Classification of Time-Series Data Using a Generative/Discriminative Hybrid

K. T. Abou-Moustafa[1,2] **M. Cheriet[2]** **C. Y. Suen[1]**

[1] CENPARMI, Concordia University, Suite GM-606, 1455 de Maisonneuve W., Montreal, H3G 1M8, Canada
[2] LIVIA, Ecole de Technologie Superieure, 1100 Notre Dame W., Montreal, H3C 1K3, Canada

Emails: k_aboumo@cenparmi.concordia.ca, mohamed.cheriet@etsmtl.ca, suen@cenparmi.concordia.ca

Abstract

Classification of Time-Series data using discriminative models such as SVMs is very hard due to the variable length of this type of data. On the other hand generative models such as HMMs have become the standard tool for modeling Time-Series data due to their efficiency. This paper proposes a general generative/discriminative hybrid that uses HMMs to map the variable length Time-Series data into a fixed P-dimensional vector that can be easily classified using any discriminative model. The hybrid system was tested on the MNIST database for unconstrained handwritten numerals and has achieved an improvement of 1.23% (on the test set) over traditional 2D discrete HMMs.

Keywords: Generative Models, Discriminative Models, HMMs, SVMs.

1. Introduction

Classification of Time-Series (TS) data occurs in many pattern recognition applications such as speech recognition [1] and handwritten word recognition [2]. In these systems, generative models such as hidden Markov models (HMMs) [3] are used to represent these variable length sequences of vectors (for continuous models) or symbols (for discrete models), and then the classification is done using Bayes decision rule. Although in a previous work [4] we have discussed several limitations of HMMs especially when they are used for classification problems, yet HMMs are still the best modeling tool for TS data. However, for classification problems, a better solution would be to use discriminative models such as Support Vector Machines (SVMs) [5] and Multi Layer Perceptrons (MLPs), which are known for their good generalization for classification problems.

This paper targets the problem of increasing the performance of classifying TS data by introducing a new framework that combines the advantages of generative and discriminative models. Such a framework should have all the power of the two complementary approaches [6]. In a previous work [7], we have presented the hybrid combination and illustrated its validity with preliminary experiments in a constrained environment and the combination showed promising results. In this paper, the proposed approach in [7] is tested in a less restricted environment by incorporating it in a real life pattern recognition application for the recognition of unconstrained handwritten numerals. The main goal of this paper is to increase the validity of the proposed approach when used for real life applications.

The framework is composed of two stages, namely, 1) the modeling stage, and 2) the classification stage. For a P-class classification problem, the modeling stage is composed of P generative models (HMMs) that are used to map the TS input pattern into a single fixed sized P-dimensional vector (the likelihood score), that is the input of the second stage. The classification stage uses a discriminative model (SVM) to classify the vectors representing the TS patterns. The remainder of the paper is organized as follows. Section (2) reviews related work in the literature. Section (3) discusses the differences between generative and discriminative models, and section (4) presents the proposed framework. In section (5), experimental results on the MNIST database of handwritten numerals are provided to illustrate the advantage of the proposed framework. Finally, conclusions are drawn in section (6).

2. Related work

HMMs have become a standard method for modeling and classifying sequential data. Increasing the performance of HMM-based classifiers depends mainly on increasing the discrimination between the models of the classifier. In the literature, two approaches are followed: 1) improving learning algorithms, or 2) optimizing the model structure (the number of states and the topology). We mention in the following some of these algorithms.

Improving learning algorithms resulted in many training algorithms such as the Maximum Mutual Information (MMI) [8], Maximum a Posteriori (MAP) [9], the Entropy based distance functions algorithm [10] and Minimum Classification Error (MCE) [11, 12]. Optimizing HMM structure is another approach and it includes algorithms such as Bayesian model merging [13], model merging and splitting according to an a priori knowledge [14], optimizing the number of states using the bi-simulation technique [15], sequential pruning [16] and

Proceedings of the 9th Int'l Workshop on Frontiers in Handwriting Recognition (IWFHR-9 2004)
0-7695-2187-8/04 $20.00 © 2004 IEEE

model selection based on Discriminative Information Criterion (DIC) [17].

