Large Margin Training for Hidden Markov Models with Partially Observed States

Trinh-Minh-Tri Do TRINH-MINH-TRI.DO@LIP6.FR
Thierry Artières THIERRY.ARTIERES@LIP6.FR
LIP6, Université Pierre et Marie Curie, 104 avenue du président Kennedy, 75016, Paris, France

Abstract

Large margin learning of Continuous Density HMMs with a partially labeled dataset has been extensively studied in the speech and handwriting recognition fields. Yet due to the non-convexity of the optimization problem, previous works usually rely on severe approximations so that it is still an open problem. We propose a new learning algorithm that relies on non-convex optimization and bundle methods and allows tackling the original optimization problem as is. It is proved to converge to a solution with accuracy ϵ with a rate $O(1/\epsilon)$. We provide experimental results gained on speech and handwriting recognition that demonstrate the potential of the method.

1. Introduction

Hidden Markov Models (HMMs) have been widely used for automatic speech recognition and handwriting recognition . Continuous Density HMMs (CDHMMs) are particularly suited for dealing with sequences of real-valued feature vectors that one gets after typical front-end processing in signal processing tasks. CDHMMs usually exploit Gaussian mixture models to describe the variability of observations in a state. HMM parameters are learnt with the Expectation-Maximization algorithm (EM) to maximize the joint likelihood of observation and of hidden state sequences.

Training is performed based on a partially labeled training set, that is a set of observation sequences together with the corresponding classes (classification case) or with the corresponding unit sequences (segmentation case). This is the usual setting in speech or handwriting recognition tasks, 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 phone means this phone has been recognized), the sequence of states translates into a sequence of labels (e.g. phones).

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) (Juang & Katagiri, 1992) or the Maximum Mutual Information (MMI) criterion (Woodland & Povey, 2002).

Recently, a promising direction has been explored with the development of margin-based methods for sequences (Taskar et al., 2004; Tsochantaridis et al., 2004). However these works mainly deal with fully supervised learning. There is still work to do to extend these works to the learning of CDHMMs with partially labeled data. Building on these seminal works a few approaches have been proposed for large margin learning of HMMs, especially in the speech recognition community (Sha & Saul, 2007; Jiang & Li, 2007) (see (Yu & Deng, 2007) for a review). However none of these works actually handle the whole problem of max-margin learning for HMM parameters in the standard partially labeled setting. (Jiang & Li, 2007) focuses on correctly predicted examples near the decision boundary only, while (Sha & Saul, 2007) mainly focuses on the fully supervised case, where the sequence

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

of states corresponding to the training sequences are determined by a traditionally trained HMM system via Maximum Likelihood Estimation (MLE).

Here we propose a new method for discriminative learning of CDHMM models in the complex unlabeled setting. The main difficulty one encounters when formulating the maximum margin learning of CDHMM with partially labeled data lies in the non-convexity of the optimization problem. While (Sha & Saul, 2007) consider a simpler and convex problem we build on non-convex optimization ideas and investigate here the direct optimization of the non-convex objective function using bundle methods (Kiwiel, 1985; Hiriart-Urruty & Lemarechal, 1993). Our contributions are:

- A new fast optimization method that can handle non-convex, non-differentiable function, with a theoretical analysis of its convergence rate.
- Experimental results showing the method is well adapted for large margin training of CDHMMs.

We first present the maximum margin training formalization in the general case of structured outputs and for HMM learning in particular. Then we show how such a learning problem may resume to a non-convex optimization problem that may be solved with a variant of bundle methods that we propose, for which we provide a detailed analysis convergence. Finally we provide experimental results on speech and on handwriting recognition that show first the performance of the classifiers and second the efficiency of the optimization method.

2. Large Margin Learning and HMMs

In this section, we address the problem of learning a model for structured outputs based on partially labeled data. Although our method is quite general we mainly focus here on the special case of learning a model for sequence segmentation based on partially labeled training sequences, which fits our case study of maximum margin learning for HMMs.

