ELSEVIER

Available online at www.sciencedirect.com
ScienceDirect

Pattern Recognition 41 (2008) 67–76

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/pr

# Data-driven decomposition for multi-class classification

Jie Zhou[a,*], Hanchuan Peng[b], Ching Y. Suen[c]

[a]Department of Computer Science, Northern Illinois University, DeKalb, IL 60115, USA
[b]Janelia Farm Research Campus, Howard Hughes Medical Institute, Ashburn, VA 20147, USA
[c]Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, Que., Canada H3G 1M8

Received 6 April 2006; received in revised form 24 May 2007; accepted 29 May 2007

**Abstract**

This paper presents a new study on a method of designing a multi-class classifier: Data-driven Error Correcting Output Coding (DECOC). DECOC is based on the principle of Error Correcting Output Coding (ECOC), which uses a code matrix to decompose a multi-class problem into multiple binary problems. ECOC for multi-class classification hinges on the design of the code matrix. We propose to explore the distribution of data classes and optimize both the composition and the number of base learners to design an effective and compact code matrix. Two real world applications are studied: (1) the holistic recognition (i.e., recognition without segmentation) of touching handwritten numeral pairs and (2) the classification of cancer tissue types based on microarray gene expression data. The results show that the proposed DECOC is able to deliver competitive accuracy compared with other ECOC methods, using parsimonious base learners than the pairwise coupling (one-vs-one) decomposition scheme. With a rejection scheme defined by a simple robustness measure, high reliabilities of around 98% are achieved in both applications.

*Keywords:* Multi-class classification; Error Correcting Output Coding (ECOC); Data-driven Error Correcting Output Coding (DECOC); Support vector machine; Handwritten numeral recognition; Gene expression classification

## 1. Introduction

Multi-class classifiers have wide and practical usages in pattern recognition for problems that involve several possible categories. Given a training sample vector $\mathbf{x}=\{x_1, \dots, x_d\}$, where $d$ is the dimension of the data sample, the task of a multi-class classifier is to assign it to one of the $K$ categories with $K \geqslant 3$. Examples of such applications include OCR problems that consist of various characters, numerals or combination of both; diagnosis of different diseases based on medical signals; and bioinformatics problem of classifying different tumor types based on gene expression information.

There are two ways of designing a multi-class classifier, one is to directly develop a multi-class algorithm; the other is to decompose a multi-class problem to multiple two-class problems. Recently, the decomposition scheme has gained a lot of attention [1–11]. The reason is twofold: First, binary classifiers are easier to implement; second, some powerful algorithms are inherently binary such as Support Vector Machine (SVM) [3]. In fact, many popular SVM libraries use the decomposition approach to solve multi-class problems including LIBSVM [1], HeroSVM [12], etc.

When a support vector machine is used as the binary base learner, there are two common decomposition methods: (1) one-vs-others, where each base learner separates one class from the remaining classes (simplified as 1vo in the rest of the paper); and (2) pairwise coupling, where each base learner separates one class from another class (also called one-vs-one) [1,2]. For the 1vo scheme, $K$ base learners are needed, while for pairwise coupling, as many as $K(K-1)/2$ base learners are needed. More generally, Dietterich and Bakiri [4] formulated the problem of multi-class classification as an issue of designing a decomposition matrix following the principle of Error Correcting Output Coding (ECOC). The basic idea of ECOC is to map every class to a bit-string called *codeword*. These codewords are the rows of the *code matrix*. Every column of the code

[*] Corresponding author. Tel.: +1 815 753 3815.
*E-mail address:* jzhou@cs.niu.edu (J. Zhou).

0031-3203/$30.00 © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2007.05.020

matrix corresponds to a binary base learner (see Section 2 for details of an ECOC code matrix). Because the decomposition is solely determined by the code matrix, ECOC is viewed as a framework for decomposing multi-class classification. While ECOC is flexible, an open issue to use ECOC is how to design the code matrix such that the effectiveness can be balanced against efficiency: while an exhaustive permutation can give a complete code matrix that includes all the possible patterns of matrix elements, the number of columns (i.e., the number of base learners) is exponential to the number of classes, which is too costly. For pattern recognition applications, the number of base learners is directly linked to the efficiency of testing when the algorithm is applied. We thus want to control the number of learners while maintaining the accuracy of the method.

Since then, many ways of designing the ECOC code matrix have been investigated, including coding based on algebraic code theory such as BCH codes [4], random codes [5], and coding based on optimization theory [6]. These methods examine the property of the code matrix itself (for example, the number of different bits between rows of the matrix) and do not involve the training data in designing the matrix. As an early work that utilizes training data for code matrix design, Alpaydin and Mayoraz [7] used the backpropagation algorithm to derive the codes of the matrix. The method is only applicable when the base learner is a multilayer perceptron. Moreover, since applications of these algorithms have been tested mostly on benchmarking data sets [7,8] (such as data sets from UCI Machine Learning repository [9]) or text classification [10,11], more studies on pattern recognition applications in various domains will be desirable for evaluating and advancing the decomposition-based approach for multi-class classification.

Motivated by providing new solutions to the problem of multi-class decomposition and extending the applications of ECOC, we propose a new data-driven decomposition approach in this paper. Different from current methods, it is a mechanism that adaptively designs the code matrix of ECOC based on the inherent structure of the training data. The method does not limit itself to any particular base learner. We name the proposed method DECOC. We then apply DECOC to two multi-class pattern recognition problems that have not yet been addressed by the ECOC approach so far. The first problem is the holistic recognition of touching handwritten numerals: the holistic approach that recognizes the whole image of touching numerals without segmenting them into individual numerals. It deals with more complex patterns than traditional OCR problems; the second problem is the bioinformatics problem of classifying tumor tissue types based on microarray gene expressions of the tissue cells. It has the issue of "high dimension low sample size" (referred as HDLSS problem in statistics [13,14]). Compared with other decomposition methods, the results from investigating DECOC on these two challenging problems suggest that DECOC is an effective decomposition approach for solving real world multi-class pattern recognition problems.

The rest of this paper is organized as follows: Section 2 gives the background of ECOC-based multi-class classification; Section 3 details the methodology of the proposed DECOC. In Section 4, we apply DECOC to two different domains and discuss the results; and Section 5 is the conclusion.

