Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 26 (2008) 578–591
www.elsevier.com/locate/imavis

Multiclass cell detection in bright field images of cell mixtures with ECOC probability estimation

Xi Long [a], W. Louis Cleveland [b,*], Y. Lawrence Yao [a]

[a] Mechanical Engineering Department, Columbia University, 220 Mudd., MC4703, New York, NY 10027, USA
[b] Department of Medicine at St. Luke's Roosevelt Hospital Center and Columbia University, New York, NY 10019, USA

Received 12 September 2005; received in revised form 5 March 2007; accepted 11 July 2007

Abstract

To achieve high throughput with robotic systems based on optical microscopy, it is necessary to replace the human observer with computer vision algorithms that can identify and localize individual cells as well as carry out additional studies on these cells in relation to biochemical parameters. The latter task is best accomplished with the use of fluorescent probes. Since the number of fluorescence channels is limited, it is highly desirable to accomplish the cell identification and localization task with transmitted light microscopy. In previous work, we developed algorithms for automatic detection of unstained cells of a single type in bright field images [X. Long, W.L. Cleveland, Y.L. Yao, A new preprocessing approach for cell recognition, IEEE Transactions on Information Technology in Biomedicine 9 (3) (2005) 407–412; X. Long, W.L. Cleveland, Y.L. Yao, Automatic detection of unstained viable cells in bright field images using a support vector machine with an improved training procedure, Computers in Biology and Medicine 36 (2006) 339–362]. Here we extend this technology to facilitate identification and localization of multiple cell types. We formulate the detection of multiple cell types in mixtures as a supervised, multiclass pattern recognition problem and solve it by extension of the Error Correcting Output Coding (ECOC) method to enable probability estimation. The use of probability estimation provides both cell type identification as well as cell localization relative to pixel coordinates. Our approach has been systematically studied under different overlap conditions and outperforms several commonly used methods, primarily due to the reduction of inconsistent labeling by introducing redundancy. Its speed and accuracy are sufficient for use in some practical systems.

Keywords: Cell detection; Error Correcting Output Coding (ECOC); Multiclass classification; Support vector machines

1. Introduction

In high throughput robotic systems that use optical microscopy, it is essential to replace the human observer with automatic cell recognition algorithms. A first step towards this goal is to develop algorithms that can distinguish between "Cell" and "Non-cell" objects. This has recently become possible for bright field images of unstained cells in cultures using statistical learning techniques [1]. Even the distinction between viable and non-viable cells in these images can be done with sufficient accuracy for practical applications [2]. To proceed further towards the goal of fully automated microscopy, it is of critical importance to develop algorithms that can sort cell objects into subtypes.

Recognition of cell subtypes is a multiclass classification problem. Although binary classification has been well developed, the problem of multiclass classification is still an ongoing research issue and is not straightforward [3,4]. Some binary classification methods, such as decision trees, Bayes classifiers, and neural networks, can easily be generalized to monolithic k-way classifiers to handle multiclass classification tasks. However, in cases where these classifiers are required to learn a very complex decision boundary, they often produce unacceptable accuracy due

* Corresponding author. Tel.: +1 212 523 7302; fax: +1 212 523 5377 (W.L. Cleveland), tel.:/fax: +1 212 666 2393 (X. Long).
E-mail addresses: xl2002@columbia.edu (X. Long), wlcl@columbia.edu (W.L. Cleveland).

0262-8856/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2007.07.004

to the limited representational capability of the learning algorithms and the limited availability of training samples [4,5]. This has led to a search for alternatives.

Since binary classification has been well developed, a natural alternative to monolithic k-way classifiers is to reduce the multiclass problem to a set of binary classification problems. Intuitively, there are two straightforward ways to accomplish this. The first possibility is to apply a classifier between one class and the remaining k-1 classes (called "1 vs. all" or "1 vs. rest" method). In the second approach, a classifier is trained between each pair of classes (called the 1 vs. 1 approach). In both cases, we are faced with the possibility of indecisive or contradictory results [3]. Furthermore, error analysis also shows that, in both 1 vs. all and 1 vs. 1 cases, poor results can be produced by the ensemble of classifiers, even though the error rates for individual classifiers are acceptable. For example, in the 1 vs. all case, suppose that n binary classifiers are used to output n hypotheses $h_1, h_2, \ldots, h_n$, each with (fractional) training error $e_1, e_2, \ldots, e_n$, respectively, Guruswami and Sahai have proved that the worst-case training error for the ensemble is $\min\{\sum_{i=1}^{n} e_i, 1\}$; and for randomized situations, the error is $\min\{\frac{n-1}{n}\sum_{i=1}^{n} e_i, 1\}$. In many practical cases, these errors are unacceptably high [6].

This problem has led to more sophisticated strategies that use a high degree of classifier redundancy. In these strategies, a large number of independently constructed classifiers "vote" on the correct class for a test sample. The "bagging" technique, for instance, first generates multiple training sets by sampling with replacement, and then trains a classifier on each generated set [7]. "Boosting" can be viewed as a special case of bagging where the sampling is adaptive, concentrating on misclassified training instances [8]. These approaches have been proven to greatly reduce classification errors in practice. However, available evidence suggests that they can only reduce the variance errors that result from random variation and noise in the learning sample and from random behavior in the learning algorithm. Bias errors, which result from systematic errors of the learning algorithm, can not be reduced by these techniques [9]. In 1995, Dietterich and Bakiri developed the Error Correcting Output Coding (ECOC) method [10], which has been shown to reduce both variance and bias errors [9,11].

Recent work has shown that ECOC offers further improvement in applications ranging from face verification [12], text classification [13,14], and cloud classification [15] to speech synthesis [16]. These promising results have led us to explore the use of ECOC in automatic cell recognition algorithms for high throughput robotic systems.

Unlike previous ECOC applications, in which ECOC can only be used for classification purposes (i.e. predicting class memberships for new samples), in a high throughput robotic system, it is often necessary not only to identify the class of a cell but also to determine its position relative to pixel coordinates, since tracking and manipulation of cells can be of critical importance. Prior ECOC methods therefore are not applicable to these systems.

In our previous work, which considered binary classification problems, we successfully achieved both classification and localization by a pixel patch decomposition method. In this method, pixel patches from the original images are mapped to "confidence values" that reflect the estimated class probability. Patches containing centered cells give the highest probability and thereby provide the localization (see Section 3 for more details) [1,2]. Here, we develop a new ECOC-based probability estimation algorithm to enable the pixel patch decomposition technique to be used for both multiclass classification and localization in images. To our knowledge, a problem of this type has not been successfully resolved before. It should also be pointed out that this new algorithm is useful not only in a high throughput robotic system, but also in all multiclass classification problems that need additional probability information.

Currently, a popular approach for multiclass probability estimation is proposed by Hastie and Tibshirani [17], In this method, the multiclass probability estimation is obtained by coupling results from pairwise (1 vs. 1) comparisons. In this paper, we generalize their approach to cases where each binary problem involves comparison of data from two "teams" (of classes) that are generated by ECOC. The class probability of each individual class is estimated through team comparisons. In one implementation using this new algorithm with Support Vector Machines (SVMs) [5,18] as base binary classifiers, we are able to subtype and localize cells in bright field images of cell mixtures prepared by mixing cells from three different cell lines. The experimental results suggest that our algorithm can reduce classification errors to the point where some practical applications are possible.

2. Materials and experimental conditions

Both microspheres and living cells were used for training and testing classifiers. The microspheres were 25 μm-diameter, red and 40 μm-diameter, green fluorescent polymer microspheres from Duke Scientific (Cat. No. 36-5, 36-7). The cell lines were K562 (human chronic myelogenous leukemic cells, ATCC; Cat. No. CCL-243), CR10.PF.G cells (obtained from D. J. Volsky) and EAT cells (Ehrlich Ascites Tumor cells, ATCC; Cat. No. CCL-77). All cells were grown at 37.0 °C in BM+1/2 TE1+TE2 +10% fetal calf serum (FCS) [19]. For microscope observation, cells in culture medium were dispensed into polystyrene 96-well microplates, which have glass bottoms that are 0.175 mm thick. Cell viability was determined by nigrosine staining [20] before and after microscope observation and was greater than 95%.

To obtain an accurate training and testing standard, the fluorescent probes for living cells (CellTracker™ CAT. No. C2925 and C34552, Molecular Probes) were used to label CR10 (green, fluorescein bandpass) and EAT (red, propidium iodide) cells. K562 cells were unlabelled. Under bright field illumination, these labels are invisible.

An Arcturus Pixcell II inverted microscope equipped with a 20× planachromat objective (Numerical Aperture: 0.4) and a Hitachi model KP-D580-S1 CCD color camera was used to obtain digitized images. For each microscope field, a set of three images was acquired (Fig. 1). One image was acquired with bright field illumination and was used for SVM training or testing. Two auxiliary fluorescence images were also acquired to distinguish different cell lines, which were either unlabelled or labeled red or green.

Sixty sets of microscope images were acquired and used in our cell detection experiments. In each experiment, two subsets were extracted: one exclusively for training and another exclusively for testing. Ambiguous objects showing both red and green fluorescence were manually deleted. The deleted objects were a very small percentage of the total number of cells.

