A New HMM-Based Ensemble Generation Method for Numeral Recognition

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

LIVIA, ETS, University of Quebec
1100 Notre-Dame West Street, Montreal, Quebec, H3C 1K3 Canada

PPGIA, Pontifical Catholic University of Parana
Rua Imaculada Conceicao, 1155, PR 80215-901, Curitiba, Brazil.

{albert@livia.etsmtl.ca, robert.sabourin@etsmtl.ca,
alceu@ppgia.pucpr.br}

Abstract. A new scheme for the optimization of codebook sizes for 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.

Key words: Hidden Markov Models, Ensemble of Classifiers, Codebook Size, Clustering Validity Index, Pattern Recognition.

1 Introduction

The Hidden Markov Model (HMM) is one of the most popular classification methods for pattern sequence recognition, especially for speech recognition and handwritten pattern recognition problems [3, 6, 29, 30, 34]. The objective of the HMM is to model a series of observable signals, and it is this signal modeling ability that makes the use of HMM a better choice than other classification methods for recognition problems. 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 [3, 29, 34]. Each state emits some observation(s), according to a codebook setting 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.

However, there are some parameters that need to be optimized in HMM, such as the number of states in the model [35], the structure of the observation emission [6], the structure of the state transition [1, 14, 15], the order of the state transition [6, 29, 30] and the optimization of the codebook size [29, 30]. HMM codebook size optimization

is, in general, carried out by constructing a number of HMM classifiers and comparing their recognition rates on a validation data set. Given the extremely time-consuming process of HMM training, HMM codebook size optimization remains a major problem. But, since discrete symbols in HMM are usually characterized as quantized vectors in its codebook by clustering, the fitness of the codebook is directly related to the fitness of the clustering, for which a number of clustering validity indices have been proposed [2, 18, 19, 26, 27]. This means that codebook size can actually be optimized by using clustering validity indices.

Another important issue in the research concerning the HMM is that the ensemble of the HMM (EoHMM) emerges as a promising scheme to improve HMM [1,10–15]. This is because an ensemble of classifiers (EoC) is known to be capable of performing better than its best single classifier [25,31]. These classifiers can be generated by changing the training set, the input features or the parameters and architecture of the base classifiers [15]. The applicable ensemble creation methods include the Bagging, Boosting and Random Subspace methods. There may be other methods for the creation of HMM classifiers, based on the choice of features [13] for isolated handwritten images, and both column HMM classifiers and row HMM classifiers can be applied to enhance performance [4, 5]. The use of various topologies such as left-right HMM, semi-jump-in, semi-jump-out HMM [14], and circular HMM [1] can also be applied.

Because a data set usually consists of multiple levels of granularity [7, 21, 32], if clustering validity indices can give multiple optimized codebook sizes for HMM, then it is possible to construct EoHMMs based on different codebook sizes. This mechanism will give local optima of a selected clustering validity index. EoHMM are then selected by various objective functions and combined by different fusion functions. Because 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. The key questions that need to be addressed are the following:

1. Can the clustering validity index help in the selection of codebook sizes for optimizing HMM?
2. 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 applied the selected index for EoHMM construction (Fig. 1). We used the HMM-based handwritten numeral recognizer in [4, 5]. It is important to note that HMM optimization is a very complex task, and there are still a great many issues associated with it. In this paper we only deal with the problem related to HMM codebook size optimization, and 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 the used clustering index. Section 3 details the process of generation, selection and combination of HMM classifiers. In section 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.

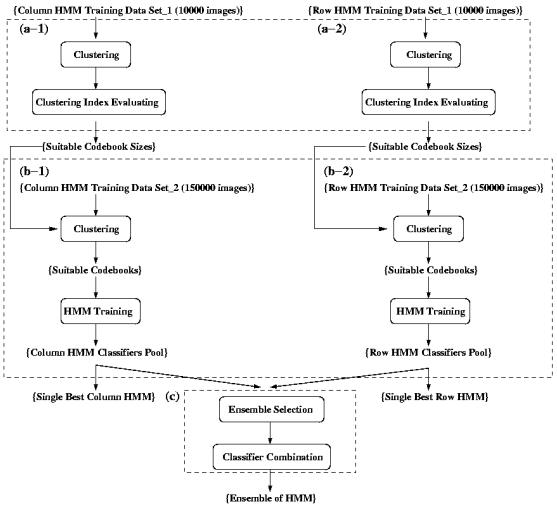

Fig. 1. The EoHMM classification system approach includes: (a) the adequate codebook sizes searching; (b) codebooks generation and HMM classifiers training (c) EoHMM selection and combination. Both (a) and (b) were carried out separately on column and row HMM classifiers.

