MAXIMUM MARGIN TRAINING OF GAUSSIAN HMMS FOR HANDWRITRING RECOGNITION

Trinh-Minh-Tri Do and Thierry Artières
LIP6 - Université Pierre et Marie Curie
104 avenue du président Kennedy, 75016 Paris France
Trinh-Minh-Tri.Do@lip6.fr, Thierry.Artieres@lip6.fr

Abstract

Recent works for learning Hidden Markov Models in a discriminant way have focused on maximum margin training, which remains an open problem due to the lack of efficient optimization algorithms. We developped a new algorithm that is based on non convex optimization ideas and that may solve maximum margin learning of GHMMs within the standard setting of partially labelled training sets. We provide experimental results on both on-line handwriting and off-line handwriting recognition.

***Index Terms*—** Large margin HMMs, On-line handwriting recognition, Off-line handwriting recognition

1. INTRODUCTION

Gaussian Hidden Markov Models (GHMMs) have been widely used for automatic speech recognition [9] and handwriting recognition. GHMMs are traditionnally learnt to maximize the joint likelihood of observation sequences and of state sequences (MLE) via an EM algorithm. Training is performed based on a partially labelled training set, that is a set of observation sequences together with the corresponding character sequences. This is the usual setting where one never gets the complete sequence of states corresponding to an observation sequence in the training stage. In test, segmentation is performed through Viterbi decoding that maps an observation sequence into a state sequence. Based on the underlying semantics of the states (e.g. passing through the three states of the left-right HMM corresponding to a particular character means this character has been recognized), the sequence of states translates into a sequence of labels.

This typical use of HMMs is very popular since it is both simple and efficient, and it scales well with large corpus. However, this learning strategy does not focus on what we are primarily concerned with, namely minimizing the classification (or the segmentation) error rate. Hence, a number of attempts have been made to develop discriminative learning methods for HMMs. First studies, in the speech recognition field, aimed at optimizing a discriminative criterion such as the Minimum Classification Error (MCE) [6] or the Maximum Mutual Information (MMI) criterion [8]. Other approaches have been proposed in the handwriting recognition field with the use of Conditional Random Fields ([2]), MMI and other derived criterion ([17]).

Recently, large margin learning of HMMs has been investigated in the speech recognition community [5, 4, 11], (see [16] for a review). This formalism has been shown (Cf. [11]) to significantly outperform traditional discriminant methods such as Conditional Maximum Likelihood (CML) and Minimum Classification Error (MCE) on continuous speech recognition tasks. Yet none of the already proposed large margin training algorithms actually handle the full problem of max-margin learning for HMM parameters in the standard partially labelled setting.

We describe here a new method for maximum margin learning of GHMM models in the usual unlabelled setting and study its efficiency on both on-line handwritten digit recognition and on off-line handwritten word recognition tasks. First we detail the formulation of maximum margin learning of GHMM and describe previous attempts for such a learning scheme. Then we present our method and we provide experimental results showing its potential on two handwriting recognition tasks.

2. LARGE MARGIN TRAINING FOR GHMMS

After introducing our HMM notations we discuss the definition of a margin criterion for HMMs and detail how previous works have proposed to solve the training problem using such a criterion.

2.1. Notations

We consider a training set that consists of K input-output pairs $(\mathbf{x}^1, \mathbf{y}^1), ..., (\mathbf{x}^K, \mathbf{y}^K) \in \mathcal{X} \times \mathcal{Y}$ where $\mathbf{x}^i =$

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.221

$(x_1^i, x_2^i, ..., x_T^i) \in (\mathcal{R}^d)^T$ stands for an observation sequence whose correct label sequence is $\mathbf{y}^i$. We consider the standard setting in handwriting recognition where the label information for the i^{th} sample (e.g. a word) does not include the full state sequence $\mathbf{s}^i = (s_1^i, ..., s_T^i)$. For a T-lengthed observation sequence $\mathbf{x}^i$, $\mathbf{y}^i$ consists in the sequence of letters (of length $L \leq T$) in the handwritten word. In particular the frame index corresponding to boundaries between letters are unknown.