The computer programs were written in MATLAB and C++. Our algorithm was implemented with the LIBSVM version 2.5 [21], which was compiled as a dynamic link library for MATLAB. All experiments were implemented in the environment of MATLAB Version 6.5.0.180913a (R13) supplemented with Image Processing Toolbox Version 3.2. A standard PC equipped with an Intel Pentium 4/2.8G processor and 256-MB RAM was used.

3. Overall framework for cell detection

In this section, an ECOC-based cell detection framework for bright field images of cultured cells is presented. The framework employs the multiclass classification and probability estimation ability of our proposed algorithm to analyze bright field images of cell mixtures. It permits not only the identification of the desired cells but also gives their locations relative to the pixel coordinates of the primary image. It also uses pixel patches as the primary input data elements. Essentially, the software is taught to classify pixel patches into different classes. Each class corresponds to a single cell type, except for the larger class containing all undesired objects (e.g. background, fragments of cells, trash), denoted as "Non-cell".

The essential aspects of this framework are illustrated in Fig. 2. Basically, we first train an ensemble of SVM classifiers with ECOC. This is done with input vectors that are derived from manually-extracted training patches and are represented as linear combinations of feature vectors derived in Principal Component Analysis (PCA) preprocessing [1,22].

For each pixel p in the testing image (excluding pixels in the margin around the edges), a pixel patch centered at that pixel is extracted and represented in the same way as that in training process. The probability that this extracted patch belongs to each class is calculated by ECOC probability

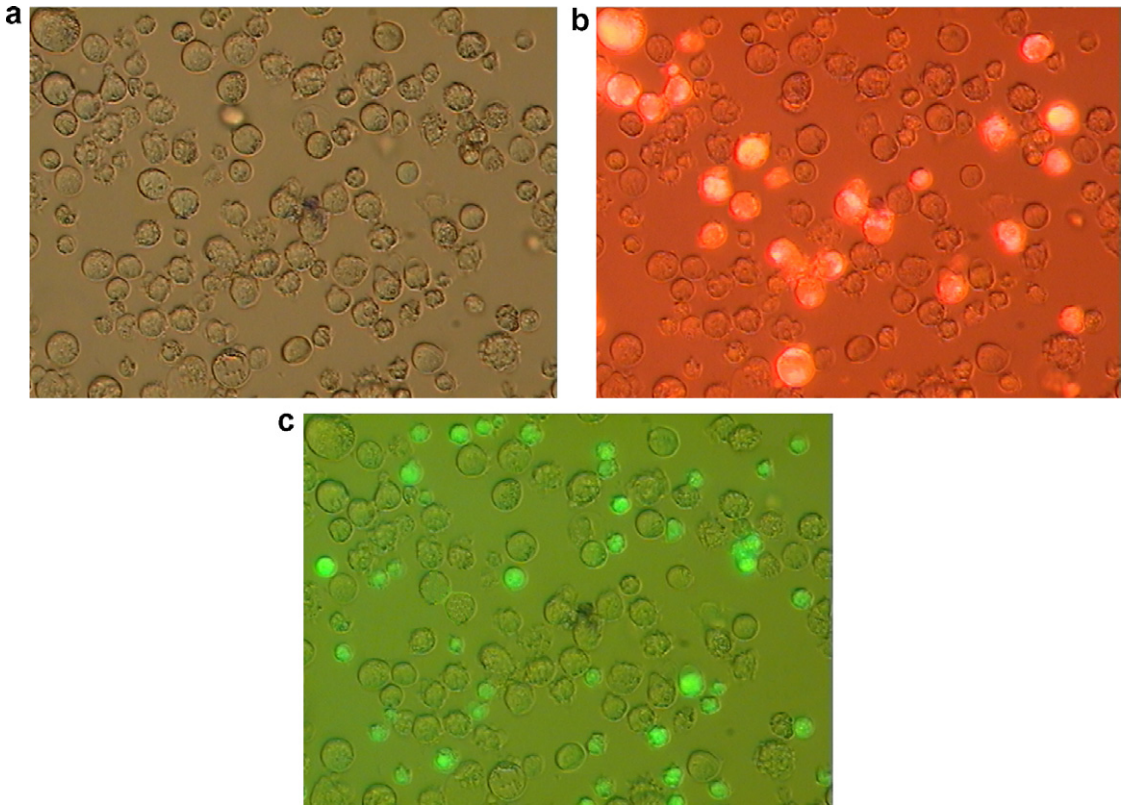

Fig. 1. Typical sample images: (a) bright field image; (b) superposition of the bright field and the red fluorescence image; (c) superposition of the bright field and the green fluorescence image.

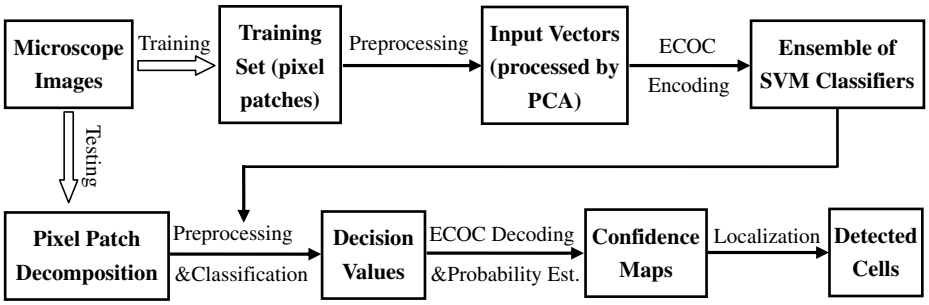

Fig. 2. Illustration of the overall multiclass cell detection process with ECOC probability estimation.

estimation. For each class corresponding to a cell type, this probability is then used as a "confidence value" $C[p] \in [0,1]$ in a "confidence map" for that cell type. Pixels in each confidence map are the confidence values of their corresponding patches in the original image and form "mountains" with large peaks representing a high probability of presence of the corresponding cell type. A given peak in a confidence map is compared with the corresponding peaks in the other confidence maps. The confidence map with the highest peak at that location gives the assignment of class membership. Localization is provided by the pixel coordinates of the highest peak. It should be pointed out that generating a confidence map for the "Non-cell" class is unnecessary in our case since localization of the non-cell objects is not important for us.

As has been mentioned above, in the ECOC approach, binary classifiers have to be trained as the base classifiers. The choice of base classifier can be arbitrary. In this work, we used Support Vector Machines (SVM) [5,18] with the RBF kernel $K(x,y) = e^{-\gamma\|x-y\|^2}$. The SVM classifier in our experiment is implemented by modifying LibSVM [21]. The regularization parameter C and the kernel parameter γ are optimized using a two-step "grid-search" method for each classifier [21]. In the first step, a coarse grid-search with a grid size of 1 was used to localize a Region of Interest (ROI) containing the optimal values (shown in Fig. 3). In the second step, a fine grid-search over the ROI with a grid size of 0.25 was used to give more precise values for C and γ. The result is shown in Fig. 4.

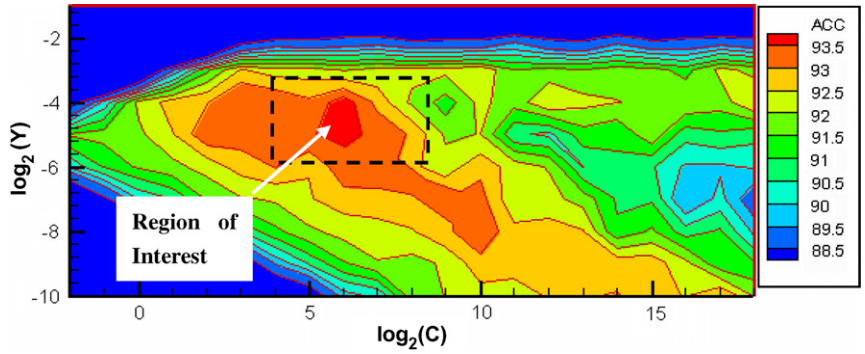

Fig. 3. Coarse grid search, grid size = 1.

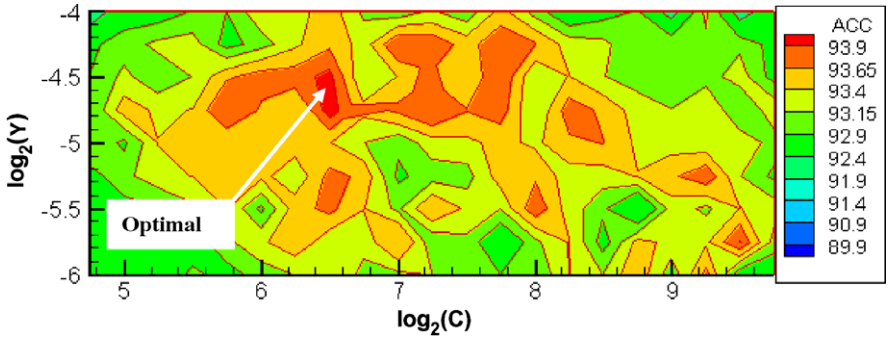

Fig. 4. Fine grid search, grid size = 0.25.

4. Extending ECOC for probability estimation

4.1. Brief summary of ECOC

As noted above, our cell identification and localization algorithm requires mapping each pixel patch in the image into a set of three "confidence values", which reflect the estimated class probabilities. For this mapping, multiclass probability estimation is needed. Since the standard ECOC method simply assigns a class label to each sample (i.e., they do not output the conditional probability of each class $P(class = c|X = x)$ given a sample x), we need to extend it to enable probability estimation. Our development of the probability estimation algorithm requires a consideration of some of the fundamental aspects of ECOC, which are briefly described in this section. A detailed introduction to ECOC can be found in [10,23].