2.1. Large Margin Learning for Structured Outputs with Partially Labeled data

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 $\mathbf{x}^i = (x_1^i, x_2^i, ..., x_T^i) \in (\mathbb{R}^d)^T$ is an observation sequence and $\mathbf{y}^i = (y_1^i, y_2^i, ..., y_L^i) \in \mathcal{L}^L$ is the corresponding label sequence (with $L \leq T$). We consider hidden variables $(\mathbf{z}^1, \mathbf{z}^2, ..., \mathbf{z}^K)$ where $\mathbf{z}^i$ stands for the missing variables corresponding to the i^{th} training sample. For instance if we wish to learn a HMM for speech recognition, $\mathbf{z}^i$ might be the full state sequence corresponding to the i^{th} speech signal $\mathbf{x}^i$ while $\mathbf{y}^i$ is the corresponding sequence of phones.

We are interested in learning a discriminant function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ over input-output pairs which can be used to predict the output $\mathbf{y}$ for an input $\mathbf{x}$:

$$h(\mathbf{x}, \mathbf{w}) = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}} F(\mathbf{x}, \mathbf{y}, \mathbf{w}) \tag{1}$$

where $\mathbf{w}$ denotes the parameter vector to be learned. In the case of partially labeled data, the discriminant function $F(\mathbf{x}, \mathbf{y}, \mathbf{w})$ may take various forms such as $F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \max_{\mathbf{z}} g(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})$ where $g(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})$ stands for an elementary discriminant function. For instance g might be a linear function $g(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}) = \langle \Phi(\mathbf{x}, \mathbf{y}, \mathbf{z}), \mathbf{w} \rangle$ with $\Phi(\mathbf{x}, \mathbf{y}, \mathbf{z})$ being a feature vector.

Following previous works (Tsochantaridis et al., 2004; Taskar et al., 2004), learning an optimal h may be done by solving the following soft margin problem:

$$\begin{array}{lll} \min\limits_{\mathbf{w},\xi} & \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \xi^i & \\ s.t. & F(\mathbf{x}^i,\mathbf{y}^i,\mathbf{w}) \geq F(\mathbf{x}^i,\mathbf{y},\mathbf{w}) + \Delta(\mathbf{y}^i,\mathbf{y}) - \xi^i & \forall i \forall \mathbf{y} \neq \mathbf{y}^i \\ & \xi^i \geq 0 & \forall i \end{array} \tag{2}$$

where $\Delta(\mathbf{y}^i, \mathbf{y})$ terms allow taking into account differences between labellings (Cf. (Taskar et al., 2004)). The equivalent unconstrained problem is:

$$\min_{\mathbf{w}} \quad \underbrace{\frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \max_{\mathbf{y}} \big(F(\mathbf{x}^i,\mathbf{y},\mathbf{w}) + \Delta(\mathbf{y}^i,\mathbf{y}) - F(\mathbf{x}^i,\mathbf{y}^i,\mathbf{w}) \big)}_{f(\mathbf{w})} \tag{3}$$

where $f(\mathbf{w})$ is the primal objective function. A variant consists in using a softmax instead of a max in (3):

$$\begin{array}{l} \max_{\mathbf{y}} \big(F(\mathbf{x}^i,\mathbf{y},\mathbf{w}) + \Delta(\mathbf{y}^i,\mathbf{y}) - F(\mathbf{x}^i,\mathbf{y}^i,\mathbf{w}) \big) \\ \approx \max \Big[0, \Big(log \sum_{\mathbf{y} \neq \mathbf{y}^i} e^{F(\mathbf{x}^i,\mathbf{y},\mathbf{w}) + \Delta(\mathbf{y}^i,\mathbf{y})} \Big) - F(\mathbf{x}^i,\mathbf{y}^i,\mathbf{w}) \Big] \end{array} \tag{4}$$

2.2. Application to CDHMMs

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

$$p(x|s) = \sum_{\mathbf{m}=1}^{M} p_{s,m} \mathcal{N}_{s,m}(x) \tag{5}$$

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 $\Sigma_{s,m}$.