We consider standard CDHMMs with Gaussian mixture as probability density function (pdf) within each state. The pdf over observation frame $x \in \mathcal{R}^d$ within a state s is defined as a Gaussian mixture:

$$p(x|s) = \sum_{\mathbf{m}=1}^{M} P_{s,m} \mathcal{N}_{s,m}(x) \qquad (1)$$

where $P_{s,m}$ stands for the prior probability of the m^{th} mixture in state s, and $\mathcal{N}_{s,m}$ stands for the Gaussian distribution whose mean vector is noted $\mu_{s,m}$ and whose covariance matrix is noted $\Sigma_{s,m}$.

$$\mathcal{N}_{s,m}(x) = \frac{1}{\Pi^{\frac{d}{2}} |\Sigma_{s,m}|^{\frac{1}{2}}} \exp -\frac{1}{2}(x - \mu_{s,m})^{\top} \Sigma_{s,m}^{-1} (x - \mu_{s,m})$$

A N states HMM is defined with a set of parameters that we note $\mathbf{w}$, where $\mathbf{w} = \{\mathbf{\Pi}, \mathbf{A}, \boldsymbol{\mu}, \mathbf{\Sigma}\}$. Using standard notations, $\mathbf{\Pi}$ stands for the initial state probabilities, $\mathbf{A}$ for transition probabilities, $\boldsymbol{\mu}$ for all mean vectors $\boldsymbol{\mu} = \{\mu_{s,m} | m \in [1, M], s \in [1, N]\}$, and $\mathbf{\Sigma}$ for the set of all covariance matrices, $\mathbf{\Sigma} = \{\Sigma_{s,m} | m \in [1, M], s \in [1, N]\}$.

There are two sets of hidden variables required for computing the joint probability of a given observation sequence $\mathbf{x} = (x_1, ..., x_T)$ and of a corresponding label sequence $\mathbf{y}$: the sequence of states (called the segmentation) $\mathbf{s} = (s_1, ..., s_T)$; and the sequence of numbers of Gaussian components (in Gaussian mixtures) responsible for the observations of $\mathbf{x}$, that we note $\mathbf{m} = (m_1, ..., m_T)$. The joint probability $p(\mathbf{x}, \mathbf{y}|\mathbf{w})$ may be computed by summation over these two sets of hidden variables, where $\mathbf{s}$ actually runs over the set $\mathbf{S}(\mathbf{y})$ of segmentation matching $\mathbf{y}$:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{w}) = \sum_{\mathbf{s} \in S(\mathbf{y})} \sum_{\mathbf{m}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\mathbf{w}) \qquad (2)$$

where:

$$p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\mathbf{w}) = \pi_{s_1} p_{s_1,m_1}(x_1) \prod_{t=2}^{T} a_{s_{t-1},s_t} p_{s_t,m_t}(x_t)$$

with the notation:

$$p_{s_t,m_t}(x_t) \overset{def}{=} P_{s_t,m_t} N_{s_t,m_t}(x_t)$$

Note that in practice one often uses the approximation:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{w}) \approx \max_{\mathbf{s}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}|\mathbf{w}) \qquad (3)$$

or even:

$$P(\mathbf{x}, \mathbf{y}|\mathbf{w}) \approx \max_{\mathbf{s},\mathbf{m}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\mathbf{w}) \qquad (4)$$

2.2. Maximum margin criterion for HMMs

Traditionally HMM parameters $\mathbf{w}$ are learnt to maximize the likelihood of the training data (MLE):

$$\mathbf{w}^{MLE} = argmax_{\mathbf{w}} \sum_i \log p(\mathbf{x}^i, \mathbf{y}^i|\mathbf{w}) \qquad (5)$$

where $p(\mathbf{x}^i, \mathbf{y}^i|\mathbf{w})$ is computed as in Eq. (2).

Few attempts have been made to exploit various discriminant criterion [6] [8] [17]. Recent studies have shown strong potential for the use of a margin based criterion [5, 12]. Noting:

$$\Delta_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y}) = \log p(\mathbf{x}^i, \mathbf{y}|\mathbf{w}) - \log p(\mathbf{x}^i, \mathbf{y}^i|\mathbf{w}), \quad (6)$$

the margin is usually defined for a training sequence i as:

$$margin_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i) = -\max_{\mathbf{y}} \Delta_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y}) \qquad (7)$$