Despite of the several algorithms mentioned above, for learning, the Baum-Welch [3] and the Viterbi [18] are still the most popular training algorithms. As for the structure, still a predetermined topology and number of states is the common method used. This is due to 1) the computational cost of the new methods with respect to the increase in performance they provide, and 2) the a priori knowledge required by some of these algorithms may be available for applications such as speech recognition but may not be available for other applications.

A new approach that appeared recently in the machine learning community is the framework of generative and discriminative models. The first comparison between both approaches was introduced in [19] and recently addressed in [21] and [22]. The first formal combination appeared in [20] and it was later applied to speech recognition and speaker verification in [6] by extracting the Fisher Kernel form the generative models. The proposed framework in this paper is in general stimulated from [20] in that generative models are used to map the variable length sequential data into a single vector with a fixed size using the likelihood score instead of the Fisher score. Despite of the simpler combination method proposed, the framework improved the results of standard 2D discrete HMM results.

3. Generative vs. discriminative models

Choosing between discriminative and generative models is problem dependent. For a density estimation problem, generative models would be the best choice. However, for classification problems, discriminative models are preferred to generative ones due to their low asymptotic error.

A main reason for this choice is succinctly articulated [21] by Vapnik [23], "one should solve the classification problem directly and never solve a more general problem as an intermediate step such as modeling $\Pr(X \mid Y)$". Despite of the low error rate achieved by discriminative models in many classification problems, it was shown in [19] that learning discriminative models might not always lead to the best classifier. In addition, it is very difficult to classify sequential data using discriminative models due to their variable length. In the following, the advantages and disadvantages of generative and discriminative models are addressed from different perspectives [6]:

- *Target of learning and the classification rule:* Generative models learn a model of the joint probability $\Pr(X, Y)$, of the input X and the label Y. Their prediction is made by computing the likelihood $\Pr(X \mid Y)$ using Bayes rule and then picking the most likely Y. On the other hand, discriminative classifiers focus on modeling the decision boundaries between classes by modeling the posterior probability $\Pr(Y \mid X)$ directly or learning the direct map from input X to the class labels. Therefore, the focus of discriminative models is on correct classification only while generative models focus on modeling the true density of the data.
- *Learning method:* Generative models use reliable and efficient techniques for Maximum Likelihood (EM algorithm) [24] or Maximum A Posteriori estimation. The EM algorithm provably converges monotonically to a local maximum likelihood solution and typically outperforms gradient ascent methods [6] that are used for discriminative models. Also, during training, the EM algorithm needs less parameter tuning than gradient descent methods.
- *Modular learning:* For generative models, an independent model is built for each class where each model is trained individually and considers only the data whose labels correspond to it. Hence, the model does not interact with other classes and avoids considering the whole training set and consequently learning is simplified and the algorithm proceeds faster. Moreover, addition of a new class or deletion of an existing one is easier. Unlike generative models, discriminative models build a single model for all classes and hence it requires simultaneous consideration of all other classes which makes training harder. Discriminative models also involve iterative algorithms and may not scale well [19].
- *Missing data:* Unlike discriminative models, generative models are capable of learning even in the presence of some missing values. This is due to their learning method which optimizes the model over the whole dimensionality and thus models all the relationships between the variables in a more equal manner.
- *Rejection of poor or corrupted data:* The likelihood value obtained from generative models is more reliable than the posterior obtained from discriminative models, since generative models try to represent the true density of the data. A corrupted input or an outlier can be easily detected by the low likelihood and hence the design of a rejection rule is made easier.

4. The Proposed Framework

4.1 Notations

For complete references on HMMs and SVMs, the reader is required to read [3] and [5] respectively. The paper uses the basic compact notation of HMMs defined in [3] where $\lambda = (A, B, \pi)$, λ is the hidden Markov model, A is the transition probability matrix, B is the observation probability matrix and π is the initial state probability. In order to avoid any confusion to the reader, this subsection illustrates the notations that are going to be used in the following subsections:

- A training set of a TS data is defined by $\Psi = \{Z_i \mid 1 \le i \le N\}$ such that, $Z_i = \{z_t^i \mid 1 \le t \le T_i\}$, $z_t^i \in \Re^d$, T_i is the length of the sequence and N is the size of the training set.
- The pair $\{Z_i, Y_i\}$ represents the training example Z_i with the label Y_i such that $Y \in \mathrm{Y} = \{C_j \mid 1 \le j \le P\}$ where P is the number of classes.