The ECOC approach essentially proceeds in two steps: training and classification. In the first step, the multiclass classification problem is decomposed into training l binary classifiers on l dichotomies of the instance space. Assuming k classes and l classifiers, each such decomposition can be represented by a coding matrix $C \in \{-1, 0, +1\}^{k\times l}$ which specifies a relation between classes and dichotomies. If $C(i,j) = +1$ (or $C(i,j) = -1$) then the samples belonging to class i ($1 \leqslant i \leqslant k$) are considered to be positive (or negative) samples for training the jth ($1 \leqslant j \leqslant l$) binary classifier, f_j. If $C(i,j) = 0$, then the samples belonging to class i are not used in training f_j. Thus a binary learning problem is built for each column of the matrix. Each class i is encoded by the ith row of the matrix C. This codeword is denoted by C_i. To classify a new instance x, the vector formed by the output of the classifiers $F(x) = (f_1(x), f_2(2) \ldots, f_l(x))$ is computed and is assigned to the class whose codeword C_i is closest to $F(x)$. In this sense, the classification can be seen as a decoding operation:

$$\text{Class of input } x = \arg\min_{i=1}^{k} d(C_i, F(x)) \tag{1}$$

where $d()$ is the decoding function.

Different decoding functions have been reported in the literature. For example, Dietterich and Bakiri initially used a simple Hamming distance [10]. In the case where margin-based classifiers are used, Allwein et al. showed the advantage of using a loss-based decoding function [23]. The loss-based function is typically a non-decreasing function of the margin and thus weights the confidence of each classifier according to the margin. However, no formal results exist that suggest the optimal choice of the decoding function. In this paper, we tried two of the most commonly used loss-based functions:

$$\text{L1 norm based function } d(C_i, F(x)) = \sum_{j=1}^{l} |C_{ij} - F_j(x)| \tag{2}$$

$$\text{and L2 norm based function } d(C_i, F(x)) = \sum_{j=1}^{l} (C_{ij} - F_j(x))^2 \tag{3}$$

Because the codewords come from an error-correcting code, the ECOC method introduces redundancy into the system by training decision boundaries multiple times. Even if some of the individual classifiers were wrong for a specific instance x, the ECOC method can still classify x in the right class. ECOC, therefore, can greatly increase the classification accuracy. It is worth noting that both 1 vs. all and 1 vs. 1 are special cases of the ECOC framework. 1 vs. all is equivalent to linear decoding with a coding matrix whose entries are always -1 except diagonal $+1$ entries. 1 vs. 1 is also equivalent to Hamming decoding with the appropriate coding matrix.

4.2. Extension of standard ECOC for probability estimation

In this section, we modify the standard ECOC method to enable probability estimation. Our new algorithm is an extension of the pairwise coupling method introduced by Hastie and Tibshirani [17]. It should also be noted that, while this work was in progress, Huang et al. independently developed a very similar algorithm based on the same strategy and formulated it as "Generalized Bradley-Terry Model" [24]. To our knowledge, they have not applied their algorithm to practical applications.

4.2.1. Hastie –Tibshirani method for pairwise coupling

The Hastie and Tibshirani's pairwise coupling method can be briefly described as follows. Assume that after training a classifier using the samples from class i (labeled $+1$) and samples from class j (labeled -1), the pairwise probability estimation for every class i and j ($i \neq j$) is $r_{ij}(x)$. According to the Bradley–Terry (BT) model [24], $r_{ij}(x)$ is related to the class posterior probabilities $p_i = P(class = i|X = x)$ ($i = 1,2,\ldots k$):

$$\begin{aligned} r_{ij}(x) &= P(class = i|class = i \cup class = j, X = x) \\ &= p_i(x)/(p_i(x) + p_j(x)) \end{aligned} \tag{4}$$

Note that p_i is also constrained by $\sum_{i=1}^{k} p_i(x) = 1$. There are $k-1$ variables but $k(k-1)/2$ constraints. When $k > 2$, $k(k-1)/2 > k-1$. This means that there may not exist p_i exactly satisfying all constraints. In this case, one must use the estimation

$$\hat{r}_{ij}(x) = \hat{p}_i(x)/(\hat{p}_i(x) + \hat{p}_j(x)) \tag{5}$$

In order to get a good estimation, Hasti and Tibshirani use the average Kullback–Leibler distance between $r_{ij}(x)$ and $\hat{r}_{ij}(x)$ as the closeness criterion, and find the P that maximizes the criterion.

$$l(P) = \sum_{i<j} n_{ij} \left[r_{ij} \log \frac{r_{ij}}{\hat{r}_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \hat{r}_{ij}} \right] \tag{6}$$

this is equivalent to minimizing the negative log-likelihood:

$$l(\hat{P}) = -\sum_{i<j} n_{ij}\left[r_{ij}\log\frac{\hat{p}_i}{\hat{p}_i+\hat{p}_j} + (1-r_{ij})\log\frac{\hat{p}_j}{\hat{p}_i+\hat{p}_j}\right] \quad (7)$$

where n_{ij} is the number of training samples used to train the binary classifier that predicts r_{ij}.

This can be solved by a simple iterative algorithm:

1. Initialize $P=[p_1,p_2,\ldots p_k]$ with random $p_i(x)>0$, $i=1,2,\ldots k$.
2. Repeat ($j=1,2,\ldots k,1,2,\ldots$) until convergence:
   (a) Calculate corresponding $\hat{r}_{ij}(x)=p_i(x)/(p_i(x)+p_j(x))$.
   (b) Calculate $\hat{P}=\left[p_1,\ldots,p_{j-1},\frac{\sum_{i:i\neq j}n_{ji}r_{ji}}{\sum_{i:i\neq j}n_{ji}\hat{r}_{ji}}p_j,p_{j+1},\ldots,p_k\right]^{\mathrm{T}}$.
   (c) Update $P=\hat{P}/\sum\hat{p}_i$.

4.2.2. Generalizaion of Hastie–Tibshirani method

It has been mentioned above that pairwise coupling is a special case of ECOC. With some generalization, Hastie and Tibshirani's pairwise strategy can be extended to ECOC with any arbitrary code matrix C. A close look at the ECOC code matrix reveals that it actually divides the samples from different classes into two groups for each binary classifier: the ones labeled "+1" and the ones labeled "−1". In this sense, ECOC with any arbitrary code matrix is equivalent to pairwise group coupling. Therefore we can generalize Hastie and Tibshirani's results to cases where each binary problem involves data in two "teams" (two disjoint subsets of samples), i.e. instead of comparing two individuals, we can compare two groups that are generated by ECOC and estimate the individual probabilities through the group comparisons.

Assuming an arbitrary code matrix C, for each column i of C, we have

$$\begin{aligned} r_i(x) &= P(class\in I_i^+ | class\in I_i^+\cup I_i^-, X=x) \\ &= \frac{\sum_{class\in I^+}p_{class}(x)}{\sum_{class\in I^+\cup I^-}p_{class}(x)} \end{aligned} \quad (8)$$

where I_i^+ and I_i^- are the set of classes for which the entries in the code matrix $C(*,i)=+1$ and $C(*,i)=-1$. If we define

$$q_i=\sum_{j\in I_i^+\cup I_i^-}p_j,\quad q_i^+=\sum_{j\in I_i^+}p_j,\quad q_i^-=\sum_{j\in I_i^-}p_j \quad (9)$$

Similar to pairwise comparison, we need to minimize the negative log-likelihood

$$\min_P\ l(P)=-\sum_{i=1}^{l}n_i\left[r_i\log\frac{q_i^+}{q_i}+(1-r_i)\log\frac{q_i^-}{q_i}\right] \quad (10)$$

where n_i is the number of training samples of the binary classifier that corresponds to the ith column of the code matrix. Above equation can be solved by a slightly more complex iterative algorithm listed below. This algorithm is equivalent to a special case on probability estimation of Huang et al.'s Generalized Bradley–Terry Model in [24]. Since the convergence of Generalized Bradley-Terry Model has been proven [24], the algorithm is also guaranteed to converge.

1. Initialize $P=[p_1,p_2,\ldots p_k]$ with random $p_i(x)>0$, $i=1,2,\ldots k$.
2. Repeat ($j=1,2,\ldots,k,1,2,\ldots$) until $\partial l(P)/\partial p_i=0$, $i=1,\ldots,k$ are satisfied.
   (a) Calculate corresponding q_i^+,q_i^-,q_i, $i=1,2,\ldots l$.
   (b) Calculate $\hat{P}=[p_1,\ldots,p_{j-1},\frac{\sum_{i:j\in I_i^+}\frac{n_ir_i}{q_i^+}+\sum_{i:j\in I_i^-}\frac{n_i(1-r_i)}{q_i^-}}{\sum_{i:j\in I_i^+\cup I_i^-}\frac{n_i}{q_i}}p_j,p_{j+1},\ldots,p_k]^{\mathrm{T}}$.
   (c) Update $P=\hat{P}/\sum\hat{p}_i$.

5. Experiments with artificial data

To gain insight into the factors that affect the classification accuracy of our algorithm, we have carried out experiments with artificial 2D data generated by the Matlab random functions. Unlike actual data vectors that have high dimensionality (e.g. 39*39, see below), the artificial 2D data vectors generate results that can be graphically represented and intuitively interpreted.

