A GENERATIVE-DISCRIMINATIVE HYBRID FOR SEQUENTIAL DATA CLASSIFICATION

*K. T. Abou-Moustafa, C. Y. Suen**

CENPARMI, Dept. of Computer Science
Concordia Univ., GM-606,
1455 de Maisonneuve, Montreal,
H3G 1M8, QC, Canada

*M. Cheriet**

LIVIA, Dept. of Automation Engineering
Ecole de Technologie Superieure,
Univ. of Quebec, 1100 Notre-Dame W.,
Montreal, H3C 1K3, QC, Canada

ABSTRACT

Classification of Sequential 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 representing sequential data due to their efficiency. This paper proposes a general generative-discriminative framework that uses HMMs to map the variable length sequential data into a fixed size P-dimensional vector (likelihood score) that can be easily classified using any discriminative model. The preliminary experiments of the framework on the MNIST database for handwritten digits have achieved a better recognition rate of 98.02% than that of standard HMMs (94.19%).

1. INTRODUCTION

Classification of sequential 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. However, for classification problems, a better solution would be to use discriminative models such as Support Vector Machines (SVMs) 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 sequential 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 [4]. The framework is composed of two stages, namely, 1) the modelling stage, and 2) the classification stage. For a P-class classification problem, the modelling stage is composed of P generative models (HMMs) that are used to map the sequential 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 sequential patterns. The remainder of the paper is organized as follows. Section (2) reviews related work in the literature. Section (3) gives a brief introduction on SVMs. Section (4) discusses the difference between generative and discriminative models, and section (5) presents the proposed framework. In section (6), experimental results on the MNIST database of handwritten digit are provided to illustrate the advantage of the proposed framework. Finally, conclusions are drawn in section (7).

*The authors wish to thank NSERC of Canada and FQAR of Quebec for their financial support

2. RELATED WORK

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 the parameter estimation algorithms which resulted in several algorithms such as: Maximum Mutual Information (MMI) [5], Maximum A Posteriori (MAP) [6] and Minimum Classification Error [7], or 2) optimizing the model structure using Bayesian model merging [8], model merging and splitting according to an a priori knowledge [9], and model selection based on Discriminative Information Criterion (DIC) [10]. A new approach that appeared recently in the machine learning community is the combination of generative and discriminative models for data classification. It was shown theoretically and experimentally in [11, 12, 13] that the combination of both models will combine the complementary power of both models. A first formal combination appeared in [14]. The idea was based on extracting discriminative features using generative models and then incorporating these features in discriminative models. The idea was more suitable for kernel methods and it was applied by extracting a kernel function (Fisher Kernel) from generative probabilistic models. The Fisher kernel was recently applied to speech recognition and speaker verification in [4] where the probabilistic

generative models were HMMs. The proposed framework in this paper is in general stimulated from [14] in that generative models are used to map the variable length sequential data into a single fixed size vector using the likelihood score instead of the Fisher score. Despite of the simpler combination method proposed, the framework boosted the results of standard HMM results by 3.83% which shows the potential of the generative-discriminative trend.

3. SUPPORT VECTOR MACHINES

A Support vector machine is a binary pattern classifier based on a new statistical learning approach proposed by Vapnik [15]. SVMs have shown a great ability in generalization for classification problems. The basic idea consists of mapping the space $S = \{X\}$ of the input examples into a very high dimensional (probably infinite) feature space. By choosing an adequate mapping, the input examples become almost linearly separable in the feature space. The optimization criterion of the SVM classifier is the width of the margin between the classes, i.e. the area around the decision surface defined by the distance to the nearest training examples in the feature space. These examples are called the support vectors because they support the decision boundary, and they define the decision function of the support vector machine. The optimization of a support vector machine consists of minimizing the number of support vectors by maximizing the margin between the two classes. The decision function derived by the SVM classifier for a two class problem can be formulated using a kernel function $K(X_0, X_i)$ of a new example X_0 (to be classified) and a member of the support vector set X_i as follows:$F(X_0) = \sum_{X_i \in SV} \alpha_i Y_i K(X_0, X_i) + \alpha_0$ where SV is the set of support vectors, $Y_i \in \{-1, +1\}$ is the label of the support vector X_i and $\alpha_i \geq 0$. The parameters α_i are optimized during the training process. There are many kernel functions $K(.,.)$, the simplest one is the dot product between the input pattern to classify X_0 and a member of the support vectors set $((K(X, X_i) = XX_i)$, which derives a linear classifier. Nonlinear SVM classifier such as Gaussian radial basis functions SVM or polynomial SVM classifier can be derived by an RBF kernel $(K(X, X_i) = exp(||X - X_i||^2)/\sigma^2))$ or by using p-th order polynomial kernel $(K(X, X_i) = (XX_i + 1)^p)$ functions respectively. The interested reader can find more details on SVMs in [16, 15].

