$ is the corresponding target, the data points are mapped into a high dimensional Hilbert space using a nonlinear function $\varphi(\cdot)$. In addition, the dot product in that high dimensional space is equivalent to a kernel function in the input space, i.e., $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j)$. The LS-SVM (Suykens and Vandewalle, 1999) classifier is constructed by minimizing

$$\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \frac{1}{2}C\sum_i e_i^2 \tag{4}$$

the subject to the equality constraints

$$y_i - (\boldsymbol{w} \cdot \varphi(\boldsymbol{x}_i) - b) = e_i,$$

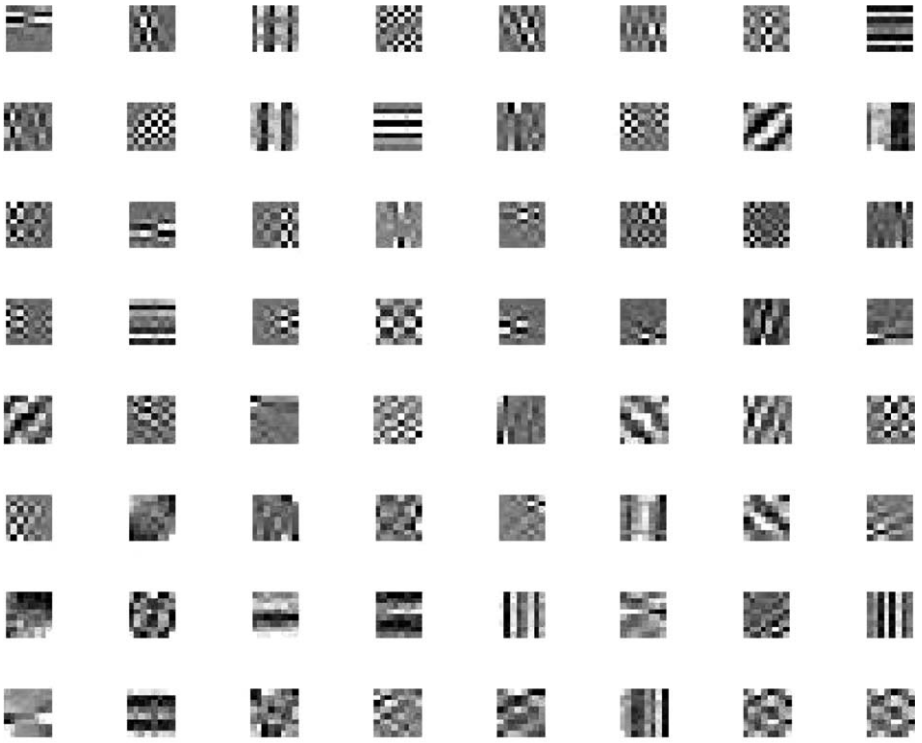

Fig. 1. Sixty-four ICA filters obtained by training an ensemble of 8×8 natural image patches.

where $C > 0$ is a regularization factor, b is a bias term, and e_i is the difference between the desired output and the actual output. The Lagrangian for problem (4) is

$$\Re(\boldsymbol{w}, e_i; \alpha_i) = \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \frac{1}{2}C\sum_i e_i^2 + \sum_i \alpha_i[y_i - \boldsymbol{w}\cdot\varphi(\boldsymbol{x}_i) + b - e_i], \tag{5}$$

where α_i are Lagrangian multipliers. The Karush–Kuhn–Tucker (KKT) conditions for optimality are

$$\begin{cases} \dfrac{\partial \Re}{\partial w} = 0, & \boldsymbol{w} = \sum_i \alpha_i \varphi(\boldsymbol{x}_i), \\ \dfrac{\partial \Re}{\partial e_i} = 0, & \alpha_i = Ce_i, \\ \dfrac{\partial \Re}{\partial \alpha_i} = 0, & y_i - \boldsymbol{w}\cdot\varphi(\boldsymbol{x}_i) - e_i = 0, \end{cases} \tag{6}$$

that constitute a linear system

$$\begin{bmatrix} \boldsymbol{Q} & 1_n \\ 1_N^{\mathrm{T}} & 0 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix}, \tag{7}$$

where $\boldsymbol{Q}_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sigma_{ij}/C$, and $\sigma_{ij} = 1$ if $i = j$ and 0 otherwise. Parameters $\boldsymbol{\alpha}$ and b can be obtained using the conjugate gradient method. LS-SVM avoids solving the quadratic programming problem and simplifies the training of a large number of classifiers in feature selection.