Four different sets of artificial data have been used (Fig. 5(a–d)). Data set 1 represents a simple scenario, where the classes are well separated. Data sets 2–4 represent progressively more difficult scenarios, with data set 4 having a very large class overlap.

The artificial data sets used in this section were constructed as follows. We first constructed a 2D data set which consists of four different multivariate normal distribution classes. After creating the first data set, three different data sets, each with the same covariance and sample numbers but different mean vectors were also constructed. Each class was given 300 samples. The covariance matrices of the four classes were (same for all data sets): $\sigma_1=\begin{bmatrix}1.5 & 0.4\\ 0.4 & 1.5\end{bmatrix}$, $\sigma_2=\begin{bmatrix}3 & 0.1\\ 0.1 & 4\end{bmatrix}$, $\sigma_3=\begin{bmatrix}2 & 0.2\\ 0.2 & 3\end{bmatrix}$ and $\sigma_4=\begin{bmatrix}3 & 0.8\\ 0.8 & 8\end{bmatrix}$. The mean vectors of the four classes for each data set are summarized in Table 1.

5.1. Reconstruction of probability distribution from ECOC probability estimation

To evaluate directly our proposed ECOC probability estimation algorithm, we used it to estimate the known probability distributions of the above artificial 2D data sets. In this experiment, ECOC was implemented with a sparse matrix that was selected from 10,000 randomly generated 4×10 matrices. To select the optimum matrix in the set of 10,000, we calculated the minimum Hamming distance between all pairs of the rows for each matrix. The matrix with the biggest minimum distance was chosen

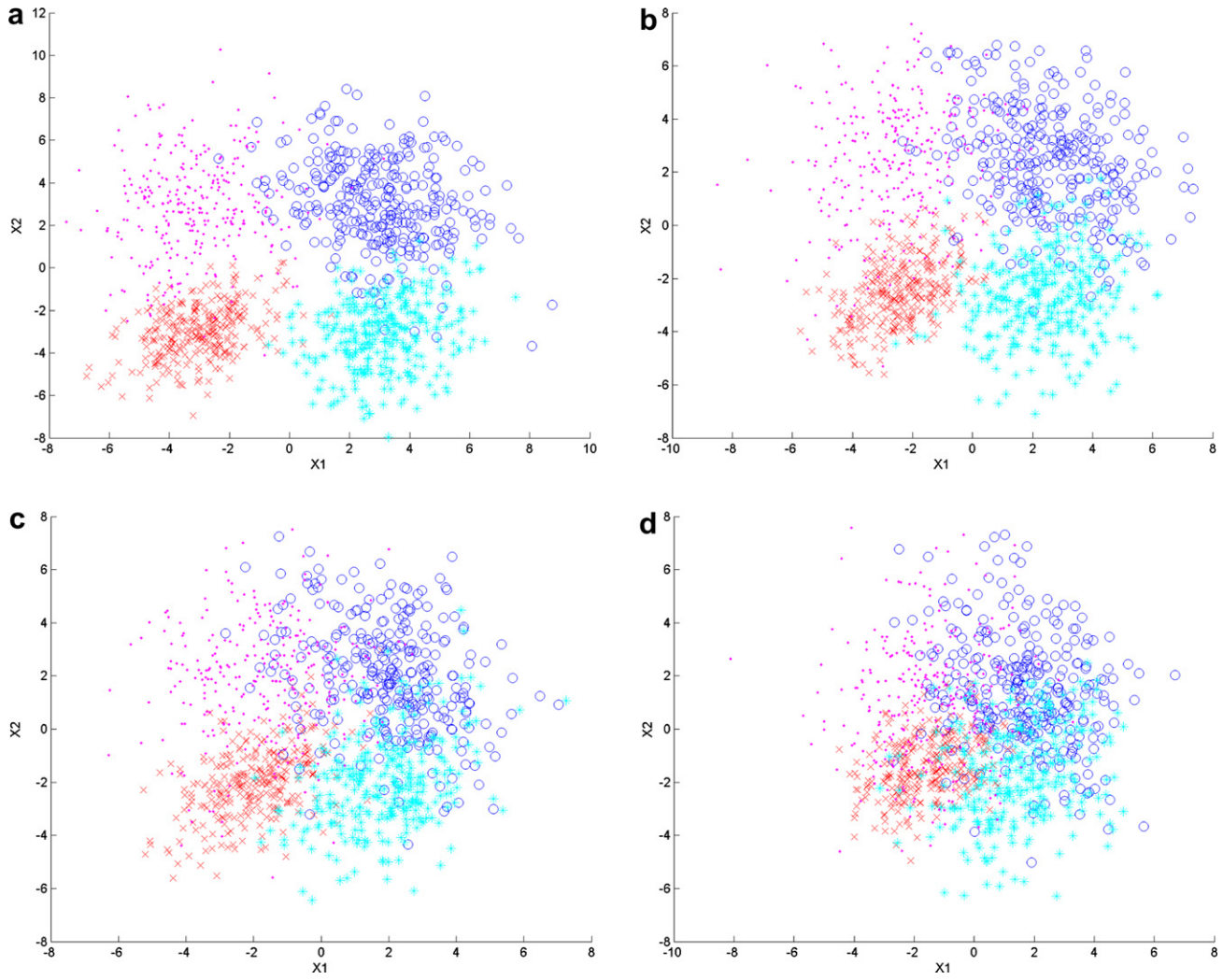

Fig. 5. Data sets used in the simulation experiment. Class number: 4; Sample number in each class: 300; Class distribution: Normal distribution. The four data sets have same covariance but different mean for each class. (a) Data set 1; (b) Data set 2; (c) Data set 3 and (d) Data set 4.

Table 1
Mean vectors used to generate artificial data for Datasets 1, 2, 3 and 4

	μ_1	μ_2	μ_3	μ_4
Data set 1	$[-3 \quad -3]^T$	$[3 \quad 3]^T$	$[3 \quad -3]^T$	$[-3 \quad 3]^T$
Data set 2	$[-2.5 \quad -2.5]^T$	$[2.5 \quad 2.5]^T$	$[2.5 \quad -2.5]^T$	$[-2.5 \quad 2.5]^T$
Data set 3	$[-2 \quad -2]^T$	$[2 \quad 2]^T$	$[2 \quad -2]^T$	$[-2 \quad 2]^T$
Data set 4	$[-1.5 \quad -1.5]^T$	$[1.5 \quad 1.5]^T$	$[1.5 \quad -1.5]^T$	$[-1.5 \quad 1.5]^T$

[21]. Since the four artificial 2D data sets have known distributions, the ideal probability distributions of the classes can be easily calculated. Fig. 6 plots the ideal class probability of the samples in Data set 2 against their coordinates. The ECOC-reconstructed class probability distribution is shown in Fig. 7. As one can see from the figures, the reconstructed probability distribution matches the ideal distribution very well.

Fig. 8 gives a quantitative evaluation of the mean square error (MSE) of the ECOC probability estimation. This result is shown in comparison with the result provided by the pairwise coupling method proposed of Hastie and Tibshirani [17]. As indicated in the figure, our ECOC algorithm is superior to the pairwise coupling method for three of the four test classes. Therefore, ECOC probability estimation has a higher overall accuracy.

5.2. Comparison of extended ECOC with other methods

Using the above artificial 2D data sets, we systematically compared the proposed ECOC probability estimation method with other widely used approaches: (1) 1 vs.all; (2) 1 vs. 1 (pairwise coupling by Hastie and Tibshirani); (3) ECOC with Hamming decoding; (4) ECOC with L1-Norm based decoding and (5) ECOC with L2-Norm based decoding. We used randomly generated sparse code matrices as described in Section 5.1 for all ECOC-based methods. For inconsistent labels (ties and contradictory votes), we adapted the strategy described in [3] and randomly chose labels for them. Results are shown in Fig. 9. As expected, ECOC-based methods are generally superior to non-ECOC approaches, i.e. 1 vs. all and 1 vs. 1. Even within ECOC-based methods, the extended ECOC with probability estimation method produces the highest

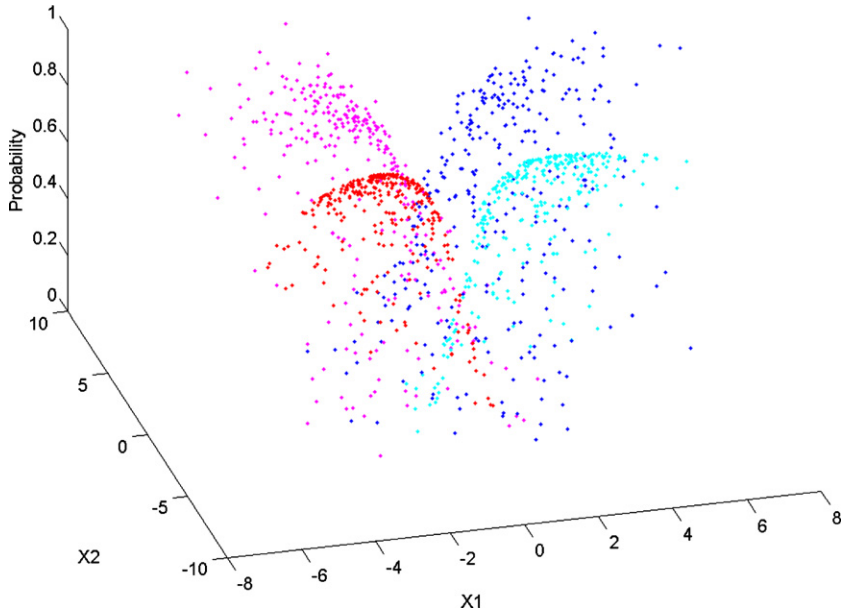

Fig. 6. Ideal class probability distribution of Data set 2.