The margin for training sequence i is positive if and only if $\mathbf{x}^i$ is correctly labelled and negative otherwise. One may use additional quantities in the margin definition as is classically done in the field of strutured output prediction, to account for the varying role of alternative segmentation $\mathbf{y}$. The idea is that the more $\mathbf{y}$ is far from $\mathbf{y}^i$ the larger $-\Delta_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y})$ should be, that is the more $\log p(\mathbf{x}^i, \mathbf{y}^i|\mathbf{w})$ should be bigger than $\log p(\mathbf{x}^i, \mathbf{y}|\mathbf{w})$. This leads to the following version of the margin definition:

$$margin_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i) = -\max_{\mathbf{y}} \left[\Delta_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y}) + \delta(\mathbf{y}^i, \mathbf{y})\right] \qquad (8)$$

where $\delta(\mathbf{y}^i, \mathbf{y})$ stands for a disimilarity measure between label sequences $\mathbf{y}^i$ and $\mathbf{y}$. A very interesting choice for $\delta(\mathbf{y}^i, \mathbf{y})$ could be the edit distance between the two sequences $\mathbf{y}^i$ and $\mathbf{y}$ but it cannot be handled efficiently in a Viterbi algorithm. A simpler choice is to use the hamming distance between state sequence $\mathbf{s}$ and $\mathbf{s}^i$ (if $\mathbf{s}^i$ is available which usually not the case). A simpler choice is a Kroeneker $\delta(\mathbf{y}^i, \mathbf{y}) = 1 \Leftrightarrow \mathbf{y}^i = \mathbf{y}$.

Maximum margin learning may be performed by maximizing the minimum margin over all training sequences:

$$\mathbf{w}^* = argmax_{\mathbf{w}} \min_i \{margin_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i)\} \qquad (9)$$

Equivalently, one may look for:

$$\begin{aligned} \mathbf{w}^* &= argmin_{\mathbf{w}} \max_i \{margin_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i)\} \\ &= argmin_{\mathbf{w}} \max_i \{\max_{\mathbf{y}} \Delta_w(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y}) + \delta(\mathbf{y}^i, \mathbf{y})\} \end{aligned} \qquad (10)$$

2.3. Related works

A large margin criterion similar to the one in Eq.(10) (but without δ terms) has been used in [5, 12]. However the resulting optimization problem is unbounded, which requires relying on approximations and adding carefully designed constraints. This explains why the method in [5] optimizes $\boldsymbol{\mu}$ parameters only while a recent extension [12] may optimize $\boldsymbol{\Sigma}$ as well, but it has been tested on limited artificially generated data only.

Actually [5, 12] proposed to maximize the large margin based criterion over a set $\mathcal{D}$ of training samples which are close to the current decision boundary and are correctly predicted by the current solution. Hence the optimization problem solved every iteration changes to focus on the subset $\mathcal{D}$ of training sequences such that the margin is positive and below a given smal value ϵ.

$$\mathbf{w}^* = argmin_{\mathbf{w}} \max_{i \in \mathcal{D}} \left\{ \max_{\mathbf{y}} \Delta_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y}) \right\} \qquad (11)$$

At the end the impact of successive approximations and simplifications are difficult to measure and the robustness of the method is questionable.

Recently, [11] proposed to cast the large margin HMM training in a structured output prediction problem [14]. This leads to the following optimization problem which is called a structured soft margin problem in its unconstrained form:

$$\begin{aligned} \mathbf{w}^* &= argmin_{\mathbf{w}} \left[\tfrac{\lambda}{2} Q(\mathbf{w}) \right. \\ &\left. + \textstyle\sum_i \max_{\mathbf{y}} \Delta_{\mathbf{w}}(\mathbf{x}^i, \mathbf{y}^i, \mathbf{y}) + \delta(\mathbf{y}^i, \mathbf{y}) \right] \end{aligned} \qquad (12)$$

where $Q(\mathbf{w})$ is a regularization factor (e.g. $Q(\mathbf{w}) = \|\mathbf{w}\|^2$). The main idea is to find the parameter set that gives a better score to the correct labelling than to other labellings, with a margin $\delta(\mathbf{y}^i, \mathbf{y})$ which guarantees good generalization behaviour. The two main differences with the formalization in Eq. (11) lie first in the regularization term $Q(\mathbf{w})$, and second in a summation over training samples instead of a maximum. The main difficulty of the formulation in Eq. (12) comes from the fact that the hidden variables $\mathbf{s}^i$ and $\mathbf{m}^i$ for computing $p(\mathbf{x}^i, \mathbf{y}^i|\mathbf{w})$ and $\mathbf{s}$ and $\mathbf{m}$ for $p(\mathbf{x}^i, \mathbf{y}|\mathbf{w})$ make the objective function in Eq. (12) non convex, since it is built as a sum of difference of concave terms. Since no efficient algorithm may be used for such a non convex optimization [11] proposed ideas to put the objective function in a good shape, i.e. convex. Their solution consists mainly in assuming the availability of an oracle (actually a non discriminant HMM system trained via MLE) that provides the ”‘correct value”’ of hidden variables $\mathbf{s}^i$ and $\mathbf{m}^i$ of any training sample $\mathbf{x}^i$. Using such an assumption indeed leads to a convex problem which may be solved with a gradient descent method. However, a major limitation of this approach has been stressed in [10] where the more complex HMM topology is (i.e. the stronger the oracle approximation is) the less interesting discriminant training is (wrt. non discriminant training).