- The function $F : \Re^d \rightarrow \Re^P$ (defined later) is a nonlinear function that takes Z_i as input and maps it to a point $X_i \in \Re^P$. Therefore the set $X = \{X_i \mid 1 \le i \le N\}$ is the image of the set Ψ under the function F.

It is worth noting that it is not a restriction that the dimensionality of X should be equal to the number of classes. It could be that due to several variations in the patterns within one class, the data could be represented using more than one model (such as in our case). Accordingly the number of models is going to be larger than the number of classes and X will have a dimensionality equal to the number of models.

4.2 The generative/discriminative framework

The above-mentioned advantages and disadvantages (section 3) led us to propose a new framework that combines the advantages of both models and overcomes the disadvantages of each separately. The framework is stimulated from [20] and it consists of two stages, namely 1) the modeling stage, and 2) the classification stage. Figure (1) shows a block diagram of the proposed framework.

The modeling stage is the first stage of the proposed framework and it consists of generative models. It has the basic role of mapping the variable length sequential pattern Z_i into a single fixed size vector. The basic idea for the modeling stage is as follows. For a P-class problem, each HMM is trained with a set of examples that belong to its class, however, when using the maximum likelihood decision rule to classify a new input pattern Z_0, each model HMM λ_j is given the input pattern Z_0 to compute the forward probability $\Pr_j(Z_0 \mid \lambda_j)$ [3] and the hope is always that the model of the correct class will output the highest likelihood. In the proposed framework, the modeling stage gets more information from all the models of the modeling stage in a P-dimensional real vector X (the likelihood score). In that sense, the modeling stage represents each sequential input as a point in the new space $\Re^P$, or more formally, it can be considered as the nonlinear function F mentioned in the previous subsection. To elaborate, consider the experiments presented in this paper. The problem tackled in this paper is a 10-class problem (handwritten digits). Each digit is modeled by two HMMs; a horizontal HMM that scans the digits from left-to-right, and a vertical HMM that scans the digit from top to bottom. Therefore the modeling stage consists of 20 HMMs, and hence the likelihood score has 20 variables (P=20), each one corresponds to one HMM.

The classification stage is the second stage of the proposed framework. It consists of a discriminative model that has the role of classifying the likelihood scores, the set X, representing the sequential patterns. The discriminative model could be a Multi Layer Perceptron (MLP) neural network, a SVM or any other discriminative model, however, we chose SVMs for their good generalization. In fact, the discriminative stage acts as an ordinary classifier and its input is the output of the modeling stage which acts as a feature extraction layer. Increasing the discrimination among generative models implies more discriminative feature vectors and consequently more accurate classification. Therefore, the modeling stage and the likelihood value are the key players of the framework. In the following, an insight of the likelihood value and the intuition lying behind the proposed model are elaborated in more details.

4.3 The likelihood score

Consider the P-class problem in hand, each class is represented by a single HMM, and that the data (training set and test set) are *i.i.d* drawn from the same unknown distribution and they exist in an Euclidean space S. The set of the P HMM models estimated from the training data form a set of local densities that allocate a certain part of the huge space S. Although, it is desired to have these densities far apart from each other in order to reduce the Bayes error, real life data (probably with noise and outliers) do not produce perfectly separated densities and ambiguities can exist easily. The likelihood score of the HMM measures the closeness of the pattern to the model itself, or how likely the model has generated this sequence. Consider the two classes C_j and C_i with the two sequences $Z_j \in C_j$ and $Z_i \in C_i$. For correctly trained models λ_j and λ_i, it should be that $\Pr(Z_i \mid \lambda_j) < \Pr(Z_j \mid \lambda_i)$ and the same for all other sequences that do not belong to C_j.

