Pattern Anal Applic
DOI 10.1007/s10044-007-0094-6

THEORETICAL ADVANCES

Ensemble of HMM classifiers based on the clustering validity index for a handwritten numeral recognizer

Albert Hung-Ren Ko · Robert Sabourin · Alceu de Souza Britto Jr.

Received: 9 June 2006 / Accepted: 5 October 2007

Abstract A new scheme for the optimization of codebook sizes for Hidden Markov Models (HMMs) and the generation of HMM ensembles is proposed in this paper. In a discrete HMM, the vector quantization procedure and the generated codebook are associated with performance degradation. By using a selected clustering validity index, we show that the optimization of HMM codebook size can be selected without training HMM classifiers. Moreover, the proposed scheme yields multiple optimized HMM classifiers, and each individual HMM is based on a different codebook size. By using these to construct an ensemble of HMM classifiers, this scheme can compensate for the degradation of a discrete HMM.

Keywords Hidden Markov Models · Ensemble of classifiers · Codebook size · Clustering validity index · Pattern recognition

1 Introduction

The purpose of pattern recognition systems is to achieve the best possible classification performance. A number of classifiers are tested in these systems, and the most appropriate one is chosen for the problem at hand. Different classifiers usually make different errors on different samples, which means that, by combining classifiers, we can arrive at an ensemble of classifiers which will make more accurate decisions. In order for the ensemble to contain classifiers which make different errors, it is advisable that the classifiers be diverse. Once these classifiers have been selected, they are grouped together into what is known as an ensemble of classifiers (EoC).

There are several methods for creating diverse classifiers, among them random subspace, bagging and boosting [1, 8, 12, 25, 32, 33, 42]. The random subspace method does this by using different subsets of features to train the classifiers. Because problems are represented in different subspaces, different classifiers develop different borders for the classification. The bagging method randomly selects subsets of samples to train the classifiers. (Intuitively, based on different sample subsets, classifiers would exhibit different behaviors.) The boosting method also uses parts of samples to train classifiers, but does not do so randomly; difficult samples have a greater probability of being selected, and easier samples have less chance of being used for training. With this mechanism, most of the classifiers created will focus on hard samples, which may make them more effective.

Random subspace, bagging and boosting are general ensemble creation methods, and they can in most cases be applied to all kinds of classification algorithms to generate diverse classifiers for ensembles. However, there are some classification algorithms that might need to use all samples and all features for training, and thus cannot use random subspace, bagging or boosting for ensemble creation. Fortunately, there are some specialized ensemble creation methods which can be applied to these target classification

A. H.-R. Ko (✉) · R. Sabourin
LIVIA, École de Technologie Supérieure, University of Quebec,
1100 Notre-Dame West Street, Montreal,
Quebec H3C 1K3, Canada
e-mail: albert@livia.etsmtl.ca

R. Sabourin
e-mail: robert.sabourin@etsmtl.ca

A. de SouzaBritto Jr.
PPGIA, Pontifical Catholic University of Parana, Rua Imaculada Conceicao, 1155, Curitiba PR 80215-901, Brazil
e-mail: alceu@ppgia.pucpr.br

algorithms. To be successful, these specialized ensemble creation methods must take into account the training process of the target classification algorithm, so that the classifiers created will be diverse enough to construct an ensemble.

One of such classification algorithm is the Hidden Markov Model (HMM). An HMM is one of the most popular classification methods for pattern sequence recognition, especially for speech recognition and handwritten pattern recognition problems [4, 9, 10, 39, 40, 45]. The objective of the HMM is to model a series of observable signals, and it is this ability that makes the HMM a better choice for recognition problems than other classification methods. As a stochastic process, HMM is constructed with a finite number of states and a set of transition functions between two states or over the same state [4, 39, 45]. Each state transmits some observations, according to a codebook which sets out corresponding emission probabilities. Such observations may be either discrete symbols or continuous signals. In a discrete HMM, a vector-quantization codebook is typically used to map the continuous input feature vector to the code word.

To perform vector-quantization to generate the codebook of an HMM, we first need to define the size of the codebook. An HMM codebook size optimization is, in general, performed by constructing a number of HMM classifiers and comparing their recognition rates on a validation data set. In other words, the process of codebook size optimization is always problem-dependent. Moreover, given the extremely time-consuming process of HMM training, HMM codebook size optimization remains a major problem.

There are various methods for solving the HMM codebook size optimization problem, the difficulty being to define the "optimal" codebook. On the one hand, according to the "no-free-lunch" theory [47, 48], no search algorithm is capable of always dominating all others on all possible datasets. On the other hand, an optimal codebook is only optimal relative to a few other evaluated codebooks. For these reasons, we believe that it is in our interest to consider multiple optimal codebooks and to use them to construct an ensemble of HMM classifiers (EoHMM), rather than to select a single, supposedly optimal, codebook.

We note that the use of EoHMM has been emerging as a promising scheme for improving HMM performance [2, 15–20]. This is because an EoC is known to be capable of performing better than its best single classifier [7, 31, 32, 41]. EoC classifiers can be generated by changing the training set, the input features or the parameters and architecture of the base classifiers [20]. There are quite a few methods for creating HMM classifiers, based on the choice of features [18] for isolated handwritten images, and both column- and row HMM classifiers can be applied to enhance performance [5, 6]. The use of various topologies, such as left–right HMM, semi-jump-in, semi-jump-out HMM [19], and circular HMM [2] can also be applied.

In our case, we want to create an EoHMM based on several codebooks. To do this, all the codebooks must be good and diverse, i.e. the symbols (codewords) that these codebooks present must be useful and different. The reason for this is quite simple: in order to obtain different and accurate HMM classifiers, we should avoid those that are identical or under-performing. The main question is, how can we select good and diverse codebook sizes for an EoHMM? In terms of a good size for a codebook, we note that discrete symbols in HMM are usually characterized as quantized vectors in the codebook by clustering, so the fitness of the codebook is directly related to the fitness of the clustering, for which a number of validity indices have been proposed [3, 23, 24, 34, 37]. This means that codebook size can actually be optimized by using clustering validity indices.

Nevertheless, in order for codebook sizes to be diversed, the clustering validity indices used must offer several adequate codebook sizes, and not just only a single optimal one. Because a data set usually consists of multiple levels of granularity [11, 27, 43], if clustering validity indices can give multiple adequate codebook sizes for HMM, and if these HMM classifiers have diverse outputs, then it is possible to construct EoHMMs based on different codebook sizes. This mechanism will give the local optima of a selected clustering validity index. EoHMMs are then selected by various objective functions and combined by different fusion functions. Since EoHMMs are constructed with multiple codebooks, the degradation associated with a single vector quantization procedure can be improved by multiple vector quantization procedures, and by then classifier combination methods.

To clarify, we want to verify two assumptions in this work. Our first assumption is that a clustering validity index might have the property of being able to generate several codebook hypotheses. The second assumption is that the codebook hypotheses generated by one clustering validity index will contain enough diversity to construct a useful ensemble of EoHMMs. In this case, an EoHMM is constructed not based on different feature subspaces or on different samples, but on different representations in several symbol spaces. The key questions that need to be addressed are the following:

1. What are the basic properties of the clustering validity indices used in clustering?
2. Which clustering validity index performs better in the selection of codebook sizes for HMM?
3. Can the clustering validity index offer more than one hypothesis on HMM codebook sizes?

4 For HMM classifiers based on different codebook sizes selected by a clustering validity index, is the diversity among them strong enough to yield an EoHMM which performs well?

To answer these questions, we carried out a general review on clustering validity indices, and applied the selected index for EoHMM construction. We used the HMM-based handwritten numeral recognizer in [5, 6], which includes the numeral-string segmentation stage and the single-character verification stage. In this paper, we focus on improving the verification stage to recognize the separated handwritten digits. At this stage, column and row HMM classifiers are used to enhance classification accuracy, and the sum of the outputs from the single best column HMM and the single best row HMM constitutes the final decision. With this system, we were able to improve verification by constructing an EoHMM with different codebooks on both column- and row-HMM classifiers, and then carrying out ensemble selection and classifier combination. It is important to note that HMM optimization is a very complex task, and there are still a great many issues associated with it. The analysis and the method presented therefore constitute only a small step towards a considerably improved understanding of HMM and EoHMM.

The paper is organized as follows. In the next section, we introduce the basic concepts of clustering validity indices. Section 3 details the process of generation, selection and combination of HMM classifiers. In Sect. 4, we report on experiments we carried out on the NIST SD19 handwritten numeral database. A discussion and a conclusion are presented in the final sections.