4. Multi-class feature selection

In this section, we present the multi-class feature selection methods for texture classification. The RFE method for binary LS-SVMs is presented in Section 4.1. In Section 4.2, we propose a novel criterion to rank the features for the multi-class classification, which is decomposed into several binary classifiers.

4.1. Recursive feature elimination (RFE)

The RFE iteratively removes the features with least influence on the classification decision and then retrains the classifier. Since the classification ability of LS-SVM depends on the classifier margin, the margin difference between feature set with and without a feature can be formulated as a ranking criterion of the feature importance

$$DW^{-m} = \sum_{i,j} \alpha_i \alpha_j \Big(K(\boldsymbol{x}_i, \boldsymbol{x}_j) - K(\boldsymbol{x}_i^{-m}, \boldsymbol{x}_j^{-m}) \Big), \tag{8}$$

where $\boldsymbol{x}_i^{-m}$, $\boldsymbol{x}_j^{-m}$ are the vectors in which the mth feature has been removed.

While various kernels can be used in SVM design, such as polynomial, RBF, and linear kernels, we consider Gaussian kernels in this study. Note that the selection of kernels is more of an art than science; currently, there is no existing systematic method for kernel selection. Generally, for small samples with high dimensionality, linear kernels may be more appropriate as samples are typically linear separable in high dimensional space. Nonlinear kernels may provide better performance than linear kernels for moderate or large size of samples.

For the nonlinear LS-SVM which uses the Gaussian kernel,

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{-1}{2\sigma^2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right) \tag{9}$$

the margin difference can be efficiently computed by

$$DW^{-m} = \sum_{i,j} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \cdot \left(1 - 1/K(\boldsymbol{x}_i^m, \boldsymbol{x}_j^m)\right), \tag{10}$$

where $\boldsymbol{x}_i^m$, $\boldsymbol{x}_j^m$ are the mth components of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. RFE can be simply implemented by the following iterative steps:

1. train the classifier,
2. compute the ranking criterion for all the features,
3. remove the features with smallest ranking values.

For the sake of computational cost, several features are usually removed at a time.

4.2. Multi-class RFE

To approach the feature selection of multi-class textures, we propose a method of extending the binary RFE. An important strategy to deal with multi-class problems is to decompose the problem into several two-class problems. In this case, the feature selection of binary classifiers needs to be combined. A common way is to use the criterion of $\sum_{k=1}^{C}|DW_k^{-m}|$ (Olshausen and Field, 1996; Hyvarinene et al., 1998; Grigorescu et al., 2002), where DW_k^{-m} is the margin difference of binary classifier k caused by the removal of feature m. The idea is then to remove feature r iteratively selected by

$$r = \arg\min_m \sum_{k=1}^{C} |DW_k^{-m}|. \tag{11}$$

In the above methods, the contribution of a feature is evaluated by the summation of the margin difference of all the binary classifiers. This objective is not necessarily optimal with respect to discrimination, however. In a multi-class classifier which combines several binary classifiers, a new data point $\boldsymbol{x}$ is classified as belonging to the class

$$c = \arg\max_k(\boldsymbol{w}_k\boldsymbol{x} + b_k). \tag{12}$$

For the purpose of discrimination, the contribution of a feature to a multi-class problem is bound by the maximum value instead of the summation of the margin differences of the binary classifiers.

In this paper, we propose a new criterion to select feature r^* such that

$$r^* = \arg\min_m\{\max\{DW_k^{-m},\ k = 1, ,2, \cdots C\}\}. \tag{13}$$

Hence, the feature that has the min–max value of margin difference is omitted. The feature selection algorithm involving k class textures is given below, where *Step* is the number of features removed at a time.