4. GENERATIVE VS DISCRIMINATIVE MODELS

In the following, we present a comparison between generative and discriminative models from the following perspectives [4]:

Target of learning and the decision 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 $Pr(X|Y)$ using Bayes rule and then picking the most likely Y. On the other hand, discriminative classifiers model the decision boundaries between classes by computing the posterior probability $Pr(Y|X)$ directly or learning the direct map from input X to the class labels. Therefore, discriminative models are only concerned with correct classification.

Learning method: Generative models use efficient and easy techniques for Maximum Likelihood or Maximum A Posteriori estimation such as the EM algorithm. The EM algorithm provably converges monotonically to a local maximum likelihood solution but it requires certain model assumptions that if are incorrect, the resulting model will be biased. In addition, these algorithms can perform well even in the presence of missing values [4]. For discriminative models, parameter estimation is more flexible and robust since it has less assumptions on the model.

Modular learning: For generative models, an independent model is built for each class where each model is trained individually on its own data set. 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. Unlike generative models, learning discriminative models requires simultaneous consideration of all the data from all classes which makes training harder, involve iterative algorithms and do not scale well [11].

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. Hence, corrupted inputs or outliers can be easily detected by the low likelihood and consequently the design of a rejection rule is made easier.

5. THE PROPOSED FRAMEWORK

The above mentioned advantages and disadvantages led us to propose a new framework that combines the advantages of both models and overcomes the disadvantages of each separately. The framework consists of two stages, namely 1) the modelling stage, and 2) the classification stage. Figure 1 shows a block diagram of the proposed framework.

The modelling stage is the first stage of the proposed framework and it consists of generative models (HMMs in in this context). It has the basic role of mapping the sequential pattern Z_i into a single fixed size vector $X_i \in \Re^P$. The basic idea for the modelling stage is as follows. For a P-class problem, each HMM is trained with a set of examples that belongs to its class. However, when using the maxi-

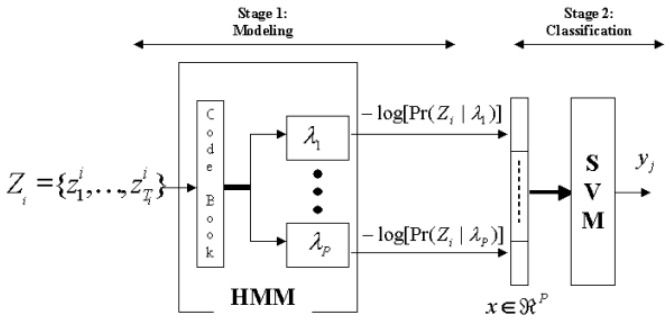

Fig. 1. A block diagram of the proposed framework

mum likelihood decision rule to classify a new input pattern Z_0, each model λ_j is given the input pattern Z_0 to compute the forward probability $Pr_j(Z_0|\lambda_j)$ [3] and the hope is always that the model of the correct class will output the highest likelihood. In the proposed framework, the modelling stage gets more information from all the models of the modelling stage in a P-dimensional real vector X (the likelihood score). In that sense, the modelling stage represents each sequential input as a point in the new space $\Re^P$, or more formally, it can be considered as a nonlinear mapping function $\mathcal{F}$ such that $\mathcal{F} : \Re^D \rightarrow \Re^P$.

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 representing the sequential patterns. In fact, the discriminative stage acts as an ordinary classifier and its input is the output of the modelling stage which acts as a feature extraction layer. Increasing the discrimination between the generative models implies more discriminative feature vectors and consequently more accurate classification.

To elaborate more the idea behind the framework, consider the simple classification problem with $P-$classes, where 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 a space $\mathcal{S}$. The set of P models estimated form the training data form a set of local densities that allocate a certain part in the space $\mathcal{S}$. Although it is desired to have these densities far apart from each other in order to reduce the Bayes error, real life data (with noise, outliers and similarity between classes in some cases) do not produce perfectly separated densities and ambiguities and overlaps can exist easily. Recall that the densities can be close to each other, 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. The proposed approach is stimulated from this observation. That is, the likelihood scores obtained from the different HMMs should have a high score from the correct class, and low scores from other classes where each low score represents how likely this model has generated this pattern. In other words, each model votes for the input pattern and instead of considering the maximum vote, all votes are considered and taken as an input for a classifier that learns the voting of these models. Therefore, when the maximum likelihood misses the correct class, the second stage classifier can recover some of the errors by using the information from other models. Apparently, the approach can be considered as a classifier combination scheme.