2 Clustering validity indices

In general, an HMM codebook is generated from a vector quantization procedure, and each code word can be actually regarded as a centroid of a cluster in feature space. The fitness of the clustering depends on a number of different factors, such as clustering methods and the number of clusters. For an adequate HMM codebook, there should be a means to select a better clustering. A clustering validity index is thus designed to evaluate the clustering results, and to assign a level of fitness to these results. Three types of clustering validity indices have been proposed in the literature, including external indices, internal indices and relative indices [23, 37]. External indices are designed to test whether or not a data set is randomly structured; internal indices are used to evaluate the clustering results by comparing them with a known partition; and relative indices are designed merely to find the best clustering results, that is, the most natural ones, regardless of sample labels. Given the fact that we have no known partition for a codebook and we are interested in finding natural clusters as code words for HMM, we focus on the known relative indices in this section, present their definitions and discuss their advantages and drawbacks in evaluating clustering. We must mention that a clustering validity index is not a clustering algorithm in and of itself, but a measure to evaluate the results of clustering algorithms and give an indication of a partitioning that best fits a data set. A clustering validity index is independent of clustering algorithms and data sets.

2.1 *R*-squared (RS) index

To explain RS index, we need to explain the *Sum of squares* (SS) measure used in this index. We have three kinds of SS:

1. SS_w: The sum of squares within the cluster. Given a cluster c_x consisting of n samples, with the members $X_1, \ldots, X_n$, and the cluster center $\bar{X}$, define

$$SS_w(x) = \sum_{j=1}^{n} (X_j - \bar{X})^2 \tag{1}$$

and for nc clusters, suppose there are n_i samples for cluster c_i, and denote X_i as the centroid of the cluster c_i, and its members as X_{ij}, the total SS_w can be written as

$$SS_w = \sum_{i=1}^{nc} \sum_{j=1}^{n_i} (X_{ij} - \bar{X}_i)^2 \tag{2}$$

2. SS_b: The sum of squares between clusters Given a data set d_x of nc cluster centroids $c_1, \ldots, c_{nc}$, and the center of all the data $\bar{C}$, define

$$SS_b = \sum_{i=1}^{nc} (c_i - \bar{C})^2 \tag{3}$$

3. SS_t: The total sum of squares

$$SS_t = SS_w + SS_b \tag{4}$$

and RS [23, 24] is defined as the ratio of SS_b to SS_t. That is,

$$RS = \frac{SS_b}{SS_t} \tag{5}$$

Note that SS_b is a measure of difference between clusters, so that the more separated the two clusters, the greater SS_b will be. Moreover, SS_w is the compact measure of a single cluster. The smaller SS_w, the more compact this cluster will be. Given the same SS_w, RS is proportional to SS_b, and is the measure of distance between clusters. We can also write

$$\mathrm{RS} = \frac{\mathrm{SS_t} - \mathrm{SS_w}}{\mathrm{SS_w}} \tag{6}$$

Given the same $\mathrm{SS_b}$, RS can be regarded as a measure of compactness. To combine both effects, RS is a measure of homogeneity between clusters. The value of RS always being between 0 and 1. The process involves drawing the curve of RS while applying different numbers of clusters, and finding its "knee".

Given a number of clusters nc, a single RS takes into account the compactness of all clusters, as well as the distance between them. However, a single RS is unable to indicate how good the clustering is, but a series of RS indices can. We expect to see a huge increase in RS value when the best number of clusters nc_{best} is applied.

2.2 Root-mean-square standard deviation (RMSSTD) index

RMSSTD index is a measure based on sample variances and sample means. Supposing we have nc clusters in the data, and cluster c_i has n_i samples, $1 \leq i \leq nc$, then the mean of the cluster c_i is defined as

$$\bar{X}_i = \frac{1}{n_i}\sum_{j=1}^{n_i} X_j \tag{7}$$

where X_j, $1 \leq j \leq n_i$, are samples of cluster c_i. Moreover, the variance of cluster c_i is defined as:

$$S_i^2 = \frac{1}{n_i - 1}\sum_{j=1}^{n_i}(X_j - \bar{X}_i) \tag{8}$$

Similarly, RMSSTD [23, 24] is defined as

$$\mathrm{RMSSTD} = \left(\frac{\sum_{i=1}^{nc}\sum_{j=1}^{n_i}(X_{ij} - \bar{X}_i)}{\sum_{i=1}^{nc} n_i - 1}\right)^{\frac{1}{2}} \tag{9}$$

where n_i, $1 \leq i \leq nc$ is the number of samples of cluster c_i, and $\bar{X}_i$ is the centroid of cluster c_i, X_{ij}, $1 \leq j \leq n_i$ is a sample belonging to cluster c_i. From this, it is clear that RMSSTD decreases when the number of clusters increases, because the more clusters it has, the smaller the variance will be for each cluster.

Like RS, the best clustering can be located on the "knee" of RMSSTD curve.

2.3 Dunn's index

Assuming that the clustering process generates nc clusters, and that, for all clusters $c_1, \ldots, c_{nc}$, we define the dissimilarity of two clusters c_i, c_j, where $1 \leq i, j \leq nc$, $i \neq j$ as:

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} d(x, y) \tag{10}$$

where x and y are any points in cluster c_i and c_j, respectively, and $d(x,y)$ is the distance between x and y. We also define the diameter of a cluster c_i as:

$$\mathrm{diam}(c_i) = \max_{x,y \in c_i} d(x, y) \tag{11}$$

Then, Dunn's index [3, 23, 24, 34, 37] is defined as:

$$\mathrm{Dunn's} = \min_{i=1,\ldots,nc}\left\{\min_{j=i+1,\ldots,nc}\left(\frac{d(c_i, c_j)}{\max_{k=1,\ldots,nc}\cdot \mathrm{diam}(c_k)}\right)\right\} \tag{12}$$

It is clear that the larger Dunn′s index, the better the clustering results will be.

2.4 Xie-Beni (XB) index

XB index [3, 23, 24, 37] was originally a fuzzy clustering validity index. For a fuzzy clustering scheme, suppose we have the data set $X = \{x_i,\ 1 \leq i \leq N\}$, where N is the number of samples and the centroids v_j of clusters c_j, $1 \leq j \leq nc$, where nc is the total number of clusters. We seek to define the matrix of membership $U = u_{ij}$, where u_{ij} denotes the degree of membership of the sample x_i in the cluster c_j. To define the XB index, first one must define the sum of squared errors for fuzzy clustering. The sum of squared errors is defined as

$$J_m(U, V) = \sum_{i=1}^{N}\sum_{j=1}^{nc}(u_{ij})^m \|x_i - v_j\|^2 \tag{13}$$

where $1 \leq m \leq \infty$. In general, we use J_1 for the calculation. U is a partition matrix of fuzzy membership $U = u_{ij}$, and V is the set of cluster centroids $V = v_i$. In addition, the minimum inter cluster distance $d_{\min}$ must also be defined, as

$$d_{\min} = \min_{i,j} \|v_i - v_j\| \tag{14}$$

Supposing that we have N samples on the total data, XB index can be defined as

$$\mathrm{XB} = \frac{J_m}{N \times (d_{\min})^2} \tag{15}$$

XB index is designed to measure the fitness of fuzzy clustering, but it is also suitable for crisp clustering. The XB index has been mathematically justified in [49]. The lower the value of the XB index, the better the clustering should be.

2.5 PBM index

Like the XB index, the PBM index [37] is suitable for both fuzzy clustering and crisp clustering. Supposing that we have a data set with N samples $X = \{x_1, \ldots, x_N\}$, and nc clusters c_i, $1 \le i \le nc$, with respect centroids v_i, $1 \le i \le nc$ and a given a matrix of membership $U = \{u_{ij}\}$ to denote the degree of membership of the sample x_i in the cluster c_j, we define the measure of within-cluster scatter E_{nc} as

$$E_{nc} = \sum_{i=1}^{nc} \sum_{j=1}^{n_i} u_{ij} \|x_j - v_i\| \tag{16}$$

Then we define the inter-cluster measure D_{nc} as

$$D_{nc} = \max_{i,j}^{nc} \|v_i - v_j\| \tag{17}$$

The final PBM index is thus defined by:

$$\text{PBM} = \left(\frac{1}{nc} \times \frac{E_1}{E_{nc}} \times D_{nc} \right)^2 \tag{18}$$

where E_1 is a constant for a given data set, we could simply set E_1 equal to 1. In general, the larger the PBM index, the more compact each cluster shall be.