F contains all the features;
Repeat until the number of the remaining features is equal to a predefined number
  For $k = 1 : C$;
    Train LS-SVM_k and obtain $\boldsymbol{\alpha}^k$ and b^k;
  End-for;
  For $j = 1 : Step$;
    Remove feature $m^* = \arg\min_m\{\max_k\{DW_k^{-m}\}\}$ from $\boldsymbol{F}$;
  End;
End-repeat;

5. Experimental results

We have carried out the experiments using a data set of 30 textures (Fig. 2) selected from the Brodatz Album (Brodatz, 1966). Each texture image is 640×640 with 256 grayscales. Each image is divided into 400 32×32 non-overlapping segments. A small fraction (1.25%, 2.5%, and 3.75%) of the 400 images is used in training the LS-SVM, and the rest are used for testing. To intensify the reliability of the experimental results, we use 10 random partitions of training and test data over all 30 textures. The classification accuracy of test data, averaged over the 10 data sets, will be used to evaluate the results.

The one-against-all strategy is utilized to combine 30 binary classifiers to implement the 30-class texture classification. In each binary classifier, one texture is assigned as the positive class and the others as the negative class. Since in each binary classifier the number of samples are unbalanced, we introduce different regularization parameters

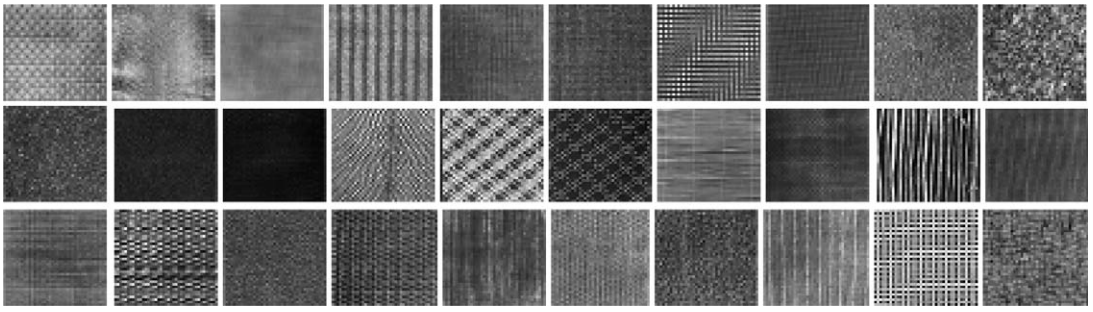

Fig. 2. The 30 Brodatz textures used in the experiment.

Table 1
Average classification accuracy of 64 Gabor filters and ICA filters with different number of training samples

Proportion of training samples (%)	ICA filters	Gabor filters
1.25	88.65	82.56
2.50	91.15	89.16
3.75	94.85	91.96

C_1 and C_2 for the positive and the negative class. The LS-SVM algorithm is modified, $\boldsymbol{Q}_{ii} = K(\boldsymbol{x}_i, \boldsymbol{x}_i) + 1/C_1$ if $\boldsymbol{x}_i$ are samples in the positive class. Otherwise, the algorithm is $\boldsymbol{Q}_{ii} = K(\boldsymbol{x}_i, \boldsymbol{x}_i) + 1/C_2$. Nonlinear LS-SVM with Gaussian kernel is used as the binary classifier. The leave-one-out cross validation is carried out to determine the optimal parameters σ^2, C_1, and C_2 for the initial LS-SVM with the full feature set.

5.1. ICA filters versus Gabor filters

We first compare the classification performance of Gabor filters and the ICA filters shown in Fig. 1. We use the following family of Gabor functions:

$$g_1(x_1, y_1, \theta, \sigma) = \exp\left(-\frac{x_1^2 + \gamma^2 y_1^2}{2\sigma^2}\right)\cos\left(\frac{2\pi x_1}{\lambda}\right), \tag{14}$$

where

$$x_1 = x\cos\theta + y\sin\theta, \quad y_1 = -x\sin\theta + y\cos\theta,$$
$$\lambda = 2\sigma \quad \text{and} \quad \gamma = 0.5.$$

Each bank comprises 64 Gabor filters that use 8 spatial frequencies $\sigma = 20 + 8k$ and 8 different orientations $\theta = k(\pi/8)$, $k = 0, \ldots, 7$. The result is summarized in Table 1. It is clear that the ICA filters outperform the Gabor filters in all the three cases. The advantage of the ICA filters is especially obvious when the proportion of training samples is 1.25%.