## 2. Related work: ECOC in multi-class classification

ECOC have been used in the fields of network communication and information theory for the purpose of enhancing the reliability of transmitting binary signals and maintaining information integrity [15,16]. It adds the redundant parity bits to an information word. The result is called a code word, which is a binary code string with 0's and 1's. Distances between two code words are described using Hamming distance, which is the count of the different bits in the two patterns. On the receiving end, a decoding process examines the Hamming distances between the received binary message and all the valid code words to detect errors and recover signals.

When ECOC was introduced to the design of multi-class classifiers [4], the coding process is a mapping from a set of classes to the set of code words: $\{C_1, \ldots, C_K\} \to \{w_1, \ldots, w_K\}$. The code words are rows of the coding matrix $M = [b_{k,i}]$, with $k$ as the index of classes (rows), $i$ as the index of base learners (columns).

Table 1 gives an illustrative example of the ECOC code matrix of a 5-class classification problem, decomposed into six binary classification problems.

The code matrix is used to guide the training and testing processes of ECOC classifiers. In training, six binary base learners are trained. For the binary base learner $f_i$ $(i = 1, \ldots, 6)$, if an element $b_{k,i}$ $(k = 1, \ldots, 5$ and $i = 1, \ldots, 6)$ in the code matrix is 1, then all samples of class $k$ will be considered positive. The remaining samples are considered negative for $f_i$ (can be labeled as $-1$ or 0. Here we use 0 for analogy of binary coding). The testing process determines the class label $y$ of a testing sample $\boldsymbol{x}$ by first applying all base learners to the unknown sample, yielding a codeword $w(\boldsymbol{x})$ (a bitstring of 1's and 0's). A decision on the label is then made based on the shortest Hamming distance:

$$y = \operatorname*{argmin}_k H(w_k, w(x)), \quad k = 1, \ldots, K,$$

where $w_k$ is the ideal code word for class $k$, i.e., the $k$th row of the code matrix. $H(w_k, w(x))$ is the decoding function which computes the Hamming distance between $w_k$ and $w(x)$. We assign the class label of the closest codeword, i.e., with the shortest Hamming distance, to the testing sample. In the case of the code matrix given in Table 1, the ideal code word for class 1 is [1 1 0 1 1 1], and the ideal code word for class 5 is [1 1 1 0 1 0]. If a testing sample yields a code word [1 1 0 1 1 1], it will be determined as class 1 which corresponds to the shortest Hamming distance.

In summary, the ECOC scheme is an ensemble-classification that uses the results of all base learners (binary classifiers) to determine the class label of a testing sample. The key is to design the code matrix so that training and testing of base learners can be conducted accordingly. In this framework, the pairwise scheme (i.e., one-vs-one) of multi-class classification will involve $K * (K - 1)/2$ base learners. The one-vs-others scheme

Table 1
Example of an ECOC code matrix

| Class | Base learner 1 | Base learner 2 | Base learner 3 | Base learner 4 | Base learner 5 | Base learner 6 |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 0 | 1 | 0 | 1 | 0 |
| 5 | 1 | 1 | 1 | 0 | 1 | 0 |

will involve $K$ base learners. One motivation of this study is to explore the distribution of data classes and optimize both the composition and the number of base learners to design an effective and compact code matrix to achieve improved accuracy and robustness.

## 3. Methodology: DECOC

### 3.1. Basic concepts

We propose DECOC to design the code matrix for ECOC by choosing the code words utilizing the intrinsic information from the training data. In a preset decomposition mechanism for a $K$-class problem such as pairwise coupling, $K*(K-1)/2$ base learners are always needed which can be a large number of base learners when $K$ gets big. The key idea of DECOC is to selectively include some of the binary learners into the code matrix based on a *confidence score* we define for a binary base learner. This measure will help us to determine how likely we will include a learner in the ensemble.

Before introducing the confidence score, we first define the *separability criterion* for a group of samples of multiple class labels, with $G$ as the set of the class labels of these samples:

$$S(G)=\begin{cases}\dfrac{2}{|G|^2-|G|}\displaystyle\sum_{j\neq k;c_j,c_k\in G} d(c_j,c_k) & |G|\neq 1 \quad \text{and} \quad |G|\neq K-1 \\ 0 & \text{otherwise}\end{cases} \qquad (1)$$

where $d(c_j, c_k)$ is the distance between two classes $c_j$ and $c_k$, which is the Euclidean distance of the mean or median vectors of the two classes; $|G|$ is the size of the set $G$, i.e., number of classes in the group of samples associated with classes of $G$; $2/(|G|^2-|G|)$ is the normalization factor. If there is only one class or there are $K-1$ classes in $G$, then $S(G)$ is set to 0 since both situations correspond to the 1vo partition of classes which is a special case to be considered separately. $S(G)$ also indirectly describes the inherent homogeneity of a group of samples: the smaller the $S(G)$ is, the more homogenous the group of samples is. Note that sample groups associated with $G$ are drawn from the training samples.

The confidence score of a binary base learner $f$ in DECOC is then defined as

$$C(f)=\begin{cases}\dfrac{S(G_{+/-}(f))}{S(G_+(f))+S(G_-(f))}, & |G_+|\neq 1 \text{ and } |G_+|\neq K-1, \\ 0 & \text{otherwise,}\end{cases} \qquad (2)$$

where $G_+(f)$ is the set of classes whose samples are considered as positive by the base learner $f$, $G_-(f)$ is the set of remaining classes whose samples are considered negative. For example, for the base learner 4 in Table 1, classes 1, 2 and 3 among five classes are considered as positive, so $G_+(f)=\{1,2,3\}$ and $S(G_+(f))=2/(3^2-3)^*\sum_{cj,ck\in\{1,2,3\}}\mathrm{d}(c_j,c_k)$.

$S(G_{+/-}(f))$ denotes the separability by viewing the data set as two groups of positive and negative samples separated by the base learner $f$. $S(G_{+/-}(f))$ equals the distance between the two groups: $S(G_{+/-}(f))=2/(2^2-2)^*d(c_+,c_-)=d(c_+,c_-)$, where $c_+$ represents the group of all the samples considered as positive by the base learner $f$, *and* $c_-$ represents the group of all the samples considered as negative by $f$.