2.6 Davies–Bouldin (DB) index

The Davies–Bouldin (DB) index [3, 23, 24, 34, 37] is a function of the ratio of the sum of within-cluster scatter to between-cluster separation. The scatter within the it h cluster is computed as

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{x \in C_i} \{ \|x - z_i\|^q \} \right)^{\frac{1}{q}} \tag{19}$$

where $|C_i|$ is the number of samples belonging to cluster C_i, and z_i is the centroid of cluster C_i. Usually, we use $q = 2$ for the DB index, and the distance between cluster C_i and C_j is defined as

$$d_{ij,t} = \left(\sum_{s=1}^{p} \|z_{is} - z_{js}\|^t \right)^{\frac{1}{t}} = \|z_i - z_j\| \tag{20}$$

where $S_{i,q}$ is the qth root of the qth moment of the points in cluster i with respect to their mean, and is a measure of the dispersion of the points in cluster i. $S_{i,q}$ is the average Euclidean distance of the vectors in class i from the centroid of class i. $d_{ij,t}$ is the Minkowski distance of order t between the centroids that characterize clusters i and j. p is the dimension of features, and, in general, $t = 2$ is used for $d_{ij,t}$. Subsequently, the measurement based on the ratio of within-cluster scatter to between-cluster separation can be obtained:

$$R_{i,qt} = \max_{j, j \ne i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\} \tag{21}$$

The Davies–Bouldin index is then defined as

$$\text{DB} = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt} \tag{22}$$

where K is the number of clusters. In practice, we set $q = 1$, $t = 1$, so that:

$$S_i = \left(\frac{1}{|C_i|} \sum_{x \in C_i} \{ \|x - z_i\|_2 \} \right) \tag{23}$$

$$d_{ij} = \left(\|z_i - z_j\| \right) = \|z_i - z_j\| \tag{24}$$

$$R_i = \max_{j, j \ne i} \left\{ \frac{S_i + S_j}{d_{ij}} \right\} \tag{25}$$

It is clear that S_i is the average of Euclidean distance of the vector X in cluster C_i with respect to its centroid z_i, and d_{ij} is the Euclidean distance between the centroids z_i and z_j of the clusters C_i and C_j, respectively. The smaller the DB index, the better the clustering is supposed to be.

2.7 Clustering validity index for codebook size selection

Among the above clustering validity indices, Dunn's index, the DB index and the XB index are considered as the most adequate ones [3, 23, 24, 37]. However, the drawback of Dunn's index is its high calculation complexity. The derivation of the DB index has convincing theoretical support, but its problem is that it sums all the maximum values obtained on all clusters, which means that one extremely bad cluster separation may overwhelm all the other good cluster separations. In contrast, the XB index uses only the minimum distance between centroids of cluster pairs, focusing on the nearest cluster pairs and ignores the distribution of other clusters.

However, to obtain a group of potentially adequate codebook sizes, the applied clustering validity index is not only supposed to find a single best number of clusters, but also several best numbers of clusters. In other words, the clustering validity index used must have several optima that can depict a data set at multiple levels of granularity [11, 27, 43]. This property is important because the best number of clusters depends on different hierarchical levels. An adequate clustering validity index should not only offer different clusterings, but also a reasonable distinction among them. Again, we shall make it clear that the diversity does not guarantee a decrease in the error rate of an EoHMM. In fact, the diversity among HMM classifiers only offer a "possibility" to improve the EoHMM [41, 44].

The XB index is found to have this desirable property in our problem. The plot of XB index values versus the numbers of clusters gives a lot of minima with XB index values smaller than those of their neighbors, and these are actual optima for codebook sizes and are thus adequate for the construction of an EoHMM. We need to mention that if we only consider the optimization of codebook size, we might apply other kinds of clustering validity indices or even other clustering techniques. The use of XB index is due to its multiple minima for the purpose of construction of ensemble.

In the next section, we detail the process for construction of EoHMMs based the on XB index, and the ensemble selection and classifier combination schemes considered.

2.8 Generation of HMM classifiers

Given a data set of $X = \{x_i,\ 1 \leq i \leq N\}$, where N is the number of samples, and defining a possible range M for the numbers of clusters j, $1 \leq j \leq M$, the cluster index should give the fitness $F_t(j)$ for these M clusterings, with $1 \leq j \leq M$. Due to the tremendous size of data set, our machines just cannot store the data and do the calculation, so we need to use a smaller data set with N_s samples extracted from N samples for clustering goodness evaluation, $N_s = \eta \dot{N}$, where η is the proportion of samples used. The selection of the N_s sample is conducted just to make the machine work and at the same time to have much data as possible.

Assuming that we intend to select L best clusterings, then these clusterings could be selected with clustering validity index values $F_t(j)$, $1 \leq j \leq L$. These selected numbers of clusterings then serve as the sizes of the codebook of HMM classifiers. The selected codebook sizes are used again for the clustering on all N samples, with the result that the respective codebooks are generated. Each HMM is then trained with a different codebook. This pool of HMM classifiers must go further through the ensemble selection process to decide which classifiers are adequate for construction of an ensemble. Then the selected classifiers would be combined according to a fusion function.

Given the various scheme of objective functions for ensemble selection and the fusion functions for classifier combination, it is of the great interest to test these schemes on real problem. We perform the experiment on a benchmark data base in the next section.

3 Experiments with EoHMMs

The experimental data was extracted from NIST SD19 as a 10-class handwritten numeral recognition problem. As a result, there is an HMM model for each class, and 10 HMM models for an HMM classifier. Five databases were used: the training set with 150,000 samples (hsf_{0−3}) was used to create 40 HMM classifiers, 20 of them being column-HMM classifiers and other 20 being row-HMM classifiers. The large size of the data set for training can lead to a better recognition rate for each individual classifier. For codebook size selection evaluated by clustering validity indices, due to the extremely large data set (150,000 images are equivalent to 5,048,907 columns and 6,825,152 rows, with 47 features per column or per row), we use only the first 10,000 images from the training data set to evaluate the quality of the clustering, and they are equal to 342,910 columns and 461,146 rows.

We need to make it clear why we use both row- and column-HMM classifiers in our experiment. The use of both row- and column-HMM classifiers have a dual purpose. First, we can demonstrate that our ensemble generation method works not only for a column HMM classifier but also for a row HMM classifier. Second, since we are applying multiple classifier system, column- and row-HMM classifiers are also designed to enhance the diversity among classifiers and thus enhance the performance of final EoHMM.

The sampling may present a slight bias in clustering, but, because even the sampled data set contains 0.34 millions column samples and 0.46 millions row samples, we believe it is large enough to evaluate the quality of the clustering and discover the multiple-level granularity of the data set. Note that, at the clustering evaluation stage, we only examined the different numbers of clusters with the clustering validity index to select several suitable codebook sizes for an EoHMM. Then, the codebooks were generated with the whole training set, according to the previously selected codebook sizes. The training validation set of 15,000 samples was used to stop HMM classifiers training once the optimum had been achieved. The optimization set containing 15,000 samples (hsf_{0−3}) was used for GA searching for ensemble selection. To avoid overfitting during GA searching, the selection set containing 15,000 samples (hsf_{0−3}) was used to select the best solution from the current population according to the defined objective function and then to store it in a separate archive after each generation. The selection set is also used for the final validation of HMM classifiers. Using the best solution from this archive, the test set containing 60,089 samples (hsf_{7}) was used to evaluate the accuracies of EoC.

Each column HMM used 47 features obtained from each column, and each row used 47 features obtained from each row (see Fig. 1). The features were extracted by the same means described in [5, 6], and K-means was used for vector-quantization to generate codebooks for the HMM. Note that K-means is used because it forms spherical and compact clusters, and when we apply different numbers of

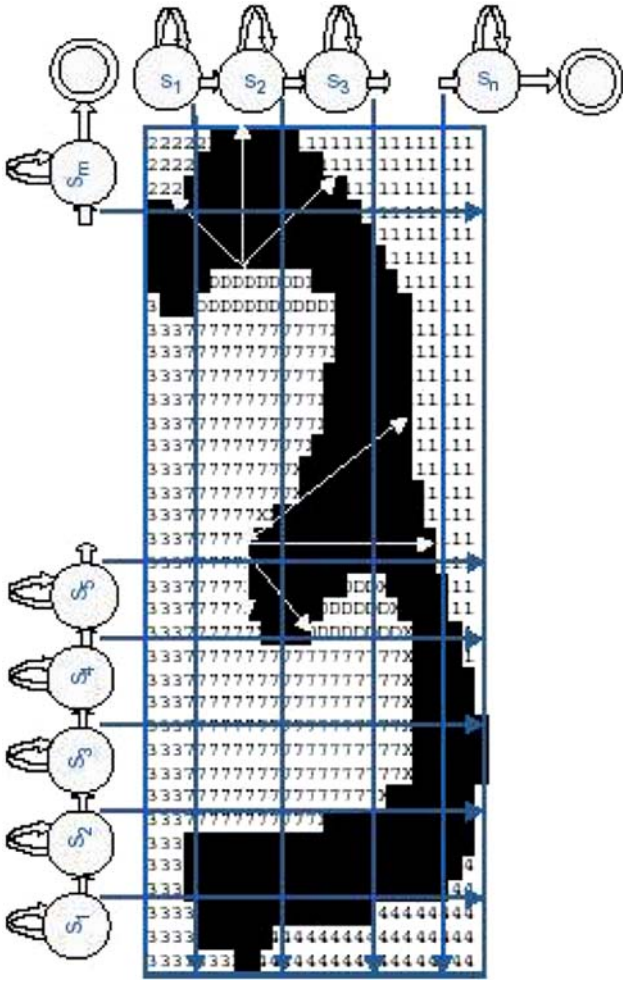

Fig. 1 The benchmark HMM classifiers in [5, 6]: For any character image, we scan the image from left to right, and obtain a sequence of columns as the observations; we then scan this image again from top to bottom, and obtain a sequence of rows as the observations. By this means, features are extracted from each column and each row, a column HMM classifier and a row HMM classifier are thus constructed for isolated handwritten numeral recognition

clusters, K-means thus actually helps create rather different clusters and generate some diversity among different clusterings. However, K-means can be substituted by other clustering techniques, and this will not change the proposed ensemble generation method (Fig. 2).