5.2. ICA feature selection for multi-class texture classification

In this section, we compare the proposed method (RFE_max) with the conventional RFE method (RFE_sum) and the Fisher Criterion (FC). Starting from the initial LS-SVM with all 64 features, the feature selection methods iteratively omit features with minimum criterion values.

In the FC method, we rank the features by

$$FJ(i) = \frac{\sum_{j=1}^{C}\sum_{k=1}^{C}(\mu_{ij} - \mu_{ik})^2}{\sum_{j=1}^{C}\sigma_{ij}}, \tag{15}$$

where μ_{ij}, σ_{ij} are the mean value and variance of the ith feature in the jth class. The multi-class feature selection scheme described in Section 4.2 is used for the RFE methods, while the maximum value is used in RFE_max and the summation is used in RFE_sum as the selection criterion. We retrain the LS-SVM after four features are removed. Although the RFE methods need to retrain the LS-SVM, the computation time is reasonable due to the small training set.

The average classification accuracy of the test data and the numbers of features are shown in Table 2. It is noted that the performance of the FC method dramatically degrades with the removal of features. Apparently, it is difficult to select a few features using the FC method within small training sets. The FC method achieves the best performance when the number of features is larger than 56. The reason is that the correlated features contribute little to the LS-SVM classification. The FC method is effective in removing these correlated features. The RFE methods use information about a single feature, which has no effect on correlated features.

To fill up the deficiency of RFE, we remove the first 4 features using the FC method and select the remaining features using the RFE methods. Comparisons of the RFE methods and the corresponding hybrid methods are shown in Fig. 3. It is observed that the performance of RFE methods is improved by the modification. In general, RFE_sum benefits more than RFE_max from combining with the FC method. When the proportion of training samples is 1.25% and 2.5%, the hybrid method (FC+RFE_sum) keeps the superiority to the RFE_sum under all the phases from 60 until 12 features. In the case of 3.75%, the hybrid method (FC+RFE_sum) loses the superiority when the number of features is less than 12, which is observed in all three proportion cases for the RFE_max method. The smaller difference between the RFE method and the corresponding hybrid method infers a higher ability of the RFE method. Looking at the problem from another viewpoint, one can

Table 2
Average classification accuracy of three proportions of training samples versus the number of features

Number of features	1.25%			2.50%			3.75%		
	RFE_max	RFE_sum	FC	RFE_max	RFE_sum	FC	RFE_max	RFE_sum	FC
64	88.65	88.65	88.65	91.15	91.15	91.15	94.85	94.85	94.85
56	88.80	88.75	**89.14**	91.37	91.33	**91.64**	95.05	95.00	**95.15**
48	**89.02**	88.94	87.86	**91.68**	91.60	90.59	**95.29**	95.21	94.58
40	**89.32**	89.19	85.90	**91.90**	91.79	89.25	**95.56**	95.42	93.78
32	**89.67**	89.05	84.03	**92.22**	91.81	87.74	**95.72**	95.41	92.01
24	**89.60**	88.80	81.30	**92.52**	91.67	85.60	**95.68**	95.37	90.45
16	**89.37**	87.68	77.79	**92.00**	90.73	81.67	**95.50**	94.66	86.14
8	**84.35**	82.25	62.86	**87.65**	87.05	64.75	**92.12**	91.33	73.12