Let $F$ denote the set of base learners selected by DECOC, we then define the objective of optimization as selecting the set of base learners $F$ that maximizes

$$O(F)=\frac{\sum_{i=1}^{N}C(f_i)}{\max(2N/(K^2-K),1)}, \quad f_i\subset F, \ i=1,\ldots,N, \qquad (3)$$

where $N$ is the total number of base learners used by DECOC; the denominator represents the penalty caused by a large number of base learners especially when it is larger than $(K^2-K)/2$ (it is the number of base learners used by the pairwise coupling scheme which serves as the baseline). This factor is to discourage too many base learners which can lower the efficiency when applying the DECOC classifier in the testing.

In addition, reducing testing errors is also essential for many real world applications. With a measurable robustness of the classification decision, reasonable rejections can be generated, and the reliability on prediction of the testing samples can be increased. Intuitively we can use the difference between two closest Hamming distances as the measure of robustness of the ECOC classifier [4]. Here we extend the idea and define the robustness coefficient of a decision of DECOC as follows:

$$RC=\frac{H(w_2,w(x))-H(w_y,w(x))}{N}, \qquad (4)$$

where $w_y$ is the closest row of the code matrix corresponding to label $y$ given by DECOC for sample $x$; $w_2$ is the row of the code matrix that is second closest to the code word of the testing

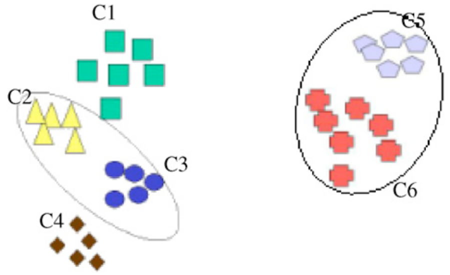

Fig. 1. A conceptual example that illustrates DECOC's rationale in defining $C(f)$: DECOC prefers the base learner (i.e., higher $C(f)$) that separates the group of classes C5 and C6 from other classes even though C2 and C3 may be closer in terms of inter-class distance, because the remaining classes for C2 and C3 are less homogeneous.

sample $x$. A robustness threshold can be set on $RC$ so that testing samples with $RC$ smaller than the threshold can be rejected. The threshold can be adjusted based on the application-specific tradeoff between recognition rate and error rate. For example, for applications with low tolerance on errors such as those in financial domain, the threshold is set higher and the error rate can be reduced at the cost of more rejections. A common choice of the threshold is around 0.1, so if the difference between the top two Hamming distances is less than 10% of the length of codeword, then the testing sample is rejected.

#### 3.1.1. *The rationale of DECOC and confidence score $C(f)$*

As stated by Eq. (3), we aim to maximize the sum of confidence scores of the learners involved in the ensemble classifier. The rationale is partially related to the observation of Allwein et al. [8], who pointed out that the average error of individual base learner contributes to the upper bound of ECOC's overall error. Schapire also mentioned in Ref. [17] (extension of Theorem 1) that a committee machine's generalization capability is related to the confidence of classification.

The motivation of increasing the overall confidence of the ECOC ensemble leads to DECOC's definition of confidence score of a base learner, and selecting the base learners that deal with highly separable groups and generate confident individual classification decisions. We also expect that, by including the most confident learners, comparably good accuracy can be achieved with less base learners than pairwise coupling.

Utilizing structural information in the training data enables the code matrix to be adaptive and dynamic instead of being preset regardless of data properties. The separability criterion defined in Eq. (1) uses the scheme that involves inter-cluster separation important for cluster validation [18,19].

In Eq. (2), we define confidence score $C(f)$ of a base learner $f$ based on the separability criterion of the class groups. $C(f)$ aims to find a base learner that maximizes the inherent homogeneity of both groups. The rationale of including the homogeneity of both class groups $G_+$ and $G_-$ in Eq. (2) is that, while the homogeneity of one group is an important determining factor, we also need to consider the homogeneity of the remaining classes. This is explained in Fig. 1.

### 3.2. *Algorithm implementation*

The optimization of Eq. (3) can be done by maximizing the numerator (i.e., the sum of confidence values of base learners) and minimizing the denominator (i.e., the number of learners). We explain our implementation in the following two subsections for these two goals, respectively.

#### 3.2.1. *Selecting base learners based on sorted confidence score—maximizing $\sum_{i=1}^{N} C(f_i)$*

If the numbers of different types of learners are determined, then the numerator of Eq. (3) can be maximized by maximizing the confidence scores of all selected learners.

We treat the 1vo learners specially by always including them in the DECOC code matrix since only $K$ 1vo base learners are required. These 1vo learners do not contribute to the maximization of Eq. (3) according to the definition of confidence score (Eq. (1)). So what we need to maximize now is $\sum_{i=1}^{N-K} C(f_i)$. We can achieve this by sorting all the $C(f_i)$ and choosing the top base learners with the highest confidences in different types of learners. To avoid an exponential number of possible partitions, we focus on three types of base learners in this study: 1vo, 2vo and 3vo, denoted by $f_{1\mathrm{vo}}$, $f_{2\mathrm{vo}}$, $f_{3\mathrm{vo}}$. Let the proportion of $f_{3\mathrm{vo}}$ in all base learners be denoted by $p_{3\mathrm{vo}}$, then the set of base learners $F$ to be included in DECOC is

$$F = \{f_{1\mathrm{vo}}^{i} | i = 1, \ldots, K\} \cup \{f_{2\mathrm{vo}}^{1}, \ldots, f_{2\mathrm{vo}}^{N-K-N*p_{3\mathrm{vo}}}\} \cup \{f_{3\mathrm{vo}}^{1}, \ldots, f_{3\mathrm{vo}}^{N*p_{3\mathrm{vo}}}\}$$

with

$$C(f_{2\mathrm{vo}}^{1}) > C(f_{2\mathrm{vo}}^{2}) > \cdots > C(f_{2\mathrm{vo}}^{N-K-N*p_{3\mathrm{vo}}}) \quad \text{and} \quad C(f_{3\mathrm{vo}}^{1}) > C(f_{3\mathrm{vo}}^{2}) > \cdots > C(f_{3\mathrm{vo}}^{N*p_{3\mathrm{vo}}}). \tag{5}$$

Fig. 2 describes the flow of calculating the confidence scores and selecting the base learners, which is the core of the DECOC algorithm. We can see that DECOC is a data-driven approach of designing code matrix: instead of having a preset matrix, DECOC adaptively generates the code matrix based on the structure of the given training data.