3. TRAINING ALGORITHM

We choose in our work not to rely on strong assumptions and approximations and we consider the original non convex formulation in Eq. (12). This problem may be solved using a non convex optimization method that we developed but that we cannot detail here. We only provide a sketch of our method, more details may be found in [3]. The optimization problem in Eq. (12) is in the shape:

$$\mathbf{w}^* = argmin_{\mathbf{w}} f(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + R(\mathbf{w}) \qquad (13)$$

where $R(\mathbf{w})$ is an upper bound of the empirical risk that we want to minimize. The approach described in [13] belongs to the familiy of bundle methods and solves the general optimization problem in Eq. (13) whatever $R(\mathbf{w})$ provided that it is convex. We briefly describe the method in the case where $R(\mathbf{w})$ is convex, it relies on the cutting plane method. A cutting plane of $R(\mathbf{w})$ at $\mathbf{w}'$ is defined as:

$$\begin{array}{ll} c_{\mathbf{w}'}(\mathbf{w}) & = \langle a_{\mathbf{w}'}, \mathbf{w} \rangle + b_{\mathbf{w}'} \\ s.t. & c_{\mathbf{w}'}(\mathbf{w}') = R(\mathbf{w}') \\ and & \partial_{\mathbf{w}} c_{\mathbf{w}'}(\mathbf{w}') \in \partial_{\mathbf{w}} R(\mathbf{w}') \end{array} \qquad (14)$$

Here $a_{\mathbf{w}'} \in \partial_{\mathbf{w}} R(\mathbf{w}')$ is a subgradient of $R(\mathbf{w})$ at $\mathbf{w}'$ and $b_{\mathbf{w}'} = R(\mathbf{w}') - \langle a_{\mathbf{w}'}, \mathbf{w}' \rangle$. Such a cutting plane $c_{\mathbf{w}'}(\mathbf{w})$ is a linear lower bound of the risk $R(\mathbf{w})$ and $\frac{\lambda}{2}\|\mathbf{w}\|^2 + c_{\mathbf{w}'}(\mathbf{w})$ is a quadratic lower bound of $f(\mathbf{w})$.

Bundle methods aim at iteratively building an increasingly accurate piecewise quadratic lower bound of the objective function. Starting with an initial (e.g. random) solution $\mathbf{w}_1$, one first determines the solution $\mathbf{w}_2$ minimizing the approximation problem $g_1(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + c_{\mathbf{w}_1}(\mathbf{w})$. Then a new cutting plane $c_{\mathbf{w}_2}$ is built and one looks for the minimum $\mathbf{w}_3$ of the more accurate approximation problem $g_2(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \max(c_{\mathbf{w}_1}(\mathbf{w}), c_{\mathbf{w}_2}(\mathbf{w}))$. More generally at iteration t, one adds a new cutting plane at the current solution $\mathbf{w}_t$, and looks for the solution $\mathbf{w}_{t+1}$ minimizing the new approximated problem:

$$\begin{array}{ll} \mathbf{w}_{t+1} & = argmin_{\mathbf{w}} g_t(\mathbf{w}) \\ with & g_t(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \max_{j=1..t}\left\{c_{\mathbf{w}_j}(\mathbf{w})\right\} \end{array} \qquad (15)$$

At iteration t the minimization of the approximated problem in Eq. (15) can be solved by quadratic programming.

Unfortunately, the approach in [13] cannot be used for non-convex problems as the one in Eq. (12), since in this case a cutting plane is not necessarily a lower bound, which leads to a poor approximation. Indeed the linearization error $(R(\mathbf{w}) - c_{\mathbf{w}'}(\mathbf{w}))$ of a cutting plane $c_{\mathbf{w}'}$ at a point $\mathbf{w}$ may be negative, meaning that the function is overestimated at that point, we will say there is a conflict. Standard solution to overcome conflicts is to lower any new cutting plane $c_{\mathbf{w}_t}(\mathbf{w}) = \langle a_{\mathbf{w}_t}, \mathbf{w} \rangle + b_{\mathbf{w}_t}$ that gives negative linearization errors for the solutions $\mathbf{w}_j$ at previous iterations [1]. This may be done by tuning offset $b_{\mathbf{w}_t}$. In this case the convergence rate is not guaranteed any more and too slow to converge to a good solution in practice.