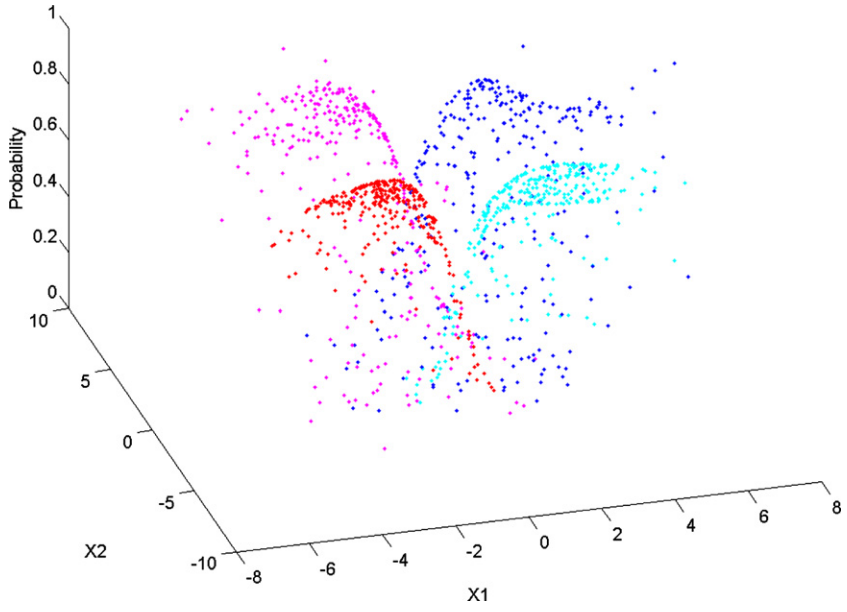

Fig. 7. Class probability distribution of Data set 2, estimated by our ECOC probability estimation method.

classification accuracy. Finally and most interestingly, all candidate methods perform very well on Data set 1, which represent a very simple case. However, as the scenario gets more and more complex, ECOC-based methods show a greater advantage over other approaches.

We hypothesized that the superiority of ECOC methods was largely due to the fact that these methods generated more decision boundaries, which can greatly reduce inconsistent labeling areas, i.e. areas in which sample points can not be consistently labeled using the majority voting strategy. To verify this hypothesis, the decision boundaries of different candidate methods were plotted and compared (Fig. 10). Fig. 10(a) and (b) show decision boundaries of Data set 1 that are generated by the 1 vs. all and ECOC probability estimation method, respectively. Fig. 10(c) and (d) show those of Data set 4. One can see that although there exist many areas with inconsistent labeling in Fig. 10(a), most areas are very close to class interfaces. Since there is little overlap in Data set 1, few sample points fall into these areas. Therefore, 1 vs. all method works almost as well as ECOC probability estimation, which dramatically eliminates the inconsistent labeling areas (Fig. 10 (b)). On the other hand, since there is a large overlap in Data set 4, a great proportion of the sample points fall into the inconsistent labeling areas when the 1 vs. all method is used. In this case, ECOC probability estimation outper-

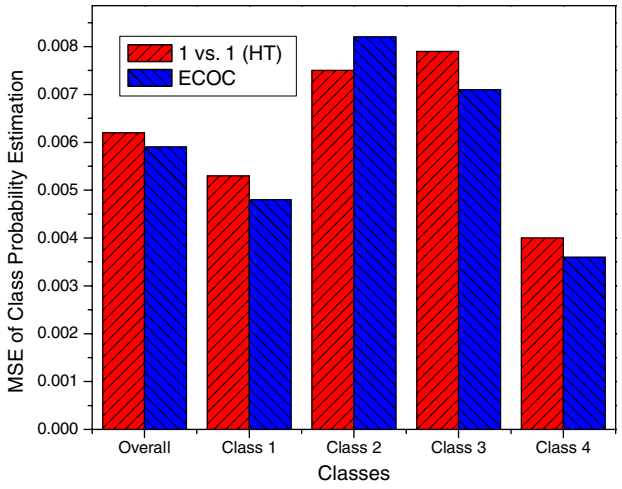

Fig. 8. Comparison of probability estimation errors of our ECOC probability estimation and the pairwize (1 vs. 1) probability estimation by Hastie and Tibshirani. The ideal probability distribution was used as reference.

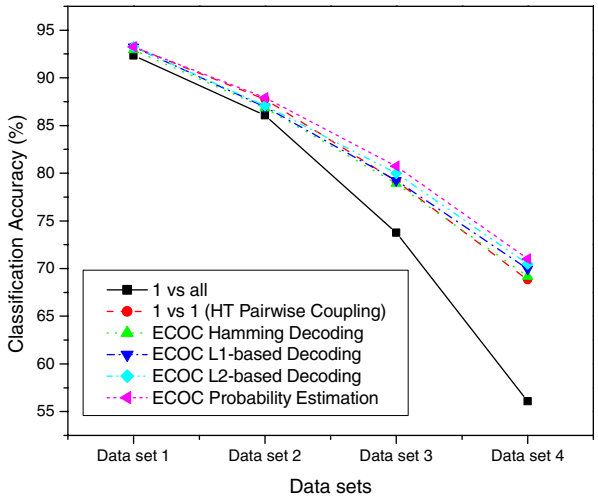

Fig. 9. Classification accuracy of different methods on the artificial data sets. The methods used are (1) 1 vs.all; (2) 1 vs. 1 by Hastie and Tibshirani; (3) ECOC with Hamming decoding; (4) ECOC with L1-Norm based decoding; (5) ECOC with L2-Norm based decoding; (6) ECOC with probability estimation.

formed the 1 vs. all method by a very large margin. Our hypothesis is very consistent with the experimental results shown in Fig. 9.

6. Experiments with bright field images of living cells

In this section, we evaluate quantitatively the extended ECOC-based cell detection method for bright field images of cell mixture prepared by mixing cells from three different cell lines. The overall framework of this approach has been described in Section 3. In what follows, the detailed experiment is described in steps. The experimental result is also quantitatively analyzed.

6.1. Pixel patch extraction and construction of preclassified training set

Since individual cells typically occupy only a small percentage of total image area, it is advantageous to decompose an image using pixel patches that just large enough to contain the largest cells in the image. In actual experiments, 39×39 pixel patches centered at all possible locations in the 640×480 microscope image were extracted (except in the 20-pixel margin around the edges). Our experiments indicate that performance is not very sensitive to small variations in patch size, e.g. a patch size of 37×37 produced similar results (data not shown). Since many locations in the image are uniform background, a "mask" was created to exclude these patches. Essentially, the "mask" eliminated all pixel patches whose average pixel intensities were below a user-chosen threshold.

A training set was created with the aid of an interactive program that displays the digitized microscope images and allows a user to select the locations of cell centers with a mouse cursor after manual comparison of bright field and fluorescence images. For each cell type, the pixel patches extracted from the selected cell locations were preprocessed by PCA [1,22] and used as input vectors of that class. The pixel patches in the "Non-cell" class were then generated automatically by extracting all the pixel patches whose centers were $r \geqslant 8$ pixels away from any of the manually selected cell locations. The value of r was empirically chosen in relation to the sizes of cells and pixel patches. PCA preprocessing was used to reduce dimensionality to 10 for all input vectors. After all input vectors are preprocessed, each attribute of the PCA-preprocessed vectors was linearly scaled to the range $[-1,+1]$. The main advantage of scaling is to avoid computational difficulties and to avoid the dominance of attributes with greater numeric ranges over those with smaller numeric ranges [21]. Finally, the classes were labeled with ordinal numbers.

6.2. ECOC training

We followed the procedure described in the simulation experiment and used randomly generated sparse code matrices for all ECOC-based methods in this section. For each binary SVM classifier, the parameters are independently optimized following the aforementioned two-step grid search procedure. During the process of binary classifier training, the Compensatory Iterative Sample Selection (CISS) algorithm [2], a new SVM training procedure which we developed previously, was employed to address the imbalance problem caused by the large "Non-cell" sample set. This algorithm maintains a fixed-size "working set", in which the training samples are kept balanced by iteratively choosing the most representative training samples for the SVM. These samples are close to the boundary and are therefore more difficult to classify. This scheme can make the decision boundary more accurate, especially when applied to difficult scenarios.