#### 3.2.2. *Determining $N$ and $p_{3\mathrm{vo}}$*

The sum of different types of learners is $N$. Since we always include $K$ 1vo learners, our task is to determine $N-K$ and $p_{3\mathrm{vo}}$. When $K<5$, $p_{3\mathrm{vo}}$ is 0 (no need for $f_{3\mathrm{vo}}$); otherwise, $p_{3\mathrm{vo}} \in [0, 1-K/N]$. Candidates of $p_{3\mathrm{vo}}$ come from the discrete set of $\{0.0, 0.1, \ldots, 1-K/N\}$. We determine $N-K$ and $p_{3\mathrm{vo}}$ using iterative search and cross-validation as follows:

*Step* 1: Initialize the integer $m$ (the variable for setting $N-K$) at a small number.

*Step* 2: Use cross-validation on the training set to find the best candidate $p_{3\mathrm{vo}}$ and its corresponding accuracy, following the algorithm of Fig. 2.

*Step* 3: Increment $m$ by 1 and go back to Step 2. Repeat until no improvement on accuracy is observed for several consecutive iterations.

The iterative search is based on the general rule from theoretical studies on many committee machines and experimen-

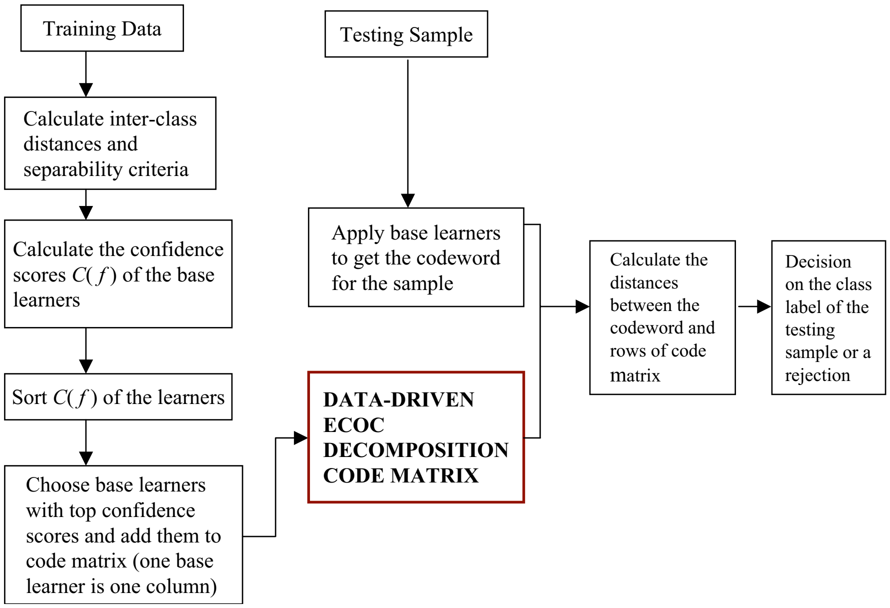
Training Data
Testing Sample
Calculate inter-class distances and separability criteria
Calculate the confidence scores C( f ) of the base learners
Apply base learners to get the codeword for the sample
Calculate the distances between the codeword and rows of code matrix
Decision on the class label of the testing sample or a rejection
Sort C( f ) of the learners
DATA-DRIVEN ECOC DECOMPOSITION CODE MATRIX
Choose base learners with top confidence scores and add them to code matrix (one base learner is one column)

Fig. 2. The flow of the training and testing algorithms for DECOC.

tal studies of ECOC [17,20–22]: the more base learners, the more accurate the classifier ensemble tends to be, and this tendency reduces when $N$ increases. Since Eq. (3) mainly punishes a larger $N > K(K-1)/2$, the search increment can be bigger for smaller $N$ to save training time. The trained learners in early iterations can also be stored for use in later iterations so that each iteration only needs to train a very small number of new learners. But in general, since training can be done offline, for many real world applications, we would care more about testing time, which is directly related to the application performance. In fact, as a phenomenon observed during early study [7,22] and experiments in this paper, it is found that data-driven decomposition typically needs less base learners than pairwise coupling.

### *3.2.3. A synthetic example to illustrate the DECOC algorithm*

Fig. 3 depicts the synthetic 2D data samples belonging to 11 different classes ($K = 11$). There are 550 training samples with 50 per class and 550 testing samples, also 50 per class. The samples are randomly generated from normal distribution.

During training, the confidence scores of all 2vo and 3vo base learners are calculated and sorted. Fig. 3 shows the decision boundary of the most confident 2vo base learner $f_{2\text{vo}}^1$, which separates classes C1 and C2 from the rest, while $f_{3\text{vo}}^1$ separates samples of C1, C2 and C3 from the rest. The length $N$ of code matrix of DECOC is determined based on methods described in the previous subsection, which set $N = 33$ and $P_{3\text{vo}} = 0.2$. So the matrix has 11 rows and 33 columns. The 33 selected base-learners columns include 11 1vo and 15 2v0 and 7 3vo learners with the highest ranking scores.

Comparatively, the pairwise coupling approach needs $K^*(K-1)/2$ learners, which equals 55, and has 67% more learners than those utilized by the DECOC scheme.

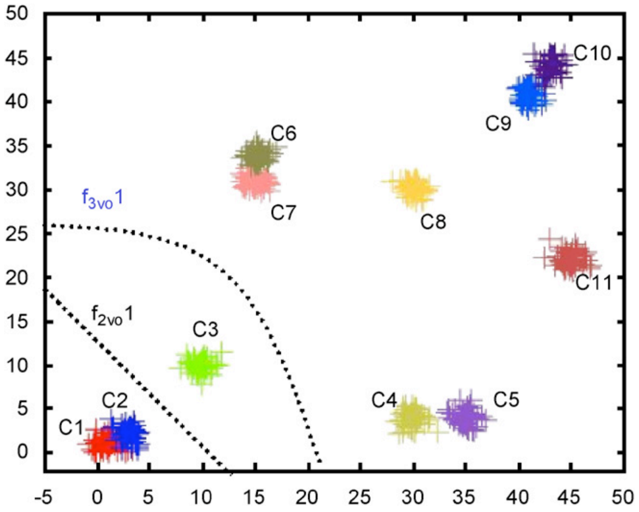

Fig. 3. A synthetic 2D example of 11 classes. Two dashed lines illustrate the decision boundaries of two base learners $f_{2\text{vo}}^1$ and $f_{3\text{vo}}^1$. DECOC uses 33 base learners to obtain the same performance achieved by pairwise coupling using 55 learners.