The number of HMM states was optimized by the method described in [46]. The HMMs were trained by Baum–Welch algorithm [39, 40]. The benchmark HMM classifiers used 47 features, with the codebook size of 256 clusters [5, 6]. For benchmark column HMM, we have a recognition rate of 97.60%, and for benchmark row HMM the classification accuracy was about 96.76%, while the combination of the benchmark column HMM and the benchmark row HMM achieved a rate of 98.00%. The possible upper limit of classification accuracy (the oracle) is defined as the ratio of samples which are classified correctly by at least one classifier in the pool to all samples. The oracle achieved a rate of 99.76% on the test set, considering the pool of the whole HMM classifiers. For combining classifiers, 12 different fusion functions were tested.

3.1 Behaviors of clustering validity indices in HMM features

To decide on suitable codebook sizes of HMM, we carried out clusterings on HMM features. Due to the large data size, it is clear that we could not use all the training set to do the clusterings, all with different numbers of clusters. As a result, the first 10,000 images in training set were used for clustering, these images containing 342,910 columns and 461,146 rows.

Before we constructed the EoHMM, we performed K-means clusterings with different numbers of clusters on HMM features, and showed the properties of clustering validity indices in this problem. Processing clusterings from 3 to 2,048 clusters for the clustering task, we showed the relationship between the XB index and the number of clusters for column HMM features, and many minima can be observed (Fig. 3a). The optimum codebook size defined by the XB index value is 1893 clusters, and, with this codebook size, the column HMM classifier can achieve 98.92% recognition rate on the validation set, and 98.32% on the test set. A similar tendency can be observed in row HMM features (Fig. 3b). This property, as we argued, is important to get multiple levels of granularity of the data set, and it offers codebook sizes for HMMs with the potential to perform well.

In contrast, the relationship between the DB index and the number of clusters was much more ambiguous. In general, for column HMM features, the curve reached its minimum at 5 and maximum at 132, then decreased almost constantly (Fig. 4a). Apparently, a simple 5-cluster optimum is not useful for the codebook, as the corresponding column HMM can achieve a classification accuracy of only 71.69% on validation set, and 69.43% on the test set. Moreover, most of the optima selected by the DB index will contain fewer than 132 clusters.

As we stated previously, the PBM index is less convincing. The PBM index suggests that the best clustering is with 3 clusters for column HMM (Fig. 5a), which can achieve a recognition rate of only 63.49% on the validation set, and 61.72% on the test set. Note that the maximum value in PBM represents the optimum. After slight variation in the beginning, the curve decreases continuously. The PBM thus encourages the use of small codebook sizes, which cannot lead to any useful results for this problem.

For RS and RMSSTD, the optima are located on the knees of the curves, but it might not be easy to find out these knees. For RS, the so called knee might be found at

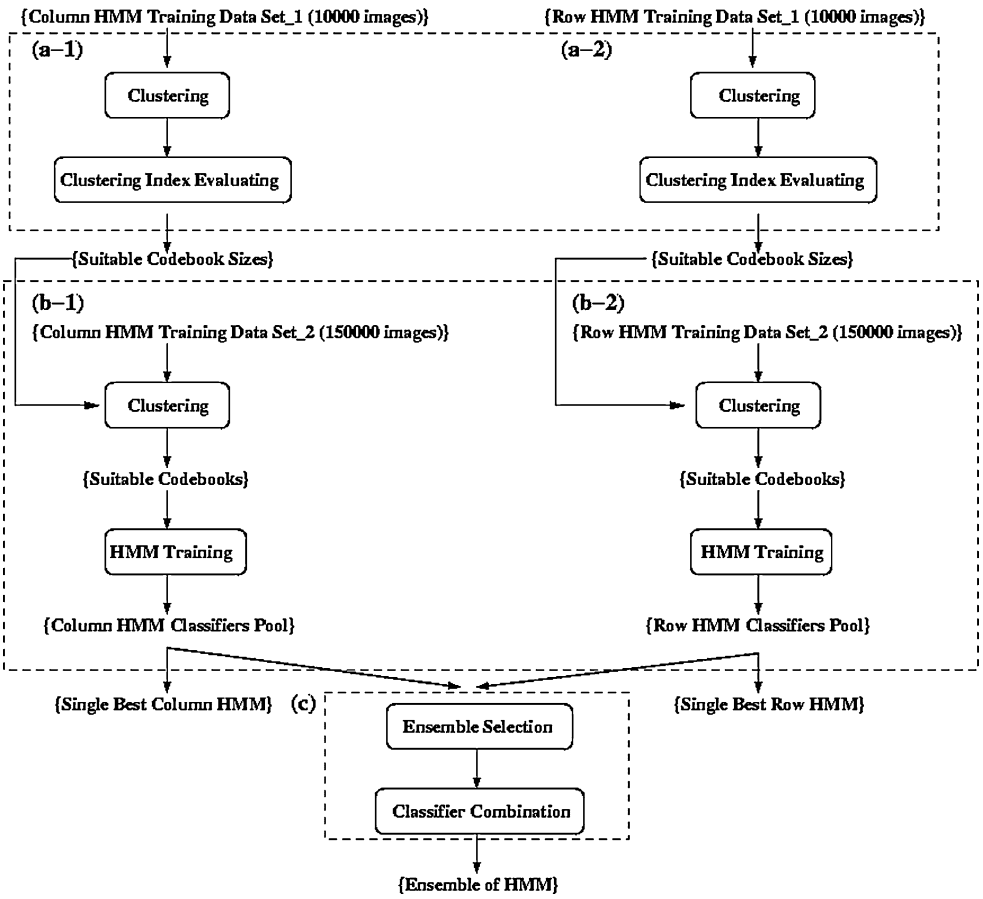

Fig. 2 The EoHMM classification system approach includes: **a** the adequate codebook sizes searching; **b** codebooks generation and HMM classifiers training; **c** EoHMM selection and combination. Both **a** and **b** were carried out separately on column and row HMM classifiers

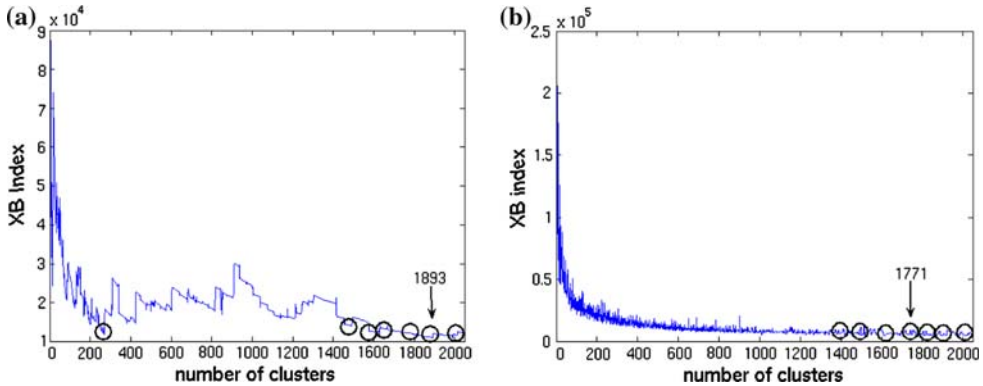

Fig. 3 The relationship between XB index and the number of clusters for: **a** HMM column features; **b** HMM row features. The *circled areas* indicate the places where the best 40 optima were found. The *arrow* indicates the smallest XB value with the respective number of clusters. Note that clusterings were carried out on the first 10,000 images of the training data set (see Table 2 for details)

roughly 140 clusters for column HMM (Fig. 7a), where column HMM achieved 98.14% recognition rate on the validation set, and 97.36% on the test set. For RMSSTD, the knee is roughly at 131 clusters for column HMM (Fig. 6a), with which column HMM can achieved a 98.08% classification accuracy on the validation set, and 97.12% on the test set. But the disadvantage common to the RS and RMSSTD indices is that they give only one optimum solution, and there is no way to find multiple optima, which makes it impossible to use them for the construction of an EoHMM. Finally, we must mention that, given the size of the data set, it is impossible to evaluate Dunn′s index, because Dunn′s index has to calculate the distances between $342{,}910^2$ sample pairs for column HMM and $461{,}146^2$ sample pairs for row HMM (Table 1).