6. EXPERIMENTAL RESULTS

We designed a simple prototype for the proposed framework to conduct some preliminary experiments on the recognition of unconstrained handwritten digits. The experiments were conducted on the MNIST database [17], one of the very well known databases for unconstrained handwritten digits that has a training set of 60,000 samples and a test set of 10,000 samples from approximately 250 writers. The digits are cropped and scaled to be contained in a 20x20 pixels images. The gray level values of the images were normalized to be from 0 to 1. To extract the sequential data from these images, we used a sliding window with 3 pixels width, 20 pixels height and an overlap of 2 pixels between successive windows that scanned the image from left to right. A feature vector is extracted from each window by computing the average gray level value in each row of the window, i.e. the sum of gray level pixels in each row divided by the window width. This resulted in an observation sequence length of 18 vectors from each image. The HMMs of the modelling stage consisted of 10 discrete HMMs. Each model had 10 states with a simple left-to-right topology with self-state transition. Three codebooks of size 1024, 512 and 256 were constructed using the K-Means clustering algorithm. In order to overcome the problem of initialization of the K-Means, the algorithm was run using 10 different initializations. Since the output probabilities of the models are usually very small, the negative log of the output probabilities were stored instead. For the classification stage, the package of SVM^{Light} V 5.00 [18] was used as to construct the discriminative models. The stage consisted of 10 SVM models (one against all strategy) with a Gaussian kernel. The constant parameter C of the kernel was fixed to 10 and the gamma parameter [18] of the kernel was adjusted until the minimum error rate could be achieved on a predefined validation set. Table 1 shows a comparison between the results obtained from the HMM-based classifier only (2^{nd} column) on the test set with different codebooks and the results obtained from the proposed framework. It can be seen from Table 1 how the framework significantly boosted the results of the standard HMMs.

Table 1. Comparison of performance of the HMMs and the proposed framework on the test set.

CodeBook size	*HMMs*	*Proposed framework*
256	93.53 %	97.8 %
512	93.97 %	97.89%
1024	94.19%	98.02%

7. CONCLUSION & FUTURE WORK

We introduced a new framework that combines generative and discriminative models for the classification of sequential data. The framework outperformed standard HMMs by 3.83%, when tested on the MNIST database for handwritten digits which illustrates the advantages of the framework. The performance of the framework depends mainly on the modelling capability of the HMMs in the modelling stage. Increasing the discrimination between the HMMs will help the discriminative stage to have better decision boundaries between classes. Such an improvement can be achieved by using mixtures of generative models and training the models using MCE [7]. As for the discriminative stage and for real life applications such as speech recognition, neural networks can replace SVMs for two reasons; 1) despite that speeding up the test time of SVMs is currently a hot research topic in machine learning, yet, MLPs have a faster testing time compared to SVMs, and 2) the output of MLPs is more meaningful since it can be considered as a posteriori probability of the input pattern, unlike the output of SVMs which is the distance of the input pattern from the margin of the classifier.

8. 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] L. Quan and S. Bengio, "Hybrid generative-discriminative models for speech and speaker recognition," Tech. Rep., IDIAP, March 2002.

[5] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. of of ICASSP, Tokyo*, 1986, pp. 49–52.

[6] J-L. Gauvain and C-H. Lee, "Map estimation of continuous density hmm: Theory and applications," in *Proc. of DARPA Speech & Nat. Lang. Processing*, Feb. 1992.

[7] L. Saul and M. Rahim, "Maximum likelihood and minimum classification error rate factor analysis for automatic speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 8, no. 2, pp. 115–125, 2000.

[8] A. Stolcke and S. Omuhundro, "Hidden markov model induction by bayesian model merging," in *Advances in Neural Information Processing 5*, S. Hanson, J. Cowan, and C. Giles, Eds., pp. 11–18. Morgan Kaufmann, 1992.

[9] T. Brants, "Estimating markov model structures," *Proc. of ICLSP Philadelphia, PA*, 1996.

[10] A. Biem, "A model selection criterion for classification: Application to hmm topology optimization," in *Proc. 17th ICDAR, Edinburgh, U.K*, 2003, pp. 104–108.

[11] Y. Rubenstein and T. Hastie, "Discriminative vs informative learning," in *Proc. of Knowledge Discovery and Data Mining*, 1997.

[12] A. Ng and M. Jordan, "On generative vs. discriminative classifiers: A comparison of logistic regression and naive bayes," in *Proc. of Advances in Neural Information Processing 15*, 2002.

[13] G. Bouchard, "The trade-off between generative and discriminative classifiers," in *Proc. of Advances in Neural Information Processing 16*, 2003.

[14] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Proc. of Advances in Neural Information Processing 11*, 1998.

[15] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, Sussex, England, 1998.

[16] N. Cristianin and J. Shawe-Taylor, *An Introduction to support vector machines and other kernel-based learning methods*, Cambridge Univ. Press, 2000.

[17] Y. LeCun, "The mnist database of handwritten digits," `http://yann.lecun.com/exdb/mnist`.

[18] T. Joachims, "Making large-scale svm learning practical," in *Advances in Kernel Methods - Support Vector Learning*, B. Scholkopf, C. Burges, and A. Smola, Eds. MIT Press, 1999.