Experiments on the testing set by DECOC gives the accuracy of 97.82%, which is the same as pairwise coupling with 55 learners (both using LibSVM with default setting as base learner).

## 4. Applying DECOC to two pattern recognition problems

The DECOC algorithm is applied to two pattern recognition problems: recognition of touching handwritten numeral pairs and classification of cancer tissues based on microarray gene expressions. These two applications are not only new to DECOC, but also to ECOC in general. We investigate the

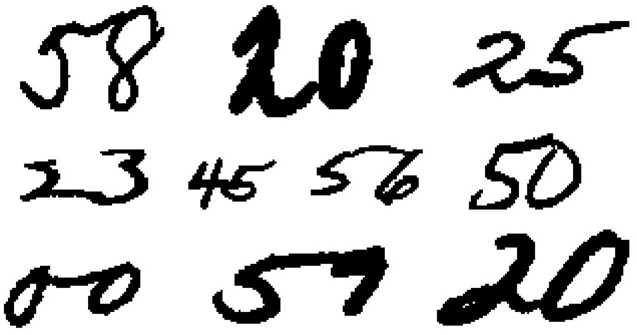

Fig. 4. Examples of touching numeral pair images.

Table 2
Data set of touching numeral pairs

| Class | Numeral pair | # of samples in training set | # of samples in testing set |
|---|---|---|---|
| 1 | 00 | 227 | 40 |
| 2 | 20 | 226 | 40 |
| 3 | 23 | 218 | 40 |
| 4 | 25 | 230 | 40 |
| 5 | 26 | 166 | 40 |
| 6 | 40 | 165 | 40 |
| 7 | 45 | 162 | 40 |
| 8 | 50 | 242 | 40 |
| 9 | 56 | 230 | 40 |
| 10 | 58 | 181 | 40 |
| 11 | 59 | 181 | 40 |
| 12 | 89 | 230 | 40 |
| Total | – | 2458 | 480 |

effectiveness of the decomposition-based method on these two problems and compare DECOC with other ECOC approaches.

### 4.1. Holistic recognition of touching numeral pairs

We recognize unconstrained handwritten numeral pairs using the holistic approach, i.e., recognizing the whole image without segmenting the numerals. So far, most studies on numeral strings/pairs recognition involve segmenting the string into separate numerals [23–25]. Segmentation of numeral strings is not trivial because typically the strings are written very freely and there exist various touching types and touching point locations. If the segmentation were wrong, even the best classifier would not help much. On the other hand, limited work has reported results of holistic recognition of numeral strings due to the increased complexity of the string images compared with the separated numerals [26,27]. In this study, we attempt the holistic approach for touching numeral pair recognition which has become feasible along with the increased power of computer and a well-designed classifier or classifier ensemble.

The experiments are conducted on touching numeral pairs from NIST 19 database. Images f0000–f4169 are used. We selected the top 12 most common pairs in the NIST 19, yielding a data set of 2938 samples in total, among which 2458 samples are used for training and 480 for testing. Fig. 4 gives some examples of the free style touching numerals. Table 2 presents the details of the data set.

In this study, we report the results with SVM as the base learner. LibSVM [28] with degree 3 polynomial kernel and default parameters is used for recognizing touching numeral pairs. (However, as we pointed out in previous sections, DECOC is a general decomposition approach applicable to other learners than SVM.) Mean vectors are used to calculate the separability criterion for sample groups. The parameter $p_{3\text{vo}}$ is set equal to 0.4.

One issue of holistically recognizing handwritten numerals is the increased requirement on handling computational complexity caused by the higher dimension of input image (400–13 K in original images, 1440 after normalization). To apply DECOC to numeral string recognition with practical efficiency, principal component analysis (PCA) is first conducted on the raw input images to extract a reduced number of variables. We use the first 43 principal components whose variance accounts for 65% of the total variance.

Fig. 5 gives an example of code matrix obtained by DECOC. Table 3 lists the results for the recognition of numeral pairs when no rejections are used. The results are compared with several common decomposition schemes. Other than the previously mentioned one-vs-others and pairwise coupling schemes, we also compare with a variation of pairwise coupling called DAGSVM [29]. It uses Directed Acyclic Graph to combine the binary base learners to allow for faster training. Table 3 shows that DECOC achieves the highest recognition rate for the application.

We then make use of the robustness measure defined in Eq. (4) to generate rejections and minimize the error rate of DECOC. The robustness threshold is set equal to 0.111. With this scheme, only 6 are mistaken out of 480 testing samples. A recognition rate of 80.4% and an error rate of as low as 1.2% have been achieved. With the reliability of the classifier defined as *Recognition Rate*/(*Recognition Rate*+*Error Rate*), DECOC achieves a reliability as high as 98.5%. Table 4 gives the confusion matrix. From the results, we observe that DECOC is capable of achieving promising results on recognizing complex numeral image patterns using the holistic approach.

### 4.2. Classification of tissue types based on gene expression

The second application of DECOC is the classification of cancer tissue types using microarray gene expressions. It is an important technique in diagnosis of cancer tissues [30], and a challenging research topic in the domain is how to build effective classifiers to separate multiple tissue types.