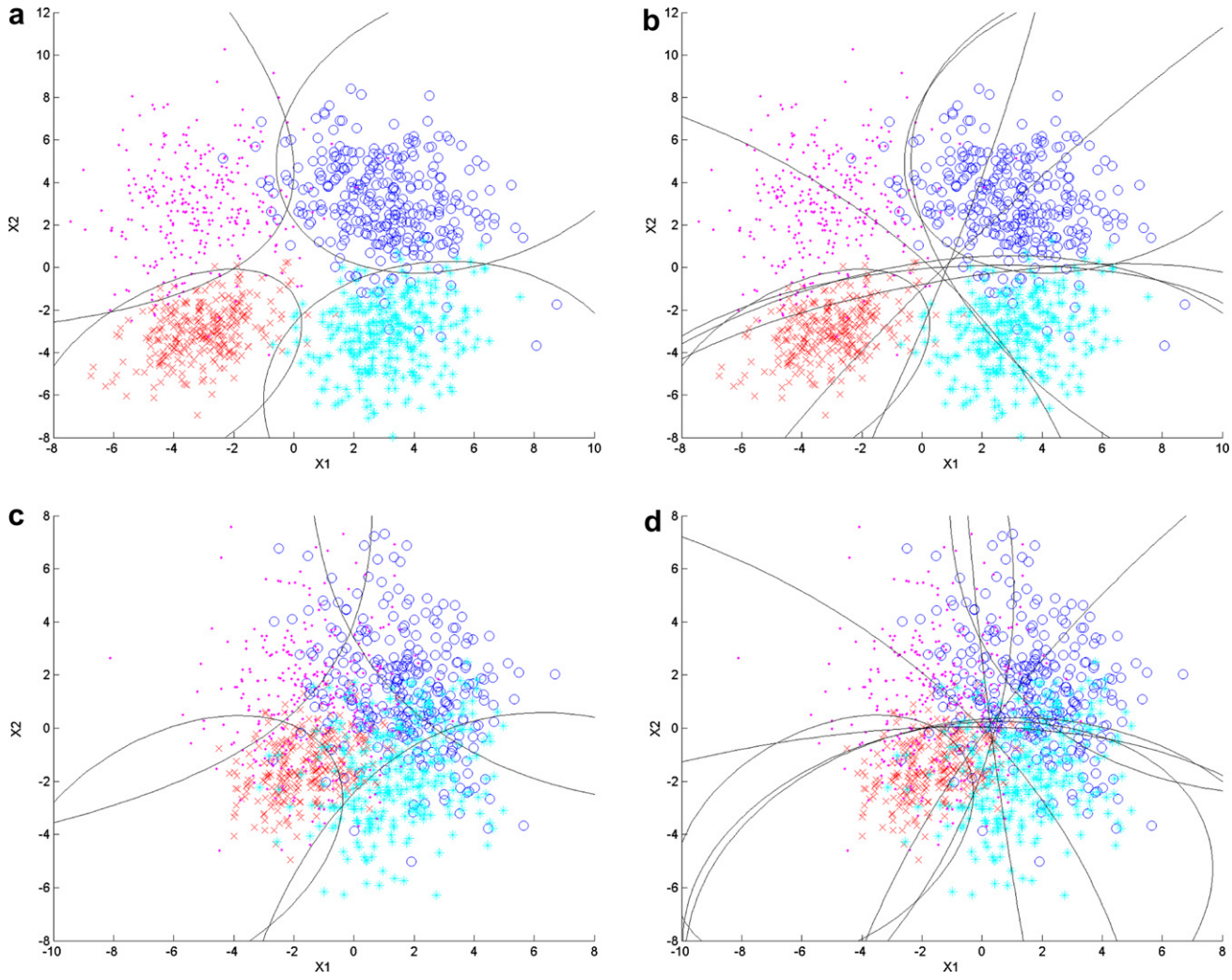

Fig. 10. Examples of decision boundaries generated by different methods on Data sets 1 and 4. (a) 1 vs. all on Data set 1; (b) ECOC probability estimation on Data set 1; (c) 1 vs. all on Data set 4; (d) ECOC probability estimation on Data set 4.

6.3. Identification and localization of living cells in bright field images

In order to examine the effect of our algorithm on images with different levels of complexity, three different scenarios were created. In Scenario 1, both red and green fluorescent microspheres were used as two types of model cells and mixed with the K562 cells. Since the microspheres have obviously different size, color and texture from living cells, this scenario represents a very simple case. Scenario 2 is more complex since it is the mixture of only one type of microspheres (red) and cells from two cell lines (K562 and CR10.PF.G). Scenario 3 represents the most complex case where three kinds of living cells (K562, CR10.PF.G and EAT) were mixed without the addition of any microspheres. Typical images from these three scenarios are shown in Fig. 11. For each scenario, there is a total of 4 classes: one for each of the desired objects (microspheres or cells) and one for all objects that are neither cells nor microspheres (the "non-cell" class).

An ensemble of SVM classifiers was trained and tested on each scenario. For each ensemble, testing samples were from the same scenario as the training samples. However, none of the samples used for training were used for testing.

After training, we first tested the classifier ensembles on manually extracted pixel patches. This is done with three testing sets. Each set has 2000 manually extracted pixel patches from one scenario. Testing set 1 consisted of pixel patches of 500 K562 cells, 500 green fluorescent microspheres, 500 red fluorescent microspheres and 500 background from Scenario 1. Testing set 2 consisted of pixel patches of 500 K562 cells, 500 CR10 cells, 500 red fluorescent microspheres and 500 background from Scenario 2. Testing set 3 consisted of pixel patches of 500 K562 cells, 500 CR10 cells, 500 EAT cells and 500 background from Scenario 3. The classification accuracy is shown in comparison with other candidate methods in Fig. 12. It should be pointed out that, in this case, we are actually able to compare our new ECOC algorithm with some standard ECOC methods such as hamming, L1 and L2 based decoding

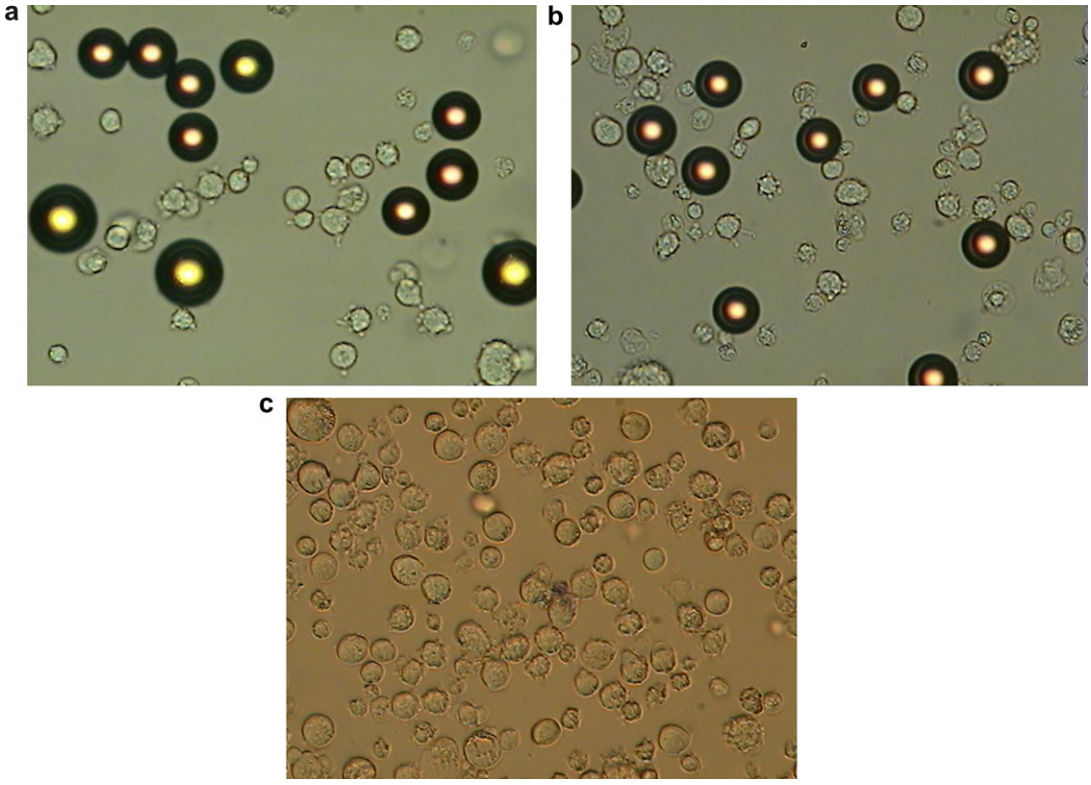

Fig. 11. Sample images for living cell experiment. (a) Scenario 1: mixture of 2 types of microspheres and 1 type of cells; (b) Scenario 2: mixture of 1 type of microspheres and 2 types of cells; (c) Scenario 3: mixture of 3 types of cells.

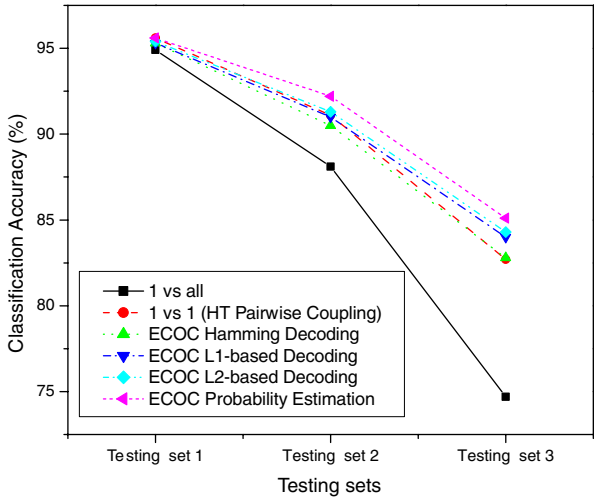

Fig. 12. Classification accuracy of different methods on living cell testing sets. The methods used are (1) 1 vs.all; (2) 1 vs. 1 by Hastie and Tibshirani; (3) ECOC with Hamming decoding; (4) ECOC with L1-Norm based decoding; (5) ECOC with L2-Norm based decoding; (6) ECOC with probability estimation.

because the pixel patches have been manually extracted and therefore no localization is required.