2 Xie-Beni (XB) Index and 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 a measure to evaluate the results of clustering algorithms and give an indication of a partitioning that best fits a data set, and a clustering validity index is independent of clustering algorithms and data sets. We used XB index as the clustering index in this experiment.

XB index [2,18,19,27] 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

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

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

$$d_{min} = min_{i,j} \|v_i - v_j\| \tag{2}$$

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

$$XB = \frac{J_m}{N \times (d_{min})^2} \tag{3}$$

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 [36]. In order to obtain a group of potentially adequate codebook sizes, the clustering validity index used must have several local optima that can depict a data set at multiple levels of granularity [7, 21, 32]. 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. When HMM classifiers are trained with the same features and with the same samples, the distinction among the codebooks is the only possibility that results in diversity among classifiers and boosts the EoHMM performance.

The XB index is found to have this desirable property in our problem (Fig. 2). The plot of XB index values versus the numbers of clusters gives local optima for codebook sizes and are thus adequate for the construction of an EoHMM. The selected codebook sizes are used again for the clustering on all samples. We perform the experiment on a benchmark database 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 150000 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. For codebook size selection evaluated by clustering validity indices, due to the extremely large data set (150000 images are equivalent to 5048907 columns and 6825152 rows, with 47 features per column or per row), we use only the first 10000 images from the training data set to evaluate the quality of the clustering, and they are equal to 342910 columns and 461146 rows. 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 15000 samples was used to stop HMM classifiers training once the optimum had been achieved. The optimization set containing 15000 samples ($hsf\_\{0-3\}$) was used for GA searching for ensemble selection. To avoid overfitting during GA searching, the selection set containing 15000 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 60089 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. The features were extracted by the same means described in [4, 5], and K-Means was used for vector-quantization to generate codebooks

for the HMM. The number of HMM states was optimized by the method described in [35]. The HMMs were trained by Baum-Welch algorithm [29, 30]. The benchmark HMM classifiers used 47 features, with the codebook size of 256 clusters [4, 5]. 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%.

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. 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 clusters to 2048 clusters for the clustering task, we showed the relationship between the XB index and the number of clusters for column HMM features, and many local minima can be observed (Fig. $2(a)$). A similar tendency can be observed in row HMM features (Fig. $2(b)$). 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.

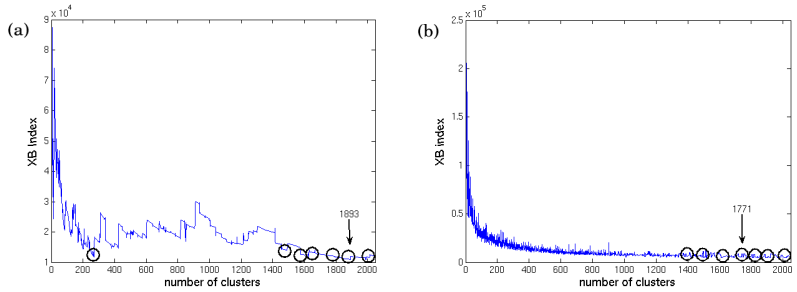

Fig. 2. The relationship between XB index and the number of clusters for: (a) HMM column features; (b) HMM row features. The circled areas indicate the places where the best 40 optima were found. The arrow indicates the smallest XB value with the respective number of clusters. Note that clusterings were carried out on the first 10000 images of the training data set.

3.2 Optimum Codebooks Selected by XB Index

Among all clusterings from 3 clusters to 2048 clusters, 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; 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.