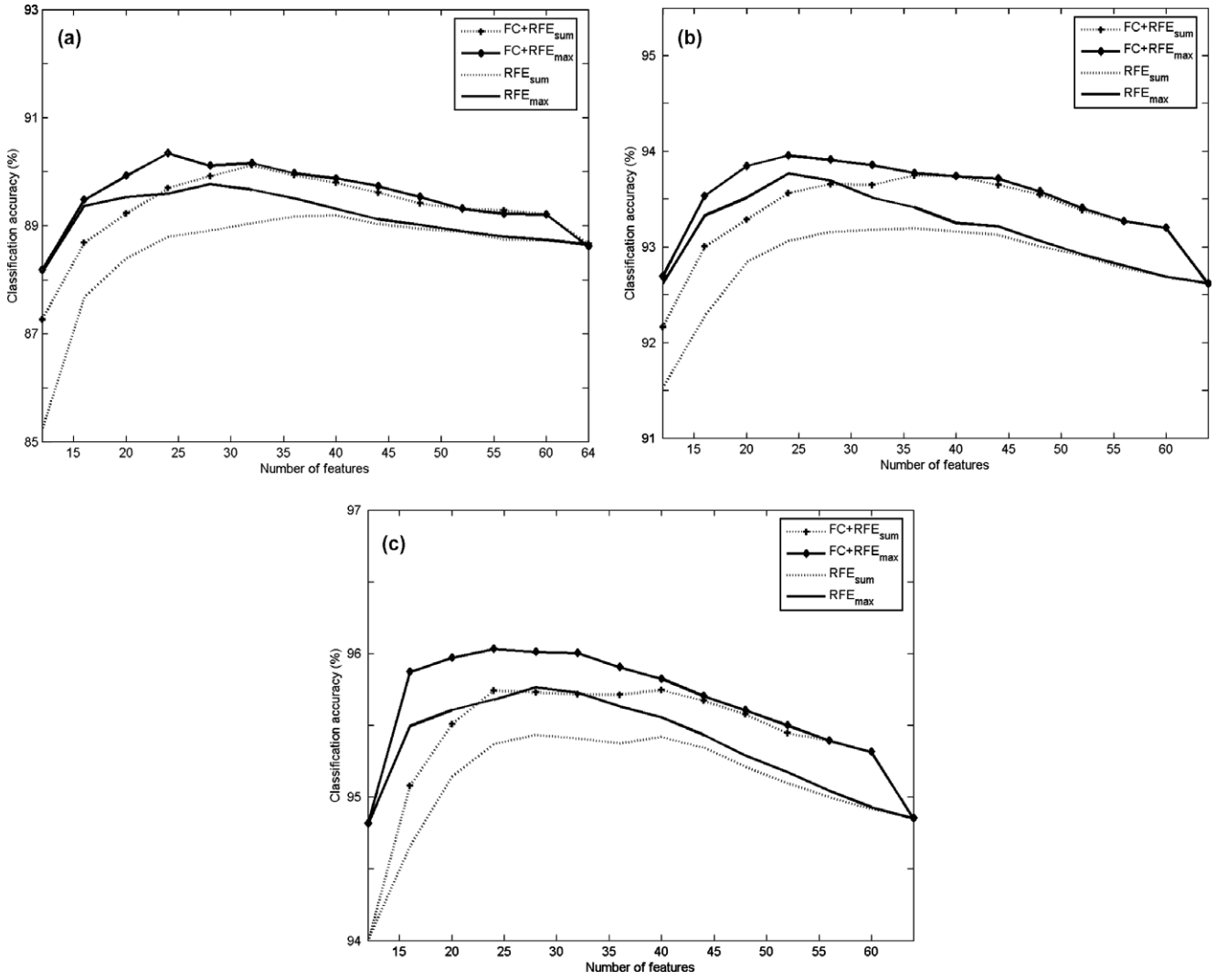

Fig. 3. Classification accuracy with the training rates of (a) 1.25%, (b) 2.5%, (c) 3.75%.

say that RFE_max is more robust than RFE_sum within small training sets.

6. Conclusions

In this paper, we present a feature selection scheme for multi-class texture classification using LS-SVM. Firstly, we demonstrated that the ICA filters used to extract the texture features possess higher accuracies of the initial classification. Secondly, a new criterion is proposed to mix the ranked lists of binary classifiers. The proposed method was compared with the commonly used summation criterion and Fisher Criterion. Simulation experiments have been carried out on 30-class Brodatz textures, which demonstrate that the proposed method outperforms the other methods.