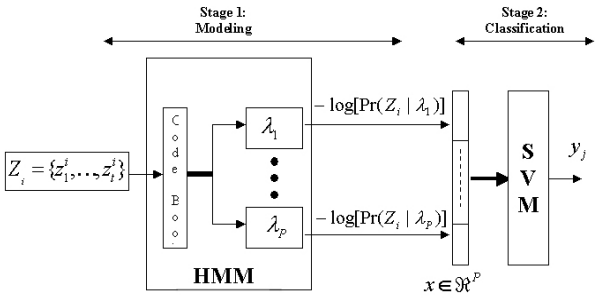

Figure 1: A block diagram of the proposed approach

In practice, likelihoods can be very close to each other and this closeness depends on the similarity between the two sequences. Therefore, the likelihood scores stored in X should have a high likelihood of the correct class and low likelihoods of other classes where each value represents how the sequence is close to its model. In case of similar patterns from different classes, the likelihoods will be close to each other which is a drawback of the proposed method. The proposed approach is stimulated from this observation. For an unknown pattern Z_0, each model votes (or scores) for this pattern and instead of considering the highest likelihood only as in traditional classification using HMMs, all the likelihoods are considered and taken as an input for a classifier that learns the voting of these models.

5. Experimental results

Recognition of unconstrained handwritten digits is an old yet a well-known problem in pattern recognition. Due to the extensive research done in this area, state-of-the-art techniques [26, 30, 31] were able to achieve very low error rates. However, the problem is considered as a standard for testing new classifiers, learning algorithms and feature sets.

5.1 The data set and features extraction

The MNIST database [25] was used in all the experiments. It is a very well known database for unconstrained handwritten digits that has a high variability in handwriting styles. The dataset has a training set of 60,000 samples and a test set of 10,000 samples from approximately 250 writers. The distribution of the digits form each class is almost uniform. For the experiments, the training set was divided to a new training set with 45000 samples and a validation set of 15000 samples. The two sets were created in a manner to keep the original distribution of each digit in the original training set. Before features extraction, all digits were cropped to be contained in the minimum bounding box. This is an important step in order to make all digits with different height and width, and hence the TS data extracted from each digit should have different observation length. The Time-Series data were extracted from the digits by using the features proposed in [27]. By following the same trend in [27], two different sets were extracted from the digits, the row based features and the column based featured.

5.2 The previous work

Before proceeding into more details and in order to avoid any confusion with our previous work [7], the differences between the current experiments and the previous ones will be highlighted briefly. The modeling stage in [7] consisted of 10 HMMs (horizontal HMMs) that scanned the digits from left to right only. The digits in the previous work had all the same height and length, and hence the TS data generated from the feature extraction had all the same length. Currently all the digits are cropped to be contained in the minimum bounding box. The features extracted in [7] were the row pixel values of the digits normalized to be from 0 to 1, and hence no complex features were used as in the current work. The validation set in [7] was taken from the test set (the first 5000 samples) of the MNIST database and not from the training set as in the current work.

5.3 Hidden Markov models

Two Discrete HMM-based classifiers (Horizontal HMM and Vertical HMM) were used in the experiments, one for each features set (column based and row based). Each HMM-based classifier consisted of ten models, one for each digit. The number of states and the codebook size for each HMM-based classifier were selected experimentally according to the best recognition rate obtained on the validation set. The best number of states for the H-HMM and V-HMM was 11 and 14 respectively, and the best codebook size for both HMM-based classifiers was found to be 1024. Table (1) shows the recognition results of the H-HMM and the V-HMM on the validation set and the test of the MNIST database.

Table 1: Recognition results of the H-HMM and the V-HMM on the validation and test sets

	H-HMM (%)	V-HMM (%)
Validation set	91.26	91.02
Test set	91.17	91.44

The H-HMM and the V-HMM represent the modeling stage of the proposed framework; therefore in order to improve the modeling capability of the modeling stage, both HMM-based classifiers were combined together (HV-HMM) by summing the *log* of the final probability obtained from the forward computation. First row of Table (2) shows the recognition results of the HV-HMM on the validation and the test set respectively.