3.2 The multiple levels of granularity in codebook size selection

These observations indicate that XB index has the properties desired for HMM codebook size selection. Note that, in order to construct an EoHMM which performs well, we need to select several fit codebooks, and, moreover, these

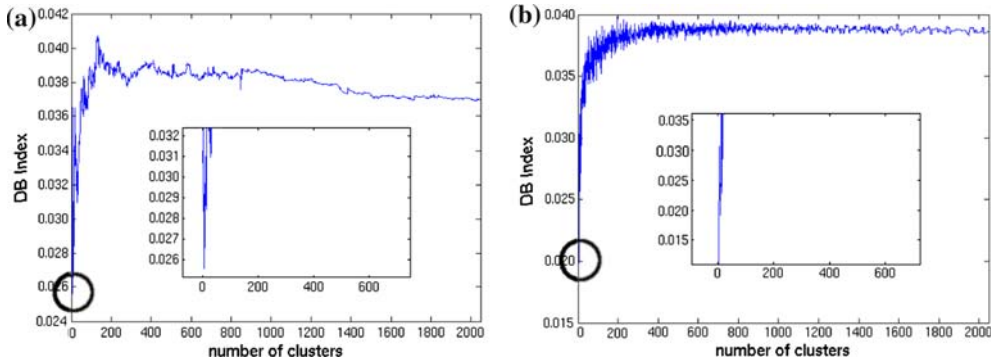

Fig. 4 The relationship between DB index and the number of clusters for: **a** HMM column features; **b** HMM row features. Optima are minima in DB index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10,000 images of the training data set

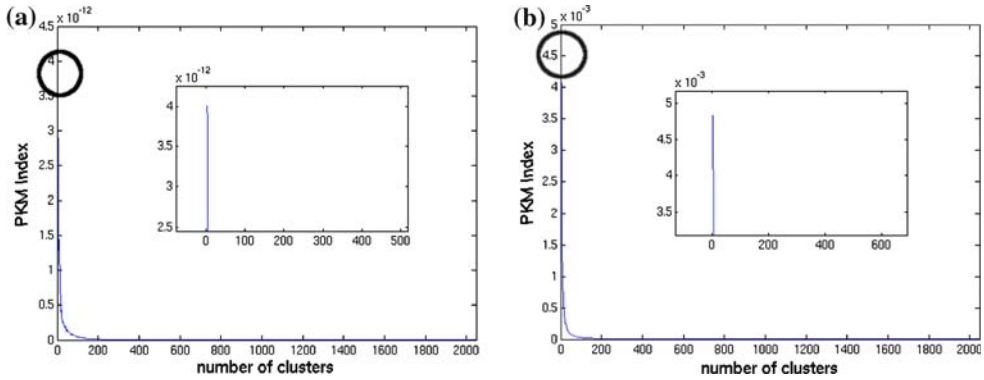

Fig. 5 The relationship between PBM index and the number of clusters for: **a** HMM column features; **b** HMM row features. The optimum has the maximum value in PBM index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10,000 images of the training data set

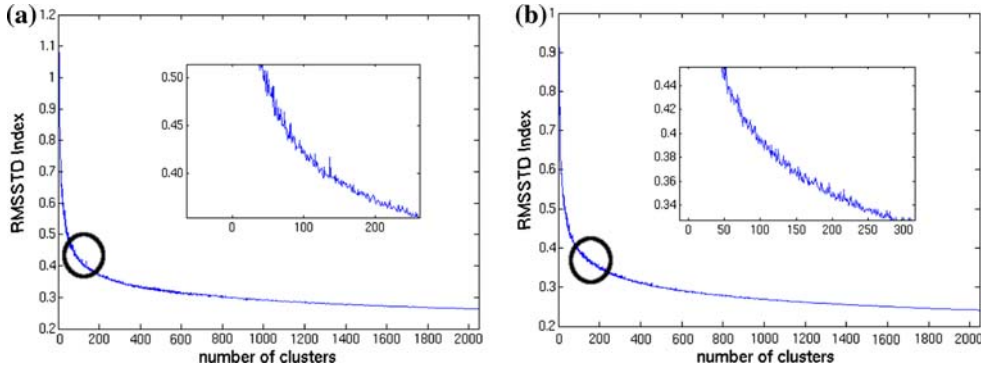

Fig. 6 The relationship between RMSSTD index and the number of clusters for: **a** HMM column features; **b** HMM row features. The optimum is located on the "knee" of the curve in RMSSTD index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10,000 images of the training data set

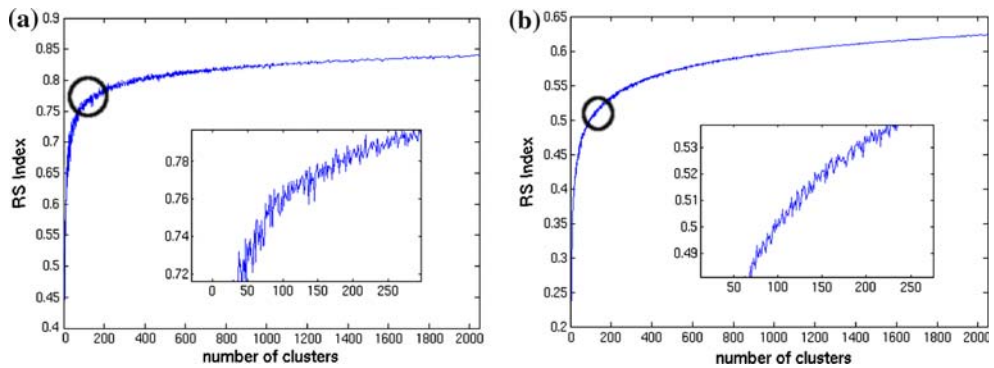

Fig. 7 The relationship between RS index and the number of clusters for: **a** HMM column features; **b** HMM row features. The optimum is located on the "knee" of the curve in RS index, we enlarge the part where the optimum is located. Note that clusterings were carried out on the first 10,000 images of the training data set

Table 1 Comparison classification accuracies of column-HMM classifiers with codebook sizes selected by different clustering validity indices

Clustering validity index	XB	DB	PBM	RS	RMSSTD	Dunn's
Validation set	98.92%	71.69%	63.49%	98.14%	98.08%	N.S.
Test set	98.32%	69.43%	61.72%	97.36%	97.12%	N.S.

codebooks must lead to diverse HMM classifiers so that the combination of these HMM classifiers can actually achieve even better performance. As we observed in the previous section, the XB index not only finds fit codebooks, but it also reveals the multiple granularity of the data set. Moreover, its calculation is much less time-consuming than Dunn's index. All these advantages favor the use of the XB index.

Intuitively, because the clusterings with different granularity levels are located in different neighborhoods, it is very unlikely that the codebook size optima found in a single neighborhood can represent the concept of the multiple-level granularity. For this reason, it is important to have clusterings in different neighborhoods. To satisfy this condition, we may simply require the selected clusterings have non-adjacent numbers of clusters.

Although the multiple-level granularity implicates the diversity related to different partitions between clusterings, we still need to confirm that the concept of the multiple-level granularity can also lead to better EoHMM performance, i.e., the optima found in different neighborhoods can lead to better EoHMM performance than those found in the same neighborhood. Thus, we investigated and compared the performances of EoHMMs constructed by codebook sizes selected by the XB index optima in the same neighborhood and those in different neighborhoods.

We performed clusterings on the first 10,000 images in the training set with numbers of clusters from 3 to 2,048. For HMM column features, the best codebook sizes defined by the XB index were 1893, 1892, 1891, 1890 and 1889 clusters, with an XB index of 10943, 10949, 10955, 10961, 10967, respectively. Note that these optima were selected by absolute minima in the XB index, and no multiple levels of granularity were involved. Consequently all selected codebook sizes are in the same neighborhood.

However, if we require that all optima have an XB index value smaller than those of their neighbors, i.e., if we require simply that for any selected number of cluster $nc \geq 2$, its XB value XB(nc) must be smaller than those of its two nearest neighbors, XB(nc) $<$ XB($nc + 1$), XB(nc) $<$ XB($nc - 1$), then we can obtain codebook sizes in different neighborhoods. Under this condition, the clusterings with 1892, 1891, 1890 and 1889 clusters do not qualify. In contrast, we will have the following best codebook sizes, as defined by the XB index: 1893, 1991, 1986, 1965 and 2012 clusters, with an XB index of 10943, 10982, 11478, 11498 and 11605, respectively. Note that, in this case, the optima were selected by relative minima in XB index, i.e. we required that these minima be the smallest in their neighborhoods, and thus we took into account of multiple levels of granularity.

The same process was carried out for HMM row features, and the best codebook sizes defined by the XB index were 1771, 1770, 1769, 1768 and 1767 clusters, with an XB index as 4565, 4569, 4572, 4574 and 4577, respectively. If we require that all optima have an XB index value smaller than those of their neighbors, we will have the following best codebook sizes, as defined by the XB index: 1771, 1809, 2022, 1975 and 1978 clusters, with an XB index of 4565, 4675, 4741, 4764 and 4782, respectively.