3.3 Column-EoHMM and Row-EoHMM

We 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 1). The ensembles were then combined by the SUM rule [22, 38] and PCM-MAJ rule [24], since these two fusion functions have been shown to be very effective [22,24]. The ensemble of all HMM classifiers gave the best performance, given that the obvious diversity between the column HMM classifiers 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 1. Comparison of classification accuracies on test data set with two different fusion functions and on different types of EoHMMs. The number of classifiers is shown in parenthesis.

Fusion Functions → / EoHMM ↓	PCM-MAJ	SUM
COL-HMM (20)	**98.56 %**	98.55 %
ROW-HMM (20)	98.20 %	**98.26 %**
ALL-HMM (40)	**98.84 %**	98.78 %

3.4 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 [31]. We also tested 10 different compound diversity functions (CDFs) [23].

Table 2. 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 & 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)

These objective functions were evaluated by genetic algorithm (GA) searching. 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 various objective functions (MVE, SUM, 10 compound diversity functions [23]), and with 30 replications. A threshold of 3 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 [22, 38] and the PCM-MAJ rule [23]. Among all objective functions, the best ensemble was selected by the CDF 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.

Table 3. Best Performances from 30 GA replications on the test data set. The numbers of classifiers are noted in parenthesis. The PCM-MAJ was used as the fusion function in EoC.

Recognizers	Column HMM classifiers	Row HMM classifiers	Column & 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)

4 Discussion

In this work, we proposed to use the XB index in order to select various codebooks for the construction of Ensemble of HMMs. 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 [4,5]. 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. 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 by EoHMM based on various codebook sizes. 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 features and column features (Table 2 & Table 3).

The proposed method also 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 $M * S$ 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.

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 [36] and has been shown effective in clustering [2, 27]. 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 HMM classifiers 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. Arica N. and Vural F.T.Y. (2000), "A shape descriptor based on circular Hidden Markov Model," *In 15th Intl. Conf. on Pattern Recognition (ICPR00)*
2. Bandyopadhyay S. and Maulik U. (2001), "Non-parametric Genetic Clustering : Comparison of Validity Indices," *IEEE Transactions on Systems, Man and Cybernetics Part-C*, vol. 31, no. 1, pp. 120-125
3. Bengio Y. (1999), "Markovian Models for Sequential Data", *Neural Computing Surveys*, vol. 2, pp. 129-162
4. Britto Jr. A (2001), "A Two-Stage HMM-Based Method for Recognizing Handwritten Numeral Strings," *Ph.D. Thesis*, Pontifical Catholic University of Paraná
5. Britto A.S., Sabourin R., Bortolozzi F. and Suen C.Y. (2003), "Recognition of Handwritten Numeral Strings Using a Two-Stage Hmm-Based Method", *International Journal on Document Analysis and Recognition (IJDAR 5)*, no. 2-3, pp. 102-117
6. Dietterich T. G. (2002), "Machine Learning for Sequential Data: A Review.", *In Structural, Syntactic, and Statistical Pattern Recognition*, Lecture Notes in Computer Science, vol. 2396, pp. 15-30, Springer-Verlag
7. 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
8. Guenter S. and Bunke H. (2005), "Off-line cursive handwriting recognition using multiple classifier systems - on the influence of vocabulary, ensemble, and training set size," *Optics and Lasers in Engineering*, vol. 43, pp. 437-454
9. Guenter S. and 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
10. Guenter S. and Bunke H. (2003), "Off-line Cursive Handwriting Recognition - On the Influence of Training Set and Vocabulary Size in Multiple Classifier Systems," *Proceedings of the 11th Conference of the International Graphonomics Society*
11. Guenter S. and Bunke H. (2002), "A new combination scheme for HMM-based classifiers and its application to handwriting recognition," *Proc. 16th Int. Conference on Pattern Recognition*, Vol. II, 332 - 337
12. Guenter S. and Bunke H. (2002), "Generating classifier ensembles from multiple prototypes and its application to handwriting recognition," *Proceedings of the 3rd International Workshop on Multiple Classifier Systems*, pp. 179 - 188
13. Guenter S. and Bunke H. (2002), "Creation of classifier ensembles for handwritten word recognition using feature selection algorithms," *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pp. 183 - 188
14. Guenter S. and Bunke H. (2003), "Ensembles of Classifiers for Handwritten Word Recognition," *International Journal of Document Analysis and Recognition*, Volume 5, Number 4, 224 - 232
15. Guenter S. and Bunke H. (2003), "New Boosting Algorithms for Classification Problems with Large Number of Classes Applied to a Handwritten Word Recognition Task," *Proceedings of the 4th International Workshop on Multiple Classifier Systems*, pp. 326 - 335
16. Guenter S. and 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
17. Guenter S. and Bunke H. (2004), "Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm," *Electronic Letters of Computer Vision and Image Analysis*, vol. 3, no. 1, pp. 25 - 44