Acknowledgements

This material is based upon work supported by the US Army Research Laboratory and the US Army Research Office under contract number DAAD19-03-1-0123.

References

Bell, A.J., Sejnowski, T.J., 1997. The independent components nature scenes are edge filters. Vision Res. 37, 3327–3338.

Brodatz, P., 1966. Textures: A Photographic Album for Artists and Designers. Dover, New York.

Burges Christopher, J.C., 1998. A tutorial on support vector machines for patter recognition. Data Min. Knowl. Disc. 2, 121–167.

Chitre, Y., Dhawan, A.P., 1999. M-band wavelet discrimination of natural textures. Pattern Recognition 32 (5), 773–789.

Forman, G., 2003. An extension empirical study of feature selection metrics for text classification. J. Machine Learn. Res. 3, 1289–1306.

Grigorescu, S.E., Petkov, N., Kruizinga, P., 2002. Comparison of texture features based on Gabor filters. IEEE Trans. Image Process. 11, 1160–1167.

Guyon, I., Elisseeff, A., 2004. An introduction to variable and feature selection. J. Machine Learn. Res. 3, 1157–1182.

Guyon, I., Weston, J., Barnhill, S., Vapnik, V., 2002. Gene selection for cancer classification using support vector machines. Machine Learn. 46, 389–422.

Hyvarinen, A., Oja, E., 1997. A fast fixed-point algorithm for independent component analysis. Neural Comput. 9, 1483–1492.

Hyvarinene, A., Hoyer, P., Oja, E., 1998. Sparse code shrinkage for image denoising. Proc. IEEE Int. Conf. Neural Networks, 859–864.

Jain, A.K., Farroknia, F., 1991. Unsupervised texture segmentation using Gabor filters. Pattern Recognition 24, 1167–1186.

Kohavi, R., John, G.H., 1997. Wrappers for feature subset selection. Artif. Intell. 97 (1–2), 273–324.

Laine, A., Fan, J., 1993. Texture classification by wavelet packet signatures. IEEE Trans. Pattern Anal. Machine Intell. 15 (11), 1186–1191.

Li, S., Kwork, J.T., Zhu, H., Wang, Y., 2003. Texture classification using the support vector machines. Pattern Recognition 36, 2883–2893.

Manian, V., Vasquez, R., 1998. Scaled and rotated texture classification using a class of basis function. Pattern Recognition 31 (12), 1937–1948.

Mao, K.Z., 2004. Feature subset selection for support vector machines through discriminative function pruning analysis. IEEE Trans. Systems Man Cyber. – Part B: Cyber. 34 (1), 60–67.

Mao, J., Jain, A.K., 1992. Texture classification and segmentation using multiresolution simultaneous autoregressive models. Pattern Recognition 25, 173–188.

Olshausen, B.A., Field, D.J., 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609.

Randen, T., Hakon Husoy, J., 1999. Filtering for texture classification: a comparative study. IEEE Trans. Pattern Anal. Machine Intel. 21, 291–310.

Sindhwani, V., Rakshit, S., Deodhare, D., Erdogmus, D., Principe, J., Niyogi, P., 2004. Feature selection in MLPs and SVMs based on maximum output information. IEEE. Trans. Neural Networks 15 (4).

Suykens, A.K., Vandewalle, J., 1999. Least squares support vector machine classifiers. Neural Process. Lett. 9, 293–300.

Unser, M., 1995. Texture classification and segmentation using wavelet frames. IEEE. Trans. Image Process. 4, 1549–1560.

Weston, J., Elisseeff, A., Scholkopf, B., Tipping, M., 2003. Use of the zero-norm with linear models and kernel methods. J. Machine Learn. Res. 3, 1439–1461.

Zeng, X.-Y., Chen, Y.-W., Nakao, Z., Lu, H., 2004. Texture representation based on pattern map. Signal Process. 84, 589–599.