In this paper, we apply DECOC multi-class classifier to two microarray gene expression data sets: one is the NCI cancer cell-lines containing 60 samples for nine different cancers [31]. Each cancer type corresponds to 2–9 samples. The dimension of each sample is 9703; the other is the Lung cancer data set with

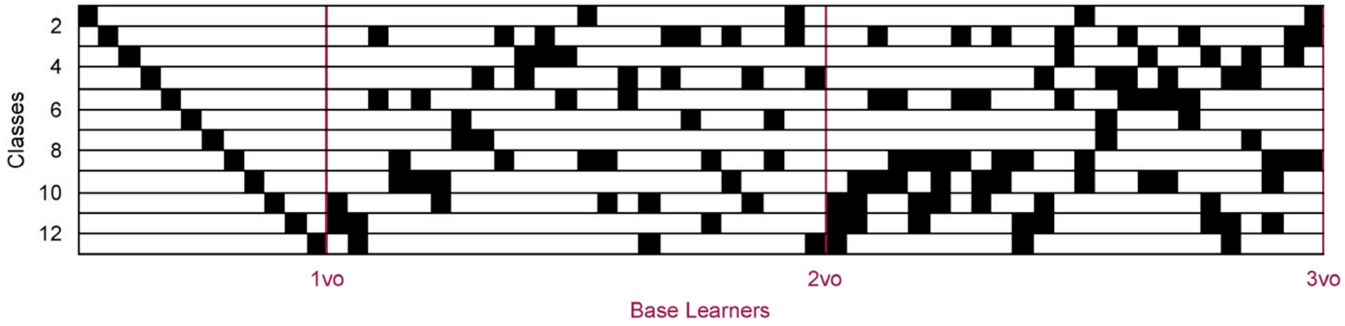

Fig. 5. A DECOC code matrix for the recognition of numeral pairs. The black box represents 1 and the white background represents 0. The rows correspond to 12 classes while the columns correspond to 60 base learners. This code matrix has 12, 1vo learners, 24, 2vo learners and 24, 3vo learners. For example, the first 2vo learner is selected to separate the two classes [class 10, class 11] from other classes. The last column of the matrix is a 3vo learner that separates classes 1, 2 and 8 from the remaining classes.

Table 3
Experimental results for handwritten numeral pairs *without rejections*

| Method | One-vs-others | DAGSVM | Pairwise coupling | DECOC |
|---|---|---|---|---|
| Recognition rate | 83.8% | 91.0% | 93.7% | 94.1% |

Table 4
Confusion matrix of touching numeral pair recognition by DECOC

| Class | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 33 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 1 | 0 | 0 | 33 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 27 | 1 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 36 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 29 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 33 | 2 |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 36 |

The columns are the real classes, while the rows are the predicted classes. For example, the element at column *i* and row *j* is the number of samples of class *i* recognized as class *j*. Robustness measure is used to generate rejections. Out of 480 testing samples, 386 are correct, 6 are errors, the remaining samples are rejected due to poor robustness, which yields a reliability of 98.5%.

Table 5
Gene expression data sets for different tissue types

| Class | NCI data set | | LUNG data set | |
|---|---|---|---|---|
| | Class name | # of samples | Class name | # of samples |
| 1 | NSCLC | 9 | Normal | 6 |
| 2 | Renal | 9 | AC-group-1 | 21 |
| 3 | Breast | 8 | AC-group-2 | 7 |
| 4 | Melanoma | 8 | AC-group-3 | 13 |
| 5 | Colon | 7 | Squamous | 16 |
| 6 | Leukemia | 6 | Small-cell | 5 |
| 7 | Ovarian | 6 | Large-cell | 5 |
| 8 | CNS | 5 | – | – |
| 9 | Prostate | 2 | – | – |
| Total # of samples | | 60 | | 73 |
| Dimensionality | | 9703 | | 918 |

Table 6
Average number of errors for tissue type classification on 14 different sizes of feature sets (3, 6, . . . , 54 genes as shown along $x$-axis of Fig. 6).

| Data set/method | One-vs-others | DAGSVM | Pairwise coupling | DECOC |
|---|---|---|---|---|
| NCI | 22.0 | 18.1 | 16.9 | 14.7 |
| LUNG | 16.6 | 11.6 | 11.3 | 8.3 |

different subtypes of lung cancer cells [32]. Besides normal cells, there are six subtypes of lung cancers. In total, there are seven classes with 73 samples. The dimension of each sample (i.e., number of genes) is 918. Details of the two data sets are listed in Table 5 .

As can be seen from Table 5, originally these data sets have contained expression levels of thousands of genes, i.e., the dimension of the data is very high. So we first applied a feature selection method, minimum Redundancy Maximum Relevance (mRMR), to reduce the dimensionality of the data [33]. Further, since the data samples are of small size, we use the standard Leave-One-Out Cross Validation (LOOCV) method to assess the accuracy of classifiers. LibSVM with linear kernel and default parameter is used as the base learner. Median vectors are used to calculate the separability criterion for sample groups.

We compare DECOC with three common decomposition methods: one-vs-others, DAGSVM, and pairwise coupling. Fig. 6 lists the experimental results in number of errors with respect to different number of features (genes) used by the classifier. For different numbers of features, most $p_{3\text{vo}}$ are set equal to 0.3, the rest are smaller. This observation indicates that less 3vo learners are needed for the tissue type classification problem compared with recognizing numeral pairs. Table 6 lists the average number of errors on the two gene expression data sets achieved by four decomposition-based

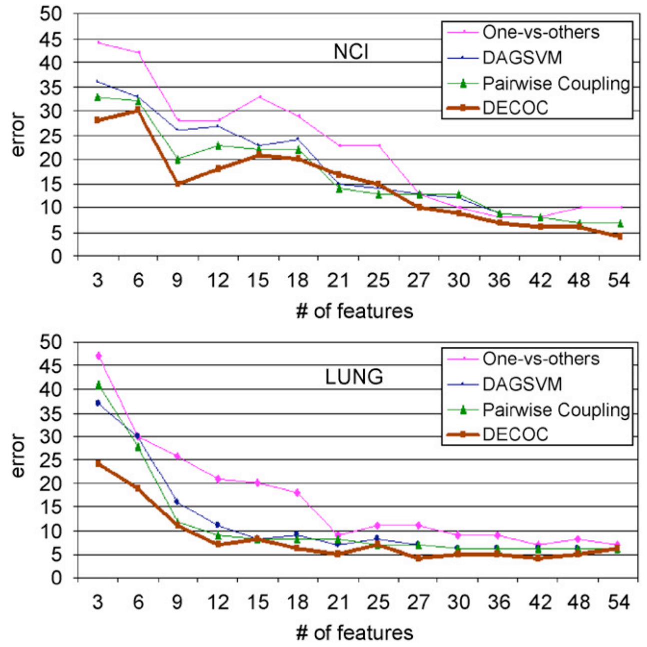

Fig. 6. Number of errors versus number of selected features of NCI and LUNG data sets for tissue type classification based on microarray gene expression.

classifiers. Both Table 6 and Fig. 6 show that DECOC delivers better results than the other three methods: while Table 6 indicates that DECOC has the lowest average errors among four methods, Fig. 6 shows that the conclusion is consistently true on almost all selected feature sets.