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

A N states HMM is defined with a set of parameters $\mathbf{w} = \{\mathbf{\Pi}, \mathbf{A}, \boldsymbol{\mu}, \mathbf{\Sigma}\}$. 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 all covariance matrices, $\mathbf{\Sigma} = \{\Sigma_{s,m} | m \in [1, M], s \in [1, N]\}$.

The joint probability $p(\mathbf{x}, \mathbf{y}|\mathbf{w})$ of an input-output pair $\mathbf{x} = (x_1, ..., x_T)$ and $\mathbf{y} = (y_1, ..., y_L)$ may be computed by summation over two sets of hidden variables: the sequence of states (called segmentation) $\mathbf{s} = (s_1, ..., s_T)$; and the sequence of numbers of Gaussian components (in Gaussian mixtures) responsible for the observations of $\mathbf{x}$, $\mathbf{m} = (m_1, ..., m_T)$. In fact $\mathbf{s}$ runs over the set $\mathbf{S}(\mathbf{y})$ of segmentations matching $\mathbf{y}$:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{w}) = \sum_{\mathbf{s} \in \mathbf{S}(\mathbf{y})} \sum_{\mathbf{m}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\mathbf{w}) \quad (6)$$

where, using notation $p_{s,m}(x) \stackrel{def}{=} p_{s,m}\mathcal{N}_{s,m}(x)$:

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

In practice one often uses the approximation $p(\mathbf{x}, \mathbf{y}|\mathbf{w}) \approx \max_{\mathbf{s}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}|\mathbf{w})$ or even:

$$p(\mathbf{x}, \mathbf{y}|\mathbf{w}) \approx \max_{\mathbf{s},\mathbf{m}} p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\mathbf{w}) \quad (7)$$

HMM max margin training can be easily cast in the formalism of previous section by defining the following score function, and considering $\mathbf{z} = (\mathbf{s}, \mathbf{m})$:

$$\begin{array}{l} g(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}, \mathbf{w}) = \log p(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}|\mathbf{w}) \\ F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \max_{\mathbf{s} \in S(\mathbf{y}), \mathbf{m}} g(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}, \mathbf{w}) \end{array} \quad (8)$$

or the following softmax variant:

$$F^{soft}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = log \left(\sum_{\mathbf{s} \in S(\mathbf{y}), \mathbf{m}} e^{g(\mathbf{x}, \mathbf{y}, \mathbf{s}, \mathbf{m}, \mathbf{w})} \right) \quad (9)$$

2.3. Solving the large margin HMM training optimization problem

In the context of our study solving the unconstrained problem Eq. (3) raises a major problem since $f(\mathbf{w})$ is naturally non-convex. For instance if $F(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \max_{\mathbf{z}} g(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})$ and assuming g is linear, then $F(\mathbf{x}, \mathbf{y}, \mathbf{w})$ is a convex function. Yet $f(\mathbf{w})$ is not convex because of the $-F(\mathbf{x}^i, \mathbf{y}^i, \mathbf{w})$ terms which are concave. Alternatively the constrained form in Eq. (2) raises problems too since there does not exist any efficient algorithm to solve such a non-convex problem with an exponential number of constraints.

Some previous works overcome the difficulty by considering a smaller sized problem, e.g. through a heuristic selection (based on Viterbi-like decoding and N-best lists) of a subset of training samples together with a subset of candidate labelings (see (Yu & Deng, 2007) for a review). This is a first step but further approximations are needed to solve the problem. For instance (Jiang & Li, 2007) also relies on an additional convex relaxation before solving the problem with semi definite programming. At the end the impact of successive approximations and simplifications is difficult to measure and the robustness of the method is questionable.