We then construct two basic EoHMMs on both the column HMM features and the row HMM features. One EoHMM was constructed with codebook sizes with XB indices that are the absolute minima, while another EoHMM was constructed with codebook sizes with XB index values that are relative minima, i.e., their XB indices are smaller than their neighbors. We then evaluated the performance of these two EoHMM on both the column HMM feature and the row HMM feature.

Even though the ensembles are constructed with a small number of classifiers, we can observe that optima found in different neighborhoods by XB index are slightly better than those found in the same neighborhoods (Table 2). Note that all HMM classifiers are trained with the same number of samples and the whole feature set, and they are different from one another only in the codebooks. We can expect that the difference will be more apparent when more HMM classifiers are used. To prove that an EoHMM constructed with optima found in different neighborhoods by the XB index can significantly enhance the performance, we went on constructing 20-column HMM classifiers and 20-row HMM classifiers with optima in different neighborhoods (see below). These HMM classifiers will later be combined and the improvement be measured.

Table 2 Comparison classification accuracy with ensembles composed of 5 absolute optima (ABS) and of 5 relative optima (REL) in terms of XB index

	COL-ABS (5)	COL-REL (5)	ROW-ABS (5)	ROW-REL (5)
Validation set	99.12%	99.13%	98.80%	98.88%
Test set	98.49%	98.54%	97.92%	98.14%

Results are shown on test set and validation set. The number of classifiers are shown in parenthesis

3.3 Optimum codebooks selected by XB index

For all clusterings from 3 to 2,048 clusters on the first 10,000 images in the training set, the 20 smallest minima with XB index values smaller than those of their neighbors were selected as the adequate numbers of clusters, i.e. the 20 most pertinent sizes of codebooks. Once the optimum codebook sizes were selected, we performed clusterings on the whole training data (including 150,000 images) with the selected numbers of clusters to generate HMM codebooks. These codebooks were then used for HMM sequence observations and HMM classifier training. This process was carried out for the column features as well as for the row features, all HMM classifiers being trained with the whole feature set and all the training samples. Thus, 20-column HMM classifiers and 20-row HMM classifiers were generated, for a total of 40 HMM classifiers (Table 3).

The best single column HMM achieved a classification accuracy of 98.42% with a codebook size of 1965, which is 0.82% better than the benchmark column HMM classifier; and the best row HMM classifier had a recognition rate of 97.97%, with a codebook size of 1786, which is 1.21% better than the benchmark row HMM. Compared with the benchmark column HMM classifier (97.60%) and with the benchmark row HMM classifier (96.76%), codebooks selected by the XB index gave some improvement to performance. Note that performance is not necessarily proportional to the size of the codebooks. Based on these HMM classifiers, we then construct the EoHMMs.

3.4 Column-EoHMM and row-EoHMM

Without carrying out any ensemble selection process, we simply constructed three ensembles composed entirely of column-HMM classifiers (COL-HMM), entirely of row HMM classifiers (ROW-HMM) and of all HMM classifiers (ALL-HMM) (Table 4). The ensembles were then combined by the SUM rule [28, 51, 52] and PCM-MAJ rule [30], since these two fusion functions have been shown to be very effective [28, 30]. We note that the ensemble of column-HMM classifiers improved by 0.14% over the single best column HMM classifier using the PCM-MAJ fusion function, while the ensemble of row HMM classifiers improved by 0.29% over the single best row HMM classifier using the SUM fusion function. This means that by using different codebook sizes to construct an EoHMM, we explored the diversity of different codebooks of HMM and achieve a better result. Moreover, the ensemble of all HMM classifiers gave the best performance, given that the obvious diversity between the column- and the row-HMM classifiers. With the PCM-MAJ rule, ALL-HMM performed 0.42% better than the single best HMM classifier, and achieved the best classification accuracy.

Table 4 Comparison of classification accuracies on test data set with two different fusion functions and on different types of EoHMMs

EoHMM	Fusion functions	
	PCM-MAJ	SUM
COL-HMM (20)	98.56%	98.55%
ROW-HMM (20)	98.20%	98.26%
ALL-HMM (40)	98.84%	98.78%

The number of classifiers are shown in parenthesis

3.5 Ensemble selection

For evaluating classifier combinations, another approach is to go through the process of ensemble selection, because one of the most important requirements of EoCs is the presence of diverse classifiers in an ensemble. We tested the simple majority voting error (MVE) and the SUM rule, because of their reputation for being two of the best objective functions for selecting classifiers for ensembles [41]. We also tested 10 different compound diversity functions (CDFs) [29], which combine the pairwise diversity measures with individual

Table 3 Classification accuracies of 20-column-HMM classifiers and 20-row HMM classifiers generated by different codebook sizes on test data set

CCS	1893	1991	1986	1965	2012	1934	1796	1998	1627	269
CA	98.32%	98.33%	98.35%	98.40%	98.30%	98.39%	98.34%	98.33%	98.33%	97.56%
CCS	2040	264	2048	1625	1715	1665	1667	1491	1488	1456
CA	98.42%	97.55%	98.35%	98.37%	98.37%	98.34%	98.32%	98.29%	98.29%	98.30%
RCS	1771	1809	2022	1975	1978	1786	1657	1897	1851	1694
CA	97.84%	97.88%	97.93%	97.73%	97.95%	97.97%	97.83%	97.86%	97.93%	97.89%
RCS	1904	1505	1503	1920	1616	1520	1517	1835	1421	1490
CA	97.83%	97.84%	97.80%	97.83%	97.89%	97.84%	97.75%	97.90%	97.70%	97.73%

CCS: Column codebook size; RCS: Row codebook size; CA: Classification accuracy. The codebook sizes are ranked by their XB index from left to right

classifier performance to estimate ensemble accuracy, but do not use the global performance of the EoC. CDFs have been shown to be better than traditional diversity functions for ensemble selection [29].

These objective functions were evaluated by genetic algorithm (GA) searching. We used GA because the complexity of population-based searching algorithms can be flexibly adjusted, depending on the size of the population and the number of generations with which to proceed. Moreover, because the algorithm returns a population with the best combination, it can potentially be exploited to prevent generalization problems [41]. GA was set up with 128 individuals in the population and with 500 generations, which means 64,000 ensembles were evaluated in each experiment. The mutation probability was set to 0.01, and the crossover probability to 50%. With 12 different objective functions (MVE, SUM, 10 compound diversity functions, including the disagreement measure (CDF-DM), the double-fault (CDF-DF), Kohavi-Wolpert variance (CDF-KW), the interrater agreement (CDF-INT), the entropy measure (CDF-EN), the difficulty measure (CDF-DIFF), generalized diversity (CDF-GD), coincident failure diversity (CDF-CFD), Q-statistics (CDF-Q), and the correlation coefficient (CDF-COR) [29]), and with 30 replications, 23.04 million ensembles were searched and evaluated. A threshold of three classifiers was applied as the minimum number of classifiers for an EoC during the whole searching process.

The selected ensembles were then combined by two types of fusion functions: The SUM rule [28, 52] and the PCM-MAJ rule [29]. Among all objective functions, the best ensemble was selected by the CDF-CFD and composed of 16 HMM classifiers. The recognition rate achieved by the selected ensemble is 98.80% with the SUM rule, and 98.84% with the PCM-MAJ rule. For all replications of GA searching, the variances are smaller than 0.01%, which indicates that the GA searching gives quite stable results.

We showed the results in Tables 5 and 6. We note that the selected ensemble did perform better than column-HMM classifiers and row-HMM classifiers, but showed limited improvement compared with the ensemble of all the HMM classifiers. The PCM-MAJ rule performed better than the SUM rule on the selected ensemble. The PCM-MAJ has an improvement of 0.86% compared with the Benchmark EoHMM, and of 0.16% compared with XB-Selection EoHMM.

Figures 8 and 9 showed the rejection curves of the SUM rule and of the PCM-MAJ rule, respectively. For the Sum rule, it is apparent that the selected ensemble performed better than the column-HMM ensemble and the row-HMM ensemble, and had the comparable performance with the ensemble of all HMM classifiers (Fig. 8).