Table 2: Recognition result of the HV-HMM and the proposed model on the validation and test sets

	Validation set (%)	Test set (%)
HV-HMM	92.63	92.85
Proposed	93.95	**94.08**

5.3 Support vector machines

The H-HMM and the V-HMM were used to model the training set and the validation set using the proposed method in Section (4). Each input pattern was passed to each HMM-based classifier to obtain the likelihood score from each model. Accordingly, each pattern was mapped to a 20-dimensional vector; i.e. the first 10 values are for the H-HMM, and second 10 values are for the V-HMM. The obtained vectors where then used to train the discriminative stage that is represented by SVM classifiers. For the experiments, we used the SVM Light V.5.0.[30] package. The discriminative stage consisted of 10 SVMs using a one-against-all strategy. The kernel function was selected to be the RBF kernel and the sigma parameter for the kernel was adjusted using the method proposed in [35]. The SVM training was stopped when the highest recognition rate was achieved on the validation set. The constant parameter C was set to 10 in all the experiments. The second row of Table (2) shows the recognition results of the SVM classifier on the validation set and the test set. Tables (3) and (4) show the confusion matrix for the HV-HMM and proposed framework respectively.

5.4 Results' analysis

It can be seen from Table (1) how the results of the H-HMM and the V-HMM are low in general and very close to each other. A direct possible reason for that is the size of the image. The original size of the image is 28x28 (pixels) including a white border (4-5 pixels) on each side and after cropping, it becomes less than that. The features proposed in [27] were tested on the NIST SD19 database where the images are usually bigger, hence scaling the images will probably give better results. Although the results of the V-HMM and the H-HMM are close to each other, yet the combination boosted the results by more than 1.3% for each classifier. This explicitly implies that both classifiers complement each other.

Table 3: Confusion matrix for the HV-HMM classifier (%)

	0	1	2	3	4	5	6	7	8	9
0	-	-	-	11.9	2.3	7.1	11.9	-	64.3	2.3
1	-	-	25.5	-	23.2	2.3	16.2	2.3	25.5	4.6
2	4.4	1.4	-	53.7	1.4	8.9	1.4	10.4	10.4	7.4
3	2.9	-	17.3	-	-	33.3	-	20.2	17.3	8.7
4	-	2.2	11.1	2.2	-	4.4	2.2	8.9	26.67	42.2
5	1.0	1.0	2.1	64.2	2.1	-	3.1	2.1	21.0	3.1
6	3.8	5.0	11.3	-	16.4	40.5	-	-	22.7	-
7	-	2.5	26.5	3.8	12.6	5.0	-	-	13.9	35.4
8	13.3	-	10.4	13.3	9.5	12.3	9.5	7.6	-	23.8
9	1.1	4.4	2.2	29.6	16.4	4.4	-	13.1	28.5	-

Table 4: Confusion matrix for the hybrid model (%)

	0	1	2	3	4	5	6	7	8	9
0	-	-	2.78	16.6	2.78	8.33	19.44	-	47.2	2.78
1	-	-	33.3	-	18.1	3.0	18.8	3.0	18.1	6.0
2	4.4	1.4	-	52.9	1.4	1.4	2.9	17.6	10.2	7.3
3	1.3	-	17.3	-	-	38.6	-	16.0	20.0	6.6
4	-	5.5	9.2	3.7	-	1.8	5.5	9.2	12.9	51.8
5	1.1	3.5	1.1	54.1	4.7	-	3.5	3.5	24.7	3.5
6	6.9	6.9	15.5	-	17.2	27.6	-	-	25.8	-
7	-	3.9	28.9	3.9	11.8	5.2	-	-	7.8	38.1
8	17.3	-	12.5	8.6	10.5	9.6	10.5	8.6	-	22.1
9	1.0	5.4	1.0	30.1	17.2	4.3	-	16.1	24.7	-

Table (2) shows how the results of the proposed model overcome the results on the HV-HMM on the validation set and the test set by more than 1.2% for each set. Despite of the low performance of the HMMs in the modeling stage, yet the discriminative stage was able to enhance the overall classification of the system. By looking to Tables (3) and (4), one can see how the increase in performance, although seems to be promising, yet it shows interesting observation on the capability of the proposed model. It is expected that the error in all cases in Table (4) should be less that those in Table (3), however, this is not the case. There are cases where the error in Table (4) is slightly higher than or equal to that of Table (3), but when it comes to the increase achieved by the hybrid system in cases such (0,8), (1,4) and others (highlighted in grey), the error dramatically decreases down. Therefore the total of these ups and downs made the proposed model ahead. This in turns questions two issues; 1) how informative is the value of the likelihood obtained from the HMMs, and 2) how the hybrid will behave in case of an unbalanced training set.