A more direct solution has been proposed in (Sha, 2006) which is based on primal optimization (Eq. (3)) and convex relaxation ideas. These authors proposed to overcome the non-convexity of f by linearizing function g (as in Eq. (8)) and by simplifying concave terms $-F(\mathbf{x}^i, \mathbf{y}^i, \mathbf{w})$ in the following way. They remove the maximization in the computation of $F(\mathbf{x}^i, \mathbf{y}^i, \mathbf{w})$ by using an oracle (A Generative System learnt with MLE) providing a guess $\mathbf{s}^i_{gs}, \mathbf{m}^i_{gs}$ for $\text{argmax}_{\mathbf{s},\mathbf{m}}[g(\mathbf{x}^i, \mathbf{y}^i, \mathbf{s}, \mathbf{m}, \mathbf{w})]$. Hopefully, using the oracle trick the objective function becomes convex in $\mathbf{w}$:

$$f^o(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \sum_i \max_{\mathbf{y}} \left(F(\mathbf{x}^i, \mathbf{y}, \mathbf{w}) + \Delta(\mathbf{y}^i, \mathbf{y}) - g(\mathbf{x}^i, \mathbf{y}^i, \mathbf{s}^i_{gs}, \mathbf{m}^i_{gs}, \mathbf{w}) \right) \quad (10)$$

At the end, the quality of the solution is not clear since there are no guarantees that such an oracle provides relevant information for learning the discriminant system. Such a limitation has been stressed in (Sha, 2006) where the more complex HMMs topology is (i.e. the stronger the oracle approximation is) the less interesting discriminant training is.

3. Non-Convex Optimization Algorithm

We consider the optimization problem below which includes the maximum margin HMM learning problem as a special case (Cf. Eq. (3), (8)):

$$\min_{\mathbf{w}} f(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + R(\mathbf{w}) \quad (11)$$

where $R(\mathbf{w})$ is an upper bound of the empirical risk that we want to minimize.

Our algorithm is inspired by a recent variant of bundle methods for minimizing convex regularized risk in machine learning problems (Teo et al., 2007; Joachims, 2006). This variant has two main advantages, the first one being its very good convergence rate, the second one being its relevant stopping criterion, namely the gap between the objective and the minimum of the approximated problem. Our algorithm inherits both advantages, but is also able to solve non-convex problems. The approach described in (Teo et al., 2007) solves the general optimization problem in Eq. (11) whatever $R(\mathbf{w})$ provided that it is convex. We briefly

describe now the method in the case where $R(\mathbf{w})$ is convex. It relies on the cutting plane technique, where a cutting plane of $R(\mathbf{w})$ at $\mathbf{w}'$ is defined as:

$$\begin{array}{ll} c_{\mathbf{w}'}(\mathbf{w}) & = \langle a_{\mathbf{w}'}, \mathbf{w} \rangle + b_{\mathbf{w}'} \\ s.t. & c_{\mathbf{w}'}(\mathbf{w}') = R(\mathbf{w}') \\ and & \partial_{\mathbf{w}} c_{\mathbf{w}'}(\mathbf{w}') \in \partial_{\mathbf{w}} R(\mathbf{w}') \end{array} \quad (12)$$

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 $f(\mathbf{w})$.

The bundle method aims 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:

$$\begin{array}{ll} \mathbf{w}_{t+1} & = \text{argmin}_{\mathbf{w}}\, g_t(\mathbf{w}) \\ v_t & = \min_{\mathbf{w}} g_t(\mathbf{w}) \\ with & g_t(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \max_{j=1..t}\{c_j(\mathbf{w})\} \end{array} \quad (13)$$

where, in the convex case, $c_j(\mathbf{w}) \equiv c_{\mathbf{w}_j}(\mathbf{w})$ is defined as in Eq. (12). We use the notation $c_j(\mathbf{w})$ rather than $c_{\mathbf{w}_j}(\mathbf{w})$ to stress that $c_j(\mathbf{w})$ is built from solution $\mathbf{w}_j$ in iteration j but does not necessarily coincide with $c_{\mathbf{w}_j}(\mathbf{w})$ as we will see below in the non-convex case.