Table 5 Best performances from 30 GA replications on the test data set. The numbers of classifiers are noted in parenthesis. The SUM was used as the fusion function in EoC

Recognizers	Column-HMM classifiers	Row-HMM classifiers	Column- and Row-HMM classifiers
Benchmark	97.60% (1/–)	96.76% (1/–)	98.00% (2/SUM)
XB selection	98.40% (1/–)	97.97% (1/–)	98.70% (2/SUM)
Classifier pool	98.55% (20/SUM)	98.26% (20/SUM)	98.78% (40/SUM)
EoHMM selection	–	–	98.80% (16/SUM)

Table 6 Best Performances from 30 GA replications on the test data set

Recognizers	Column-HMM classifiers	Row-HMM classifiers	Column- and Row-HMM classifiers
Classifier pool	98.56% (20/PCM-MAJ)	98.20% (20/PCM-MAJ)	98.84% (40/PCM-MAJ)
EoHMM selection	–	–	98.86% (16/PCM-MAJ)

The numbers of classifiers are noted in parenthesis. The PCM-MAJ was used as the fusion function in EoC

If the PCM-MAJ rule was applied, we see that it offered a better improvement than the SUM rule for the rejection rate smaller than 2%. But unlike the SUM rule, it is hard for the PCM-MAJ rule to do more rejection when the majority of classifier-pairs agrees on the most of samples [30]. After achieving a certain threshold, the system needs a much larger rejection rate to do further rejection. What is more, if all classifier-pairs agree on the most of samples, it is impossible to have more rejection, as in the case of the column-HMM ensemble (Fig. 9). Note that to apply PCM-MAJ, the ensembles must have more than two classifiers, and thus we cannot use PCM-MAJ as a fusion function on the benchmark EoHMM and on the XB-selection EoHMM.

4 Discussion

In this work, we proposed a new ensemble of HMM classifiers generation method based on different codebook sizes. The work is to check two major assumptions: first, if we apply different codebook sizes to generate HMM classifiers, then these HMM classifiers will be adequate to construct a performing EoHMM; second, we can use the clustering validity indices to find out these adequate codebook sizes for EoHMM generation.

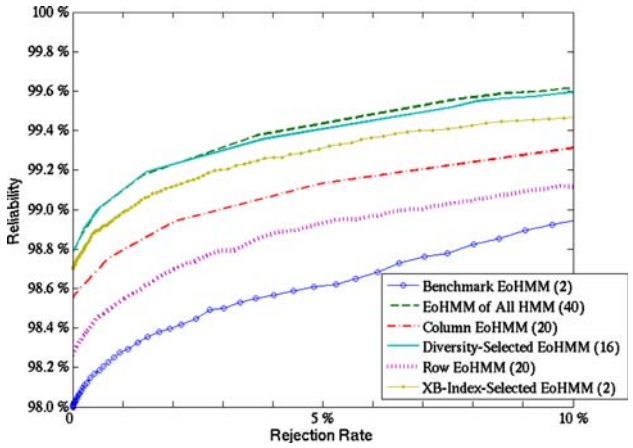

Fig. 8 The Rejection mechanism with the SUM rule

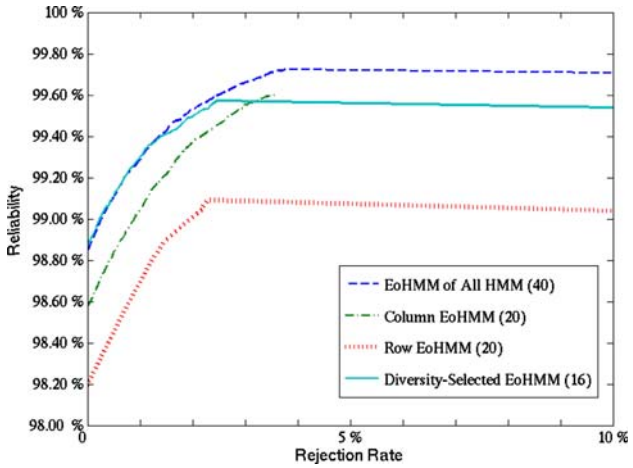

Fig. 9 The Rejection mechanism with the PCM-MAJ rule

Of most of clustering validity indices, the XB index, Dunn's index and the DB index were regarded as the most reasonable. It seems that XB index is more helpful to select an adequate codebook size for HMM. Nevertheless, the most important reason that we shall use XB index rather than other clustering validity indices is that XB index offers multiple choices for codebook sizes. Since we focus on constructing an EoHMM with various codebook size, XB index is more appealing than others.

The HMM classifiers constructed with codebook sizes selected by the XB index show a clear improvement compared with benchmark HMM classifiers, in both column-HMM classifiers and row-HMM classifiers [5, 6]. With an improvement of 0.80% over the benchmark column HMM classifier and 1.21% over the benchmark row HMM classifier, the usefulness of the XB index in optimizing HMM is undeniable.

As a by-product, we can also use these HMM classifiers trained with different codebook sizes to construct an EoHMM. With the SUM fusion function, the improvement in the classification accuracy of the ensemble of column-HMM classifiers is 0.14% over that of the single best column HMM classifier, while the improvement in the accuracy of the ensemble of row HMM classifiers is 0.29% over that of the single best row HMM classifier. Considering that the best column HMM classifier already has a classification accuracy of 98.40% and the best row HMM classifier has a recognition rate of 97.97%, this improvement is significant. Such an improvement also indicates that the disadvantage of discrete HMM can be compensated for by EoHMM based on various codebook sizes.

Considering the objective function for EoHMM ensemble selection, the SUM rule and all the CDF rules give similar and comparable results. We also note that, by combining column-HMM classifiers and row HMM classifiers, the single best EoHMM of all the replications can have a classification accuracy of 98.86%. This is about 0.30% better than COL-HMM, thanks to the further diversity contributed by row- and column features (Tables 5, 6).

We note that the proposed method has a speed-up advantage over other EoHMM creation schemes. Suppose we need to construct M HMM classifiers for EoHMM, given S possible codebook sizes, the proposed scheme evaluates S clusterings using the XB index and then trains M HMM classifiers. For other ensemble creation methods, such as bagging, boosting, and random subspaces, we need to train $M \times S$ HMM classifiers and then select among them for the best codebook size. This offers a significant speed-up in the optimization of the codebook sizes and a new ensemble creation method.

Considering other classification methods applied in the same data set, KNN with 150,000 samples can achieve 98.57% accuracy, MLP can achieve 99.16% accuracy [36], and the use of SVM can achieve a 99.30% recognition rate with a pairwise coupling strategy and a 99.37% with the one-against-all strategy [35]. EoHMM performance very close to that and its further optimization might achieve better results.

5 Conclusion

A fast codebook size optimization method for HMM and a new scheme of ensemble of discrete HMM were proposed in this paper. The codebook size was selected by evaluating the quality of clustering during the construction of codewords. Because the method does not require any HMM classifiers training, the proposed scheme offers a significant speed-up for codebook size optimization. In order to fairly evaluate clustering quality, we used a clustering validity index based on different predefined numbers of clusters.

Though a number of clustering validity indices were available, we used the XB index because it has the strong theoretical support [49] and has been shown effective in

clustering [3, 37]. Moreover, the XB index demonstrated the property of discovering multiple levels of granularity in the data set, which would allow us to select adequate codebook sizes. In general, the HMM classifiers with codebook sizes selected by the XB index demonstrated an apparently better performance than benchmark HMM classifiers. As a by-product, we can construct an EoHMM trained with the full samples and full features based on different codebook sizes. Because the XB index gives multiple fit codebook sizes, these codebook sizes could result in more accurate and diverse HMM classifiers, and thus provide us with an EoHMM. The combination of column- and row-HMM classifiers further improve the global performance of EoHMM.

To conclude, the result suggests that the new EoHMM scheme is applicable. The degradation associated with vector quantization in discrete HMM is compensated by the use of EoHMM without the need to deal with a number of optimization of parameters found in continuous HMM. EoHMM can also explore the advantage of the number of different ensemble combination methods proposed in the literature.

Future work is planned to further improve the performance of EoHMM by exploring the issue of the number of states that need to be optimized as well. With EoHMM based on different numbers of states, it will be possible to obtain further improvement without adding any parameters optimization problems, which will be of the great interest in the application of HMM. Furthermore, the codebook pruning will be also an interesting issue for the decrease of the computation cost for the construction of HMM classifiers.

Acknowledgment This work was supported in part by grant OGP0106456 to Robert Sabourin from the NSERC of Canada.