Further, for SVM, the lowest errors achieved by DECOC are among the best accuracies reported in the literature on the same gene expression data sets using the same type of classifier: for the NCI data set, the accuracy we obtained is 93.3%, and for the LUNG data set, we obtained the accuracy of 94.5%. Both results are comparable or better than the respective SVM results reported in Ref. [34]. When rejection is introduced, DECOC can further reduce the error to 1 out of 60 on NCI data set and 1 out of 73 on LUNG data set, which means we can increase the reliability to 97.9% and 98.5% on the two data sets, respectively.

### *4.3. Discussions*

#### *4.3.1. Number of learners used by DECOC*

Pairwise coupling needs 66 learners for the 12-class problem of numeral pair recognition; 36 learners for 9-class NCI data set, and 21 learners for the 7-class LUNG data set. Comparatively, the results of DECOC are achieved by a smaller number of learners. For recognizing numeral pairs, 60 learners are used. DECOC uses an average of only 25 learners on the NCI data set and 19 base learners on the LUNG data set.

#### *4.3.2. Effectiveness of the proposed method compared with other decomposition methods*

From the results presented in the previous section, we can see that DECOC delivers competitive performances in both applications compared with other decomposition schemes including pairwise coupling, one-vs-other and DAGSVM. Pairwise coupling delivers the second best performance. We also see that with a simple robustness measure, DECOC is capable of generating rejections and increasing the reliability to around 98% in both applications. Compared with the popular decomposition scheme of pairwise coupling, the DECOC method delivers a high reliability while typically utilizing a smaller number of learners. We attribute the efficiency and effectiveness of DECOC to the incorporation of more information in the design of ECOC code matrix.

During the experiment, we also tried adding one-vs-one classifiers to distinguish the two classes selected by two-vs-other learners. However, it does not always have an impact on accuracy. For example, no such pairwise classifier is needed to achieve the reliability of 98.5% for handwritten touching numeral recognition. While theoretical reason is to be further revealed, one possible explanation is that, without additional features that can distinguish the confusing class pairs and serve for verification purpose, simply lengthening the code word by adding base learners with high possibility of errors may not improve the overall decision accuracy of the ensemble. In future studies, more experiments will be conducted in this regard.

#### *4.3.3. Future look*

Our future plan includes experiments of DECOC based on other base learners such as Linear Discriminant Analysis (LDA) and decision tree. The impact of including different base learners (which may use different features) in the same classifier ensemble on improving the reliability of the pattern recognition applications is another interesting topic.

DECOC promotes the parsimony of ECOC in terms of number of base learners, which is desirable for applications. In this study we focus on three types of learners, but Eq. (3) is applicable to any p-versus-q learners in general. We expect that a larger $K$ would need more types of learners, which also increases computational complexity during training. We have pointed out that, for high dimensional inputs such as the gene expression data and touching numeral images, feature selection and extraction are necessary to alleviate the demand. Studies on the practical issues of controlling the costs of training, storing and evaluating of the ECOC base learners have been limited overall. Along with the increasing real world pattern recognition applications of ECOC, these issues will gain more attention and be further studied.

## 5. Conclusions

A flexible and systematic way of designing DECOC scheme has been described in this paper. We define confidence score of each base learner based on the structural information of the training data and use sorted confidence scores to assist the determination of code matrix of ECOC. Applications on two real world problems (three data sets) are reported with comparison to other common decomposition methods. It has been observed that DECOC is able to deliver competitive accuracy compared with other methods. Further, it uses an equal or smaller number of base learners than that of the popular pairwise coupling scheme. The results show that DECOC can be a promising decomposition-based multi-class classifier with good efficiency and effectiveness for real world applications.

## References

[1] T.-F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975 –1005.

[2] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Stat. 26 (2) (1998) 451–471.

[3] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1996.

[4] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–286.

[5] A. Berger, Error-correcting output coding for text classification, in: Proceedings of IJCAI'99: Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, 1999.

[6] K. Crammer, Y. Singer, Improved output coding for classification using continuous relaxation, in: Proceedings of NIPS, 2000.

[7] E. Alpaydin, E. Mayoraz, Learning error-correcting output codes from data, in: Proceedings of International Conference on Artificial Neural Networks (ICANN'99), vol. 2, 1999, pp. 743–748.

[8] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.

[9] C.L. Blake, C.J. Merz, UCI repository of machine learningdatabases, a huge collection of artificial and real-world data sets, 1998, Available from: ⟨http://www.ics.uci.edu/∼mlearn/MLRepository.html⟩.

[10] R. Ghani, Using error-correcting codes for text classification, in: Proceedings of ICML, 2000, pp. 303–310.

[11] J.D.M. Rennie, R. Rifkin, Improving multiclass text classification with the support vector machine, (MIT AI Memo 2001-026).

[12] J.X. Dong, A. Krzyzak, C.Y. Suen, A fast SVM training algorithm, in: S.-W. Lee, A. Verri (Eds.), International Workshop on Pattern Recognition with Support Vector Machines, Lecture Notes in Computer Science, vol. 2388, Springer, Niagara Falls, Canada, August 10, 2002, pp. 53–67.

[13] J.S. Marron, M. Todd, Distance weighted discrimination, Technical Report No. 1339, School of Operations Research and Industrial Engineering, Cornell University, July 2002.

[14] S.J. Raudy, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. PAMI 13 (3) (1991) 252–264.

[15] A. Viterbi, J. Omura, Principles of Digital Communication and Coding, Mc-Graw Hill, New York, 1979.

[16] D. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, UK, 2003.

[17] R.E. Schapire, Using output codes to boost multiclass learning problems, in: Proceedings of 14th International Conference on Machine Learning, 1997.