The classifier ensembles were also applied in combination with the pixel patch decomposition method to whole microscope images (640×480). Fig. 13 shows the confidence maps for Fig. 11(c). The range of the confidence value ($[0, 1]$) in the confidence maps has been linearly scaled to $[0, 255]$ for grayscale representation. In Figs. 14–16, the cell and microsphere positions detected are denoted by different symbols (diamond, square and cross, one for each class) in the image.

Statistical cell detection results for whole microscope images in Scenarios 1, 2 and 3 are summarized in Figs. 17–19, respectively. For each scenario, ten testing images (640×480) were used. We employed a "Free-response Receiver Operating Characteristics" method (FROC) [25] with the average false positive (FP) number of all cell types in each image and the average sensitivity (true positive percentage, i.e., the percentage of cells that are identified correctly) of all cell types as performance indexes. As described above, the cell positions are identified as "peaks" of the "mountains" in the confidence maps. This requires a user-defined threshold for the definition of "peak". The FROC curve plots the relationship of false positives and sensitivity as a function of the threshold (not explicitly represented in the plot). In a practical application, a suitable threshold can then be selected to achieve the required behavior. Generally speaking, the bigger the area under the curve, the better the result is. A total of three methods were compared in the experiment: (1) 1 vs.all; (2) 1 vs. 1 by Hastie and Tibshirani; (3) ECOC with probability estimation. It should be noted that other standard ECOC methods are not applicable here since they can not provide information needed for localization.

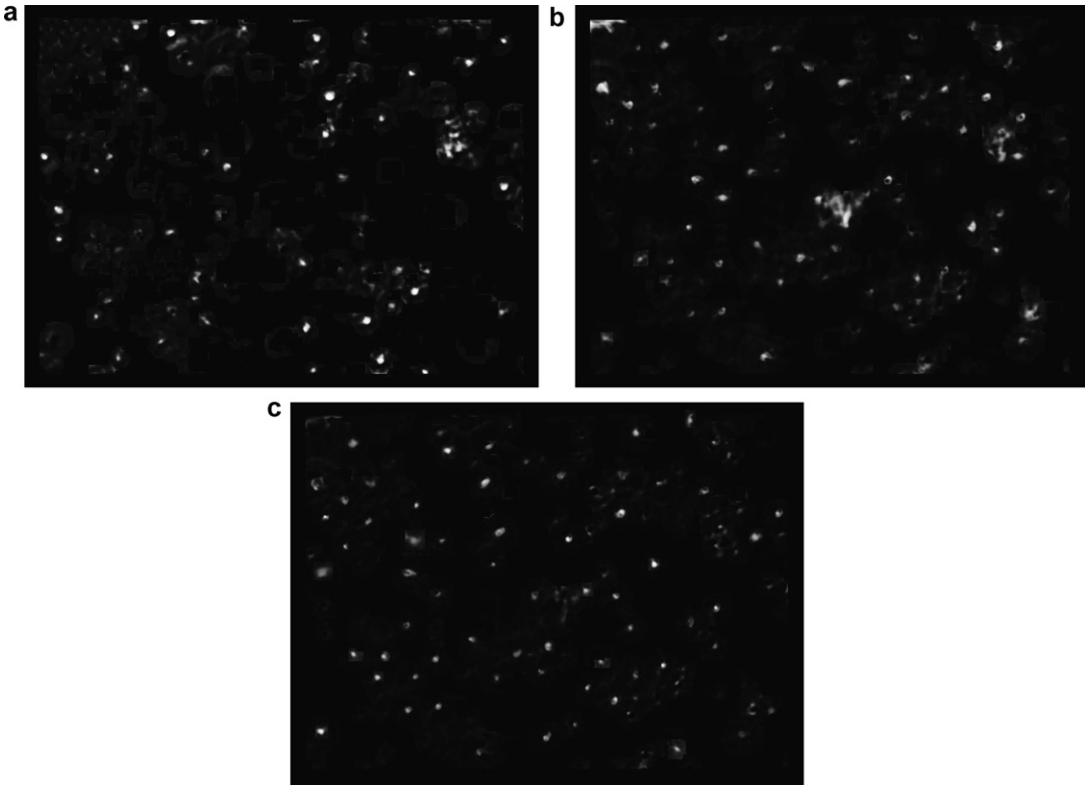

Fig. 13. Confidence maps for Fig. 11 (c). (a) Confidence map for CR10 cells; (b) confidence map for EAT cells; (c) confidence map for K562 cells. The confidence values are linearly scaled to 0–255 for display.

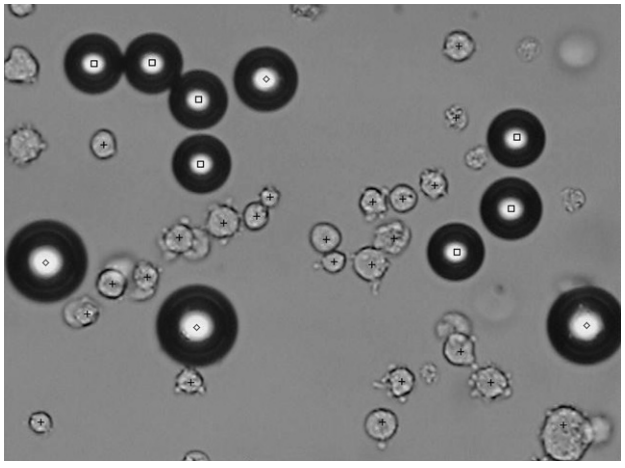

Fig. 14. Detecting result of the image in Scenario 1 using SVM with ECOC probability estimation. The positions detected are denoted by black symbols in the image. Diamond: green fluorescent microspheres; Square: red fluorescent microspheres; Cross: K562 cells.

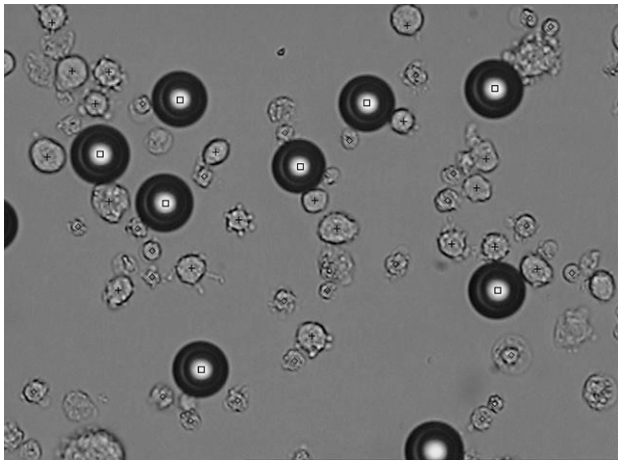

Fig. 15. Detecting result of the image in Scenario 2 using SVM with ECOC probability estimation. The positions detected are denoted by black symbols in the image. Diamond: CR10 cells; Square: red fluorescent microspheres; Cross: K562 cells.

Results with both manually and automatically extracted pixel patches show that for Scenario 1, a very easy case, all methods produce very good results. For Scenario 2, where the images are more complex, our ECOC probability estimation method (and other ECOC-based methods) starts to show some advantage. A much greater advantage is seen in the very difficult case represented by Scenario 3. For example, in Scenario 3, if the average false positive acceptance number in each image is set at 1, ECOC probability estimation achieves a sensitivity of 84.5%, which is 4 percentage points greater than that of 1 vs. 1, and 15 points greater than that of the 1 vs. all approach. The result closely parallels that obtained in the simulation experiments with artificial data as shown in Fig. 9.

As noted previously, our results with the artificial data shown in Figs. 9 and 10 suggest that ECOC-based methods

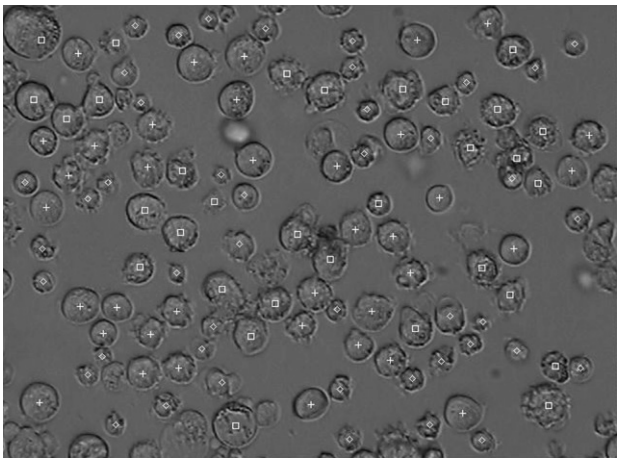

Fig. 16. Detecting result of the image in Scenario 3 using SVM with ECOC probability estimation. The cell positions detected are denoted by white symbols in the image. Diamond: CR10 cells; Square: EAT cells; Cross: K562 cells.