References

1. Altincay H (2005) A Dempster-Shafer theoretic framework for boosting based ensemble design. Pattern Anal Appl J 8(3): 287–302
2. Arica N, Vural FTY (2000) A shape descriptor based on circular Hidden Markov Model. In: 15th International conference on pattern recognition (ICPR00)
3. Bandyopadhyay S, Maulik U (2001) Non-parametric genetic clustering: comparison of validity indices. IEEE Trans Syst Man Cybern Part-C 31(1):120–125
4. Bengio Y (1999) Markovian models for sequential data. Neural Comput Surv 2:129–162
5. Britto A Jr. (2001) A two-stage HMM-based method for recognizing handwritten numeral strings. Ph.D. Thesis, Pontifical Catholic University of Paraná
6. Britto AS, Sabourin R, Bortolozzi F, Suen CY (2003) Recognition of handwritten numeral strings using a two-stage Hmm-based method. Int J Doc Anal Recognit 5(2–3):102–117
7. Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. Int J Inf Fusion 6(1):5–20
8. Conversano C (2002) Bagged mixtures of classifiers using model scoring criteria. Pattern Anal Appl 5(4):351–362
9. Davis RIA, Lovell BC (2004) Comparing and evaluating HMM ensemble training algorithms using train and test and condition number criteria. Pattern Anal Appl 6(4):327–335
10. Dietterich TG (2002) Machine learning for sequential data: a review. In: Structural, Structural, syntactic, and statistical pattern recognition, Lecture Notes in Computer Science, vol 2396. Springer, Heidelberg, , pp 15–30
11. Eppstein D (1998) Fast hierarchical clustering and other applications of dynamic closest pairs. In: Proceedings of the ninth ACM-SIAM symposium on discrete algorithms, pp 619–628
12. Grove A, Schuurmans D (1998) Boosting in the limit: maximizing the margin of learned ensembles. In: Proceedings of the fifteenth national conference on artificial intelligence, pp 692–699
13. Guenter S, Bunke H (2005) Off-line cursive handwriting recognition using multiple classifier systems—on the influence of vocabulary, ensemble, and training set size. Opt Lasers Eng 43:437–454
14. Guenter S, Bunke H (2004) Ensembles of classifiers derived from multiple prototypes and their application to handwriting recognition. International workshop on multiple classifier systems (MCS 2004), pp 314–323
15. Guenter S, Bunke H (2003) Off-line cursive handwriting recognition—on the influence of training set and vocabulary size in multiple classifier systems. In: Proceedings of the 11th conference of the international graphonomics society
16. Guenter S, Bunke H (2002) A new combination scheme for HMM-based classifiers and its application to handwriting recognition. In: Proceedings of 16th international conference on pattern recognition II, pp 332–337
17. Guenter S, Bunke H (2002) Generating classifier ensembles from multiple prototypes and its application to handwriting recognition. In: Proceedings of the 3rd international workshop on multiple classifier systems, pp 179–188
18. Guenter S, Bunke H (2002) Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. In: Proceedings of the 8th international workshop on frontiers in handwriting recognition, pp 183–188
19. Guenter S, Bunke H (2003) Ensembles of classifiers for handwritten word recognition. Int J Doc Anal Recognit 5(4):224–232
20. Guenter S, Bunke H (2003) New boosting algorithms for classification problems with large number of classes applied to a handwritten word recognition task. In: Proceedings of the 4th international workshop on multiple classifier systems, pp 326–335
21. Guenter S, Bunke H (2003) Fast feature selection in an HMM-based multiple classifier system for handwriting recognition. Pattern recognition, proceedings of the 25th DAGM symposium, pp 289–296
22. Guenter S, Bunke H (2004) Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm. Electron Lett Comput Vis Image Anal 3(1):25–44
23. Halkidi M, Batistakis Y, Vazirgiannis M (2001) On clustering validation techniques. J Intell Inf Syst 17(2–3)
24. Halkidi M, Batistakis Y, Vazirgiannis M (2002) Clustering validity checking methods: part II. SIGMOD Rec 31(3):19–27
25. Ho TK (1998) The random space method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844
26. Huang X, Acero A, Hon H (2001) Spoken language processing—a guide to theory, algorithm, and system development. Prentice-Hall, Englewood Cliffs
27. Johnson E, Kargupta H (1999) Collective, hierarchical clustering from distributed, heterogeneous data. In: Large-scale parallel KDD systems, pp 221–244
28. Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20(3):226–239

29. Ko A, Sabourin R, Britto A Jr. (2006) Combining diversity and classification accuracy for ensemble selection in random subspaces. In: IEEE world congress on computational intelligence (WCCI 2006)—international joint conference on neural networks (IJCNN 2006)
30. Ko A, Sabourin R, Britto A Jr. (2006) Evolving ensemble of classifiers in random subspace. Genetic and evolutionary computation conference (GECCO 2006)
31. Kuncheva LI (2002) A theoretical study on six classifier fusion strategies. IEEE Trans Pattern Anal Mach Intell 24(2):281–286
32. Kuncheva LI, Skurichina M, Duin RPW (2002) An experimental study on diversity for bagging and boosting with linear classifiers. Int J Inf Fusion 3(2):245–258
33. Masulli F, Valentini G (2004) Effectiveness of error correcting output coding methods in ensemble and monolithic learning machines. Pattern Anal Appl 6(4):285–300
34. Maulik U, Bandyopadhyay S (2002) Performance evaluation of some clustering algorithms and validity indices. IEEE Trans Pattern Anal Mach Intell 24(12):1650–1654
35. Milgram J, Cheriet M, Sabourin R (2005) Estimating accurate multiclass probabilities with support vector machines. International joint conference on neural networks (IJCNN 05), pp 1906–1911
36. Oliveira LS, Sabourin R, Bortolozzi F, Suen CY (2002) Automatic recognition of handwritten numerical strings: a recognition and verification strategy. IEEE Trans Pattern Anal Mach Intell 24(11):1438–1454
37. Pakhira MK, Bandyopadhyay S, Maulik U (2004) Validity index for crisp and fuzzy clusters. Pattern Recognit 37(3):487–501
38. Pekalska E, Skurichina M, Duin RPW (2004) Combining dissimilarity-based one-class classifiers. international workshop on multiple classifier systems (MCS 2004), pp 122–133
39. Rabiner LR (1989) A tutorial on hidden markov models and selected applications in speech recognition. Proc IEEE 77(2):257–286
40. Rabiner LR, Juang BH (1993) Fundamentals of speech recognition. Prentice-Hall, Engelwood Cliffs
41. Ruta D, Gabrys B (2005) Classifier selection for majority voting. Int J Inf Fusion, pp 63–81
42. Schapire RE, Freund Y, Bartlett P, Lee WS (1998) Boosting the margin: a new explanation for the effectiveness of voting methods. Ann Stat 26(5):1651–1686
43. Seo J, Shneiderman B (2002) Interactively exploring hierarchical clustering results. IEEE Comput 35(7):80–86
44. Shipp CA, Kuncheva LI (2002) Relationships between combination methods and measures of diversity in combining classifiers. Int J Inf Fusion 3(2):135–148
45. Smyth P, Heckerman D, Jordan MI (1997) Probabilistic independence networks for hidden Markov probability models. Neural Comput 9:227–269
46. Wang X (1994) Durationally constrained training of HMM without explicit state durational. Proc Inst Phonetic Sci 18:111–130
47. Wolpert DH, Macready WG (1997) No free lunch theorems for search. In: IEEE transactions on evolutionary computation
48. Whitley D (2000) Functions as permutations: regarding no free lunch, walsh analysis and summary statistics. Parallel problem solving from nature (PPSN 2000), pp 169–178
49. Xie XL, Beni G (1991) A validity measure for fuzzy clustering. IEEE transactions of pattern analysis and machine intellegence, pp 841–847
50. Xu L, Krzyzak A, Suen CY (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans Syst Man Cybern 22(3):418–435
51. Xu L, Krzyzak A, Suen CY (1992) Methods of combining multiple classifiers and their applications to handwriting recognition. IEEE Trans Syst Man Cybern 22(3):418–435
52. Zouari H, Heutte L, Lecourtier Y, Alimi A (2004) Building diverse classifier outputs to evaluate the behavior of combination methods: the case of two classifiers. International workshop on multiple classifier systems (MCS 2004), pp 273–282

Author Biographies

Albert Hung-Ren Ko received M.Sc.A, degree in Artificial Intelligence and Pattern Recognition from the Universite Pierre et Marie Curie in 2002, and his Ph.D degree in Pattern Recognition from Ecole de Technologie Superieure, Universite du Quebec. His research interests are Ensemble Classification Methods, Small World Structure and Neural Networks.

Robert Sabourin received B.ing, M.Sc.A, Ph.D. degrees in Electrical Engineering from the Ecole Polytechnique de Montreal in 1977,1980 and 1991, respectively. In 1977, he joined the Physics Department of the Universite de Montreal where he was responsible for the design and development of scientific instrumentation for the Observatoire du Mont Megantic. In 1983, he joined the staff of the Ecole de Technologie Superieure, Universite du Quebec, Montreal, P.Q, Canada, where he is currently a professeur titulaire in the Departement de Genie de la Production Automatisee. In 1995, he also joined the Computer Science Department of the Pontificia Universidade Catolica do Parana (PUC-PR, Curitiba, Brazil) where he was coresponsible since 1998 for the implementation of a Ph.D. program in applied informatics. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI). His research interests are in the areas of handwriting recognition and signature verification for banking and postal applications.

Alceu de Souza Britto, Jr. received M.Sc. degree in Industrial Informatic from the Federal Center for Technological Education of Parana (Brazil) in 1996, and Ph.D. degree in Computer Science from Pontifical Catholic University of Parana (PUC-PR, Brazil). In 1989, he joined the Computer Science Department of the Ponta Grossa University (Brazil). In 1995, he also joined the Computer Science Department of the PUC-PR. His research interests are in the areas of document analysis and handwriting recognition.