18. Halkidi M., Batistakis Y. and Vazirgiannis M. (2001), "On Clustering Validation Techniques," *Journal of Intelligent Information Systems*, vol. 17 , no. 2-3
19. Halkidi M., Batistakis Y. and Vazirgiannis M. (2002), "Clustering Validity Checking Methods: Part II," *SIGMOD Record*, vol. 31, no. 3, pp. 19-27
20. Huang X., Acero A. and Hon H. (2001), "Spoken Language Processing - a guide to theory, algorithm, and system development," *Prentice Hall PTR*
21. Johnson E. and Kargupta H. (1999), "Collective, hierarchical clustering from distributed, heterogeneous data", *In Large-Scale Parallel KDD Systems*, pp. 221–244
22. Kittler J., Hatef M., Duin R. P. W. and Matas J. (1998), "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239
23. Ko, A., Sabourin, R., Britto Jr, A., "Combining Diversity and Classification Accuracy for Ensemble Selection in Random Subspaces" , *IEEE World Congress on Computational Intelligence (WCCI 2006) - International Joint Conference on Neural Networks (IJCNN 2006)*, 2006.
24. Ko, A., Sabourin, R., Britto Jr, A., "Evolving Ensemble of Classifiers in Random Subspace" , *Genetic and Evolutionary Computation Conference (GECCO 2006)*, 2006.
25. Kuncheva L. I., Skurichina M., and Duin R. P. W. (2002), "An Experimental Study on Diversity for Bagging and Boosting with Linear Classifiers," *International Journal of Information Fusion*, vol. 3, no. 2, pp. 245-258
26. Maulik U. and Bandyopadhyay S. (2002), "Performance Evaluation of Some Clustering Algorithms and Validity Indices," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 24, no. 12, pp. 1650-1654
27. Pakhira M.K., Bandyopadhyay S. and Maulik U. (2004), "Validity index for crisp and fuzzy clusters," *Pattern Recognition*, vol. 37, No. 3, pp. 487-501
28. Pekalska E., Skurichina M. and Duin R. P. W. (2004), "Combining Dissimilarity-Based One-Class Classifiers," *International Workshop on Multiple Classifier Systems (MCS 2004)*, pp. 122-133
29. Rabiner L. R. (1989), "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, 77(2):257 – 286
30. Rabiner L. R. and Juang B. H. (1993), "Fundamentals of Speech Recognition," *Prentice-Hall*
31. Ruta D. and Gabrys B. (2005), "Classifier Selection for Majority Voting," *International Journal of Information Fusion*, pp. 63-81
32. Seo J. and Shneiderman B. (2002), "Interactively Exploring Hierarchical Clustering Results," *IEEE Computer*, vol. 35, no. 7, pp. 80-86
33. Shipp C. A. and Kuncheva L.I. (2002), "Relationships between combination methods and measures of diversity in combining classifiers," *International Journal of Information Fusion*, vol.3, no. 2, pp. 135-148
34. Smyth P., Heckerman D., and Jordan M. I. (1997), "Probabilistic independence networks for hidden Markov probability models", *Neural Computation*, vol. 9, pp. 227–269
35. Wang X. (1994), "Durationally constrained training of HMM without explicit state durational", *Proceedings of the Institute of Phonetic Sciences 18*, pp. 111-130
36. Xie X.L. and Beni G. (1991), "A validity measure for fuzzy clustering",*IEEE Transactions of Pattern Analysis and Machine Intelligence* , pp. 841-847
37. Xu L., Krzyzak A., and Suen C. Y. (1992), "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418-435
38. Xu L., Krzyzak A., and Suen C. Y. (1992), "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418-435