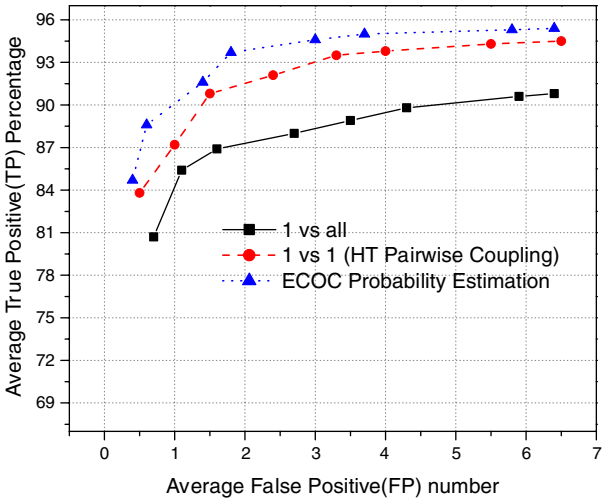

Fig. 18. FROC plots of different candidate methods when applied to Scenario 2: (1) 1 vs.all; (2) 1 vs. 1 by Hastie and Tibshirani; (3) ECOC with probability estimation. The testing set includes 10 images.

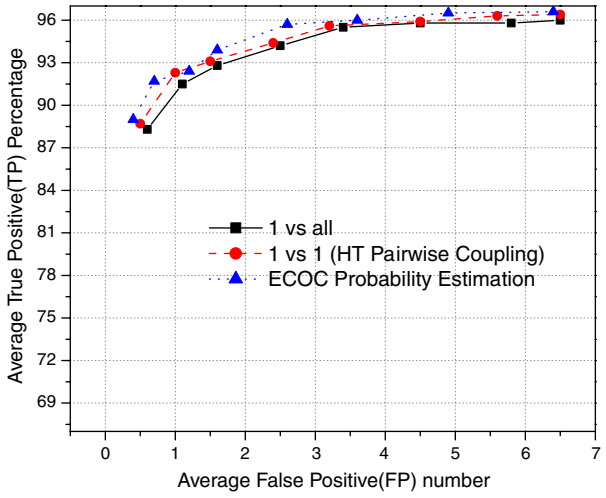

Fig. 17. FROC plots of different candidate methods when applied to Scenario 1: (1) 1 vs.all; (2) 1 vs. 1 by Hastie and Tibshirani; (3) ECOC with probability estimation. The testing set includes 10 images.

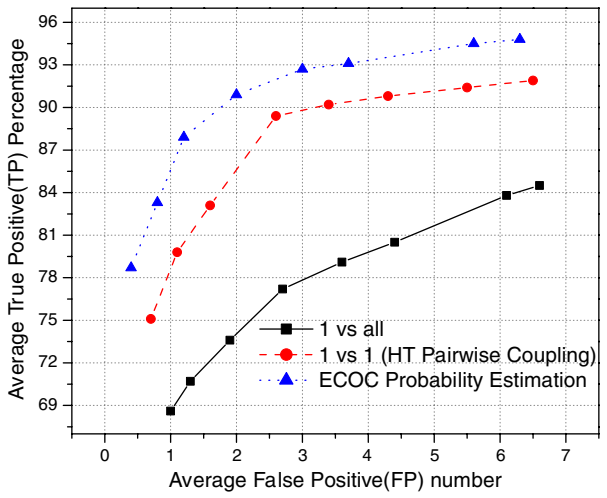

Fig. 19. FROC plots of different candidate methods when applied to Scenario 3: (1) 1 vs.all; (2) 1 vs. 1 by Hastie and Tibshirani; (3) ECOC with probability estimation. The testing set includes 10 images.

can greatly reduce the inconsistent labeling by introducing redundancy, and therefore can partition the sample space more accurately than other methods. The experimental results with living cells described in this section add further support to this claim.

It should also be noted that a close inspection of results yielded by our algorithm suggests that it can distinguish between different cell lines according to the subtle difference in the cell appearance, similar to a human observer. For example, to a human observer, the CR10 cells in the testing images are relatively small, and are rough-looking in texture. The K562 cells and the EAT cells are about the same size. However, the edge and texture of the K562 cells are slightly smoother than those of EAT cells. Our experimental results suggest that our algorithm can actually make these subtle distinctions, and thereby emulate a human expert quite well.

With regard to the processing speed, when our current method is used with a 39×39 pixel patch, a 640×480 image requires a processing time of 5–15 min, depending on the number of objects present in the image. However, as yet, optimization of speed has not been attempted.

7. Conclusion

An extended ECOC algorithm for multiclass classification has been described. Unlike prior ECOC methods, which only assign class labels, this algorithm also calculates class probabilities for each sample. This extension in conjunction with a strategy developed in our previous studies not only facilitates assignment of class membership but also permits localization of identified objects relative to pixel coordinates. Our algorithm therefore makes possible

both subtyping and localization of unstained cells in bright field images of cell mixtures. Our extended ECOC strategy has been shown to be superior to several other currently existing approaches, especially for complex scenarios. The speed and accuracy of our multiclass cell detection framework suggest that it can be useful in some systems that require automatic subtyping and localization of cells in cell mixtures.

In this study, our goal has been focused on exploring the use of ECOC in a multiclass classification and localization system. The probability estimation ability we added to previous ECOC methods was solely used in combination of the pixel patch decomposition method to provide localization information. This extended algorithm retains the classification accuracy of pre-existing ECOC methods. In fact, our experiments suggest that there is only slight difference between the performance of our new algorithm and that of other ECOC-based algorithms (see Figs. 9 and 12).

Acknowledgment

This work is supported by NIH Grant CA89841.

References

[1] X. Long, W.L. Cleveland, Y.L. Yao, A new preprocessing approach for cell recognition, IEEE Transactions on Information Technology in Biomedicine 9 (3) (2005) 407–412.
[2] X. Long, W.L. Cleveland, Y.L. Yao, Automatic detection of unstained viable cells in bright field images using a support vector machine with an improved training procedure, Computers in Biology and Medicine 36 (2006) 339–362.
[3] D.J.M. Tax, R.P.W. Duin, Using two-class classifiers for multiclass classification, ICPR16, in: Proceedings of 16th International Conference on Pattern Recognition, Quebec City, Canada, 2002, pp. 124–127.
[4] G. Valentini, F. Masulli, Ensembles of Learning Machines Neural Nets WIRN Vietri-02Series Lecture Notes in Computer Sciences, Springer-Verlag, Heidelberg, Germany, 2002.
[5] V. Vapnik, Statistical Learning Theory, Wiley, 1998.
[6] V. Guruswami, Amit Sahai, Multiclass learning, boosting, and error-correcting codes, in: Proceedings of the Twelfth Annual Conference on Computational Learning Theory, Santa Cruz, CA, USA, 1999, pp. 145–155.
[7] L. Breiman, Bagging predictors, Machine Learning 26 (2) (1996) 123–140.
[8] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[9] E. Kong, T. Dietterich, Error-correcting output coding corrects bias and variance, in: Proceedings of the 12th International Conference on Machine Learning, 1995, pp. 313–321.
[10] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1995) 263–286.
[11] G. James, T. Hastie, The error coding method and PiCTs, Journal of Computational and Graphical Statistics 7 (3) (1997) 377–387.
[12] J. Kittler, Face verification using error correcting output codes, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR01), 2001, pp. 755–760
[13] A. Berger, Error-correcting output coding for text classification, IJCAI'99: Workshop on Machine Learning for Information Filtering, Stockholm, Sweeden, 1999.
[14] R. Ghani, Using error-correcting codes for text classification, in: Proceedings of ICML-00, 17th International Conference on Machine Learning, 2000, pp. 303–310.
[15] D. Aha, R. Bankert, Cloud classification using error-correcting output codes, Artificial Intelligence Applications: Natural Resources, Agriculture, and Environmental Science 11 (1) (1997) 13–28.
[16] G. Bakiri, T. Dietterich, Achieving high-accuracy text-to-speech with machine learningData Mining in Speech Synthesis, Kluwer Academic Publishers, Boston, MA, 1999.
[17] T. Hastie, R. Tibshirani, Classification by pairwise couplingAdvances in Neural Information Processing Systems, 10, MIT Press, 1998.
[18] C. Burges, A tutorial on Support Vector Machines for pattern recognition, Data Mining and Knowledge Discovery 2 (1998) 122–167.
[19] W.L. Cleveland, I. Wood, B.F. Erlanger, Routine large-scale production of monoclonal antibodies in a protein-free culture medium, Journal of Immunological Methods 56 (1983) 221–234.
[20] B.B. Mishell et al., Preparation of mouse cell suspensions, in: B.B. Mishell, S.M. Shiigi (Eds.), Selected Methods in Cellular Immunology, W.H. Freeman and Company, New York, 1980.
[21] <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
[22] T.W. Nattkemper, H. Ritter, W. Schubert, A neural classifier enabling high-throughput topological analysis of lymphocytes in tissue sections, IEEE Transactions on Information Technology in Biomedicine 5 (2) (2001) 138–149.
[23] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.
[24] T.-K. Huang, R.C. Weng, C.-J. Lin, A Generalized Bradley-Terry Model: From Group Competition to Individual Skill, <http://www.csie.ntu.edu.tw/~cjlin/papers/generalBT.pdf>, 2004.
[25] D.P. Chakraborty, Maximum likelihood analysis of free-response receiver operating characteristic (FROC) data, Medical Physics 16 (1989) 561–568.