We propose a new algorithm with a good convergence rate that may be shown to converge to a local minimum if some hypothesis are verified. Roughly speaking, rather than solving all conflicts between a new cutting plane and all previous solutions, we focus on the conflict of this new cutting plane w.r.t the best observed solution up to now. Thus, every iteration, we check if there is any conflict between the new cutting plane and the best observed solution. If there is a conflict we first lower the cutting plane by slightly tuning b_t. At this point it may happen that the linearization error of the modified cutting plane w.r.t the best solution is still negative. Then we also modify a_t in a new way to solve the conflict and garantee convergence (see [3]).

4. EXPERIMENTS

We investigate the potential of our approach both on online handwriting recognition and on off-line handwriting recognition. Note that we implemented standard models without much tuning (e.g. same number of states per character, etc) since our major concern is more to show the significative gain one can expect from using our maximium margin training algorithm than to get the best results ever achieved on the datasets considered.

4.1. On-line handwritten digit recognition

We used a part of the Unipen international database consisting in a total of 15k samples, we used 5k samples for training and 10k samples for testing. We used a five states left-right CDHMM for each digit. The preprocessing includes standard steps such as smoothing and spatial resampling before basic features related to slant aud curvarture are computed. At the end a signal is represented as usual as a sequence of feature vectors.

Table 1 compares the performances of difference training approaches, Maximum Likelihood (MLE), Large margin with an oracle providing segmentation as in [11] (LMOracle) and our method called here Maximum Margin HMM (MMHMM). The table report classification error rates for different number of components in Gaussian mixtures (M in Eq. (2.1)). One can see that our method reaches the best results whatever the size of Gaussian mixtures. It is shown to significantly outperform the LMOracle based method (reducing the error-rate by 10% to 23%) showing that our algorithm is able to efficiently learn from partially labelled training samples. Note also that extensive results in [11] have shown significant improvement of LMOracle over various other discriminative criterion such as MMI and MCE on difficult continuous speech recognition tasks.

Table 1. Comparison of on-line handwritten digit recognition error rates for five states left-right HMMs ($N = 5$) on the Unipen database. Results are shown for a varying number of components in gaussian mixtures (Cf. Eq. (1)).

M	MLE	LMOracle	MM
1	13.50	2.01	1.53
2	10.50	1.70	1.33
3	9.48	1.82	1.53
4	15.02	1.74	1.56

Table 2. Character recognition rates on off-line handwritten words from the IAM database. Results are given for various numbers of states per charater model N and for a varying number of components in gaussian mixtures M (see text for details).

		MLE.			MM		
N	M	Big.	DicA	DicT	Big.	DicA	DicT
8	1	51.2	58.4	64.8	70.7	85.1	87.6
8	2	55.5	64.7	69.6	72.9	87.0	89.1
8	4	57.2	67.6	72.5	72.2	86.9	89.0
8	6	58.4	70.0	74.5	72.2	87.2	89.4
10	1	55.0	67.0	71.6	70.6	83.9	86.7
14	1	59.6	75.1	77.8	70.3	79.3	83.5

4.2. off-line handwritten word recognition

We also performed off-line handwriting recognition experiments on the IAM database [15]. Preprocessing stage is done as described in [7] who actually gently provided us with their preprocessed version of the database. Preprocessing consists in computing nine feaures on a liding window so that at the end a word image is represented as a sequence of feature vectors. As usual we learn character models that are concatanated to build word models.

Here again, the non discriminant system (trained with MLE) is used as initialization of discriminant system training. We report here results of our MMHMM with kroeneker-like penalty terms $\delta(\mathbf{y}_i, \mathbf{y}) = 1 \Leftrightarrow \mathbf{y}^i = \mathbf{y}$.

We performed experiments on handwritten word recognition and report both character error rates in Table 2 and

Table 3. Word recognition rates on off-line handwritten words (IAM database) for various settings (S, M) (see text for details).