For the first issue, it is clear that the likelihood value (which is a sum of the probabilities of the alpha computation) is not so informative and more statistical information can be obtained from the states of the model. For the second issue, it is known that generative models do not exploit prior probabilities of each class while discriminative models do. Hence, this factor will affect the modeling capability of the modeling stage and consequently the overall hybrid.

6. Conclusion and future work

This paper increases the validity of a previously proposed framework [7] that combines generative and discriminative models for the classification of variable length Time-Series data. The framework is composed of 1) a modeling stage that has the role of mapping the variable length Time-Series data in to a fixed size vector, and 2) a discriminative stage that has the role of classifying the output of the modeling stage. The framework was able to improve the results of a two dimensional discrete HMM by more than 1.2%.

The accuracy of the framework depends mainly on the modeling capability of the HMMs in the modeling stage, and on the type of information extracted from these models that represent the variable length Time-Series data. Increasing the discrimination between the HMMs will help the discriminative stage to produce better decision boundaries between classes. Such an improvement can be achieved by using mixtures of generative models, training the HMM discriminatively using MCE [11, 12], or training the HMMs and the SVMs simultaneously. As for the discriminative stage, testing other strategies for combining SVMs might bring better performance. For real life applications, neural networks can replace SVMs since they have a faster testing time compared to SVMs. The framework in general can accept many modifications for improvements and in this paper encourages for future research work in this direction.

Acknowledgment

The authors would like to thank NSERC of Canada and FCAR of Quebec for their financial support.

References

[1] K-F. Lee, "Automatic Speech Recognition: The Development of the SPHINX System", Kluwer Academic Press, 1999.

[2] A. El-Yacoubi, M. Gilloux, R. Sabourin and C. Y. Suen, "An HMM Based Approach for Off-line Unconstrained Handwritten Word Modeling and Recognition," IEEE Trans. PAMI, Vol. 21, No. 8, pp. 752-760, 1999.

[3] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Application in Speech Recognition," Proc. of IEEE, Vol. 77, No. 2, pp. 257-286, 1989.

[4] K. T. Abou-Moustafa, M. Cheriet and C. Y. Suen, “On The Structure of Hidden Markov Models,” Pattern Recognition Letters, Vol. 25, pp. 829-973, June 2004.

[5] N. Cristianin and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," Cambridge Univ. Press, 2000.

[6] L. Quan and S. Bengio, “Hybrid Generative-Discriminative Models for Speech and Speaker Recognition,” IDIAP Tech. Report, Mar. 2002.

[7] K. T. Abou-Moustafa, M. Cheriet and C. Y. Suen, “A Generative-Discriminative Hybrid for Sequential Data Classification,” IEEE ICASSP 2004, pp. V 805 – V 808.

[8] L. Bahl, P. Brown, P. de Souza and R. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model Parameters for Speech Recognition," Proc. of ICASSP, Tokyo, pp. 49-52, 1986.

[9] J-L. Gauvain and C-H. Lee, "MAP Estimation of Continuous Density HMM: Theory and Applications," Proc. of DARPA Speech & Nat. Lang. Processing, Feb. 1992.

[10] Y. Singer and M. Warmuth, "Training Algorithms for Hidden Markov Models Using Entropy Based Distance Functions," Proc. of Advances in Neural Information Processing Systems 9, pp. 641-647, 1996.

[11] A. Biem, "Minimum Classification Error Training for Online Handwritten Word Recognition," Proc. of 8th IWFHR, Niagra-on-the-lake, pp. 61-65, 2002.