[18] J.C. Bezdek, N.R. Pal, Some new indexes of cluster validity, IEEE Trans. Syst. Man Cybernet. B 28 (3) (1998) 301–315.

[19] A.C. Rencher, Methods of Multivariate Analysis, second ed., Wiley, New York, 2002.

[20] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2) (2002) 281–286.

[21] R.E. Schapire, Y. Freund, P. Bartlett, W.S. Lee, Boosting the margin: a new explanation for the effectiveness of voting methods, Ann. Stat. 26 (1998) 1651–1686.

[22] J. Zhou, C.Y. Suen, Unconstrained numeral pair recognition using enhanced error correcting output coding: a holistic approach, in: Proceedings of International Conference on Document Analysis and Recognition (ICDAR'2005), 2005, pp. 484–488.

[23] C.L. Liu, H. Sako, H. Fujisawa, Effects of classifier structure and training regimes on integrated segmentation and recognition of handwritten numeral strings, IEEE Trans. PAMI 26 (2004) 1395–1407.

[24] J. Zhou, A. Krzyzak, C.Y. Suen, Verification—a method of enhancing the recognizers of isolated and touching handwritten numerals, Pattern Recognition 35 (2002) 1179–1189.

[25] L.S. Oliveira, R. Sabourin, F. Bortolozzi, C.Y. Suen, Automatic recognition of handwritten numerical strings: a recognition and verification strategy, IEEE Trans. PAMI 4 (11) (2002) 1438–1454.

[26] X. Wang, V. Govindaraju, S. Srihari, Multi-experts for touching digit string recognition, in: Proceedings of ICDAR, 1999, pp. 800–803.

[27] S.-M. Choi, I.-S. Oh, A segmentation-free recognition of handwritten touching numeral paris using modular neural network, Int. J. Pattern Recognition Artif. Intell. 15 (2001) 949–966.

[28] C.-W. Hsu, C.-J. Lin, A comparison of methods for multi-class support vector machines, IEEE Trans. Neural Networks 13 (2002) 415–425.

[29] J. Platt, N. Cristianini, J. Shawe-Taylor, Large margin DAGs for multiclass classification, Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, 2000, pp. 547–553.

[30] T. Speed, Statistical Analysis of Gene Expression Microarray Data, Chapman & Hall, London Press, CRC, Boca Raton, FL, 2003.

[31] D.T. Ross, U. Sherf, et al., A cDNA microarray gene expression database for the molecular pharmacology of cancer, Nat. Genet. 24 (3) (2000) 236–244.

[32] M. Garber, O. Troyanskaya, et al., Diversity of gene expression in adenocarcinoma of the lung, PNAS USA 98 (24) (2001) 13784–13789.

[33] H.C. Peng, F.H. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency max-relevance and min-redundancy, IEEE Trans. PAMI 27 (8) (2005) 1226–1238.

[34] C. Ding, H.C. Peng, Minimum redundancy feature selection from microarray gene expression data, J. Bioinformatics Comput. Biol. 3 (2) (2005) 185–205.

**About the Author**—JIE ZHOU received her B.S. and M.S. degrees in Biomedical Engineering from Southeast University, Nanjing, China in 1993 and 1996, respectively, and her Ph.D. degree in Computer Science from Concordia University, Montreal, Canada in 2000. Since 2002, she has been an Assistant Professor in the Department of Computer Science at Northern Illinois University, USA. Her research interests include pattern recognition, machine learning, intelligent data analysis and bioinformatics.

**About the Author**—HANCHUAN PENG is a research scientist with Howard Hughes Medical Institute, Virginia, USA. He was with the Lawrence Berkeley National Laboratory and Johns Hopkins University Medical School during 2000–2005. He earned a Ph.D. degree in Biomedical Engineering from Southeast University, China, in 1999. His research interests include image data mining, bioinformatics and medical informatics, pattern recognition and computer vision, signal processing, artificial intelligence and machine learning, biocomputing.

**About the Author**—CHING Y. SUEN received an M.Sc.(Eng.) degree from the University of Hong Kong and a Ph.D. degree from the University of British Columbia, Canada. In 1972, he joined the Department of Computer Science at Concordia University where he became Professor in 1979 and served as Chairman from 1980 to 1984, and as Associate Dean for Research of the Faculty of Engineering and Computer Science from 1993 to 1997. He has guided/hosted 70 visiting scientists and professors, and supervised 65 doctoral and master's graduates. Currently he holds the distinguished Concordia Research Chair in Artificial Intelligence and Pattern Recognition, and is the Director of CENPARMI, the Centre for PR & MI.

Prof. Suen is the author/editor of 11 books and more than 400 papers on subjects ranging from computer vision and handwriting recognition, to expert systems and computational linguistics. A Google search of "Ching Y. Suen" will show some of his publications. He is the founder of "The International Journal of Computer Processing of Oriental Languages" and served as its first Editor-in-Chief for 10 years. Presently he is the Deputy Editor of Pattern Recognition, a member of the Advisory Board of Pattern Recognition Letters, and an Associate Editor of the International Journal of Pattern Recognition and Artificial

Intelligence, Signal Processing, Expert Systems with Applications, and the International Journal of Document Analysis and Recognition. He was also an Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence and Pattern Analysis and Applications.

A Fellow of the IEEE, IAPR, and the Academy of Sciences of the Royal Society of Canada, he has served several professional societies as President, Vice-President, or Governor. He is also the Founder and chair of several conference series including ICDAR, IWFHR, and VI. He was the General Chair of numerous international conferences, including the International Conference on Computer Processing of Chinese and Oriental Languages held in August 1988 in Toronto, International Workshop on Frontiers in Handwriting Recognition in April 1990 in Montreal, International Conference on Document Analysis and Recognition held in Montreal in August 1995, and the International Conference on Pattern Recognition, held in Quebec City in August 2002.

Dr. Suen has given 150 seminars at major computer industries and various government and academic institutions around the world. He has been the principal investigator of 25 industrial/government research contracts, and has received many research grants from national and provincial funding agencies. He is a recipient of prestigious awards, including the ITAC/NSERC National Award from the Information Technology Association of Canada and the Natural Sciences and Engineering Research Council of Canada in 1992, the Concordia “Research Fellow” award in 1998, and the IAPR ICDAR award in 2005.