		MLE			MM		
N	M	Big.	DicA	DicT	Big.	DicA	DicT
8	1	26.2	47.4	54.8	43.3	75.4	80.0
8	2	30.1	53.7	60.1	46.3	78.3	82.3
8	4	30.8	56.0	62.6	45.3	78.0	81.9
8	6	32.0	58.3	64.9	45.2	78.6	82.7
10	1	27.7	55.0	61.4	43.9	74.2	79.0
14	1	31.5	63.6	68.0	44.4	69.3	75.5

word error rates in Table 3. In both cases we compare MLE systems and MMHMM systems with a few recognition strategies. First we only use character models and bigrams computed on all the corpus (Big. colonn). Second we use a dictionary consisting of all words in the training and in the test set (DicA). It consists in 7220 words. Third we use the dictionary of all words in the test set only (DicT), which includes about 2490 words.

Table 2 compares character error rates when using standard non discriminant training (MLE) and when using our Max Margin learning algorithm (MM). One can show that our maximmum margin algorihtm allows impressive improvements at the character level whatever the number of states N, whatever the number of gaussian components in pdf M, and while using bigrams or any of the two dictionaries. For instance using bigrams, our algorithm drops the MLE recognition rate from 51.2% to 59.6%, depending on the number of states in HMMs, up to 70.3% to 72.9%. Similar improvements are observed when using dictionaries.

Table 3 performs a analoguous comparison, at the word level, between standard non discriminant training (MLE) and our Max Margin learning algorithm (MM). The same conclusions may be drawn from this table of experiments where one may observe very strong improvements even for complex models (e.g. with up to 14 states).

5. CONCLUSION

We proposed a maximum margin learning algorithm of Gaussian HMMs that allows learning with partially labelled training sets. We showed how such a learning problem may be cast into a non convex optimization problem for which we proposed a new algorithm based on bundle methods and cutting planes. We investigated the potential of our method on both on-line handwrittent digit recognition and on off-line handwritten word recognition. Our results show our method outperforms previous maximum margin algorithms and also show very strong improvements over non discriminant algorithms.

Aknowledgments

The autors would like to warmly thank H. Bunke and V. Finken from Bern university for providing the preprocessed version of the IAM database.

6. REFERENCES

[1] K. K. C. *Methods of Descent for Nondifferentiable Optimization*. Springer-Verlag, 1985.

[2] T.-M.-T. Do and T. Artieres. Conditional random fields for online handwriting recognition. In *IWFHR*, 2006.

[3] T.-M.-T. Do and T. Artieres. Large margin training for hidden markov modles with partially observed states. In *International Conference on Machine Learning*, 2009.

[4] H. Jiang and X. Li. Incorporating training errors for large margin hmms under semi-definite programming framework. *ICASSP*, 4:IV–629–IV–632, April 2007.

[5] H. Jiang, X. Li, and C. Liu. Large margin hidden markov models for speech recognition. *IEEE Transactions on Audio,Speech&Language Processing*, 2006.

[6] B. Juang and S. Katagiri. Discriminative learning for minimum error classification. In *IEEE Trans. Signal Processing, Vol.40, No.12*, 1992.

[7] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition systems. *Int. JOurnal of Pattern Recognition and Art. Intelligence*.

[8] W. P.C. and P. D. Large scale discriminative training of hidden markov models for speech recognition. *Computer Speech and Language*, January 2002.

[9] L. R. Rabiner. *A tutorial on hidden Markov models and selected applications in speech recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990.

[10] F. Sha. *Large margin training of acoustic models for speech recognition*. Phd Thesis, 2006.

[11] F. Sha and L. K. Saul. Large margin hidden markov models for automatic speech recognition. In *NIPS*. MIT Press, 2007.

[12] L. D. T. Chang, Z-Q Luo and C.-Y. Chi. A convex opt. method for joint mean and variance parameter estimation of large-margin cdhmm. In *ICASSP*, 2008.

[13] C. H. Teo, Q. V. Le, A. Smola, and S. V. Vishwanathan. A scalable modular convex solver for regularized risk minimization. In *KDD '07*, 2007.

[14] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML '04*, 2004.

[15] H. B. U. Marti. A full english sentence database for off-line handwriting recognition. In *ICDAR*, 1999.

[16] D. Yu and L. Deng. Large-margin discriminative training of hidden markov models for speech recognition. In *ICSC*, 2007.

[17] Y. Zhang, P. Liu, and F. Soong. Minimum error discriminative training for radical-based online chinese handwriting recognition. In *ICDAR*, 2007.