[12] L. Saul and M. Rahim, "Maximum Likelihood and Minimum Classification Error Rate Factor Analysis for Automatic Speech Recognition," IEEE Trans. Speech and Audio Processing, Vol. 8, No. 12, pp. 115-125, 2000.

[13] A. Stolcke and S. Omuhundro, "Hidden Markov Model Induction by Bayesian Model Merging,” Advances in Neural Information Processing 5, Morgan Kaufmann, S. Hanson, J. Cowan and C. Giles, editors, pp. 11-18, 1992.

[14] T. Brants, “Estimating Markov Model Structures,” Proceedings of the Fourth Conference on Spoken Language Processing (ICSLP), Philadelphia, PA, 1996.

[15] M. Bicego, A. Dovier and V. Murino, "Designing the Minimal Structure Hidden Markov Model by Bisimulation," in Energy Minimization Methods in Computer Vision and Pattern Recognition, Springer, M. Figueiredo, J. Zerubia and A. K. Jain, editors, pp. 75-90, 2001.

[16] M. Bicego, V. Murino and M. Figueiredo, "A Sequential Pruning Strategy for the Selection of the Number of States in Hidden Markov Models, Pattern Recognition Letters, Vol. 24, pp. 1395-1407, 2003.

[17] A. Biem, "A Model Selection Criterion for Classification: Application to HMM Topology Optimization," Proc. 17th ICDAR, Edinburgh, U.K, pp. 104-108, 2003.

[18] A. J. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm," IEEE. Trans. Information Theory, Vol. 13, No. 4, pp. 179-190, 1983.

[19] Y. Rubenstein and T. Hastie, “Discriminative vs Informative Learning,” Proc. of Knowledge Discovery and Data Mining, 1997.

[20] T. Jaakkola and D. Haussler, "Exploiting Generative Models in Discriminative Classifiers, Proc. of Advances in Neural Information Processing (NIPS) 11, 1998.

[21] A. Ng and M. Jordan, "On Generative vs. Discriminative Classifiers: A Comparison of Logistic Regression and Naive Bayes," Proc. of Advances in Neural Information Processing 15, 2002.

[22] G. Bouchard, "The Trade-off Between Generative and Discriminative Classifiers," Proc. of Advances in Neural Information Processing (NIPS) 16, 2003.

[23] V. N. Vapnik, "Statistical Learning Theory," John Wiley & Sons, 1998.

[24] A. Dempster, N. Laird and D. Rubin, "Maximum-likelihood from Incomplete Data Via the EM Algorithm," J. Royal Stat. Soc. Ser. B., Vol. 39, pp. 1-38, 1977.

[25] Y. LeCun, "The MNIST Database of Handwritten Digits," http://yann.lecun.com/exdb/mnist.

[26] C-L. Liu, K. Nakashima, H. Sako and H. Fujisawa, "Handwritten Digit Recognition: Benchmarking of State-of-the-art Techniques," Pattern Recognition, Vol. 36, pp. 2271-2285, 2003.

[27] S. de Britto, R. Sabourin, F. Bortolozzi and C. Y. Suen, “Complementary Features Combined in an HMM-based System to Recognize Handwritten Digits," Proc. of 12th Intl. Conf. on Image Analysis and Processing, pp. 670-675, 2003.

[28] T. Joachims, "Making Large-Scale SVM Learning Practical,” Advances in Kernel Methods and Support Vector Learning, MIT Press, B. Scholkopf, C. Burges and A. Smola, editors, 1999.

[29] L-N. Teow and K-F. Loe, "Robust Vision-based and Classification Schemes for Off-line Handwritten Digit Recognition,” Pattern Recognition, Vol. 35, No. 11, pp. 2355-2364, 2002.

[30] P. Simard, D. Steinkraus and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," Proc. of 17th ICDAR, Edinburgh, U.K, pp. 965-962, 2003.

[31] J. Dong, "Speed and Accuracy: Large-scale Machine Learning Algorithms and Their Applications," CENPARMI, Department of Computer Science, Concordia University, Montreal, Canada, 2003.

[32] N. Cristianini, C. Campbell and J. Shawe-Taylor, “Dynamically Adapting Kernels in Support Vector Machines,” Neural Information Processing Systems, pp. 204-210, 1998.