At iteration t the minimization of the approximated problem in Eq. (13) can be solved by quadratic programming. Note that by construction of the quadratic lower bound $g_t(\mathbf{w})$ the minimum of the approximated problem $v_t = g_t(\mathbf{w}_{t+1})$ increases every iteration. It may be shown that the gap between the minimum observed value of the objective function and the minimum of the approximated problem, $f(\mathbf{w}) - g_t(\mathbf{w}_{t+1})$, decreases towards zero and that it requires $O(1/\lambda\epsilon)$ iterations to reach a gap less than ϵ (Teo et al., 2007).

Handling non-convex risk function

Unfortunately, the approach in (Teo et al., 2007) cannot be used for non-convex problems since the minimization of the approximated problem may lead to a local maximum $\mathbf{w}$. Figure 1b illustrates a situation where the method designed for convex cases yields such an improper solution. Consider we get the two cutting planes computed at $\mathbf{w}_1$ and $\mathbf{w}_2$ after two iterations. Then minimizing the approximated problem gives $\mathbf{w}_3$ which is a local maximum.

The problem comes from the fact that a cutting plane of a non-convex function is not necessarily a lower bound, which leads to a poor approximation (see Figure 1a). 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. In the following we will say in such a case that there is a conflict between $c_{\mathbf{w}'}$ and $\mathbf{w}$.

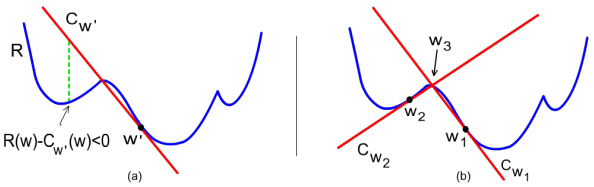


Figure 1. Cutting planes and linearization errors

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 (Kiwiel, 1985). This may be done by tuning offset $b_{\mathbf{w}_t}$. However, this approach does not guarantee any more the improvement of v_t, as the change in the cutting plane parameters changes the approximated problem. This is the reason why the convergence rate of standard bundle methods is not guaranteed, and usually slow in practice.

Instead, our algorithm handles non-convex risk while still preserving the good convergence rate $O(1/\lambda\epsilon)$, it is described in Algorithm 1. One key idea lies in that rather than solving all conflicts between a new cutting plane and all previous solutions, we focus on the conflict of the new cutting plane $c_{\mathbf{w}_t}$ w.r.t the best observed solution up to now, $\mathbf{w}_t^*$ (see line 4). Fundamentally we allow eventual overestimation at other previous solutions and focus on the area of the best observed solution by considering there is a conflict if and only if condition (14) is not satisfied:

$$c_{\mathbf{w}_t}(\mathbf{w}_t^*) = \langle a_{\mathbf{w}_t}, \mathbf{w}_t^* \rangle + b_{\mathbf{w}_t} \leq R(\mathbf{w}_t^*) \quad (14)$$

A second main idea lies in the modification procedure of $c_{\mathbf{w}_t}$ in case of conflict where $c_{\mathbf{w}_t}$ is adjusted into c_t. Finally the next solution $\mathbf{w}_{t+1}$ is found by minimization of the approximated problem Eq. (13) like in the convex case but where c_j denotes now either the "true" cutting plane $c_{\mathbf{w}_j}$ computed at iteration j (Cf. Eq. (12)), or its modified version in case there was a conflict at iteration j.

SOLVE CONFLICT

In case of conflict we look for a modified cutting plane that satisfies both Eq. (14) and Eq. (15):

$$\frac{\lambda}{2}\|\mathbf{w}_t\|^2 + c_t(\mathbf{w}_t) \geq f(\mathbf{w}_t^*) \tag{15}$$

whose interest will appear later in the convergence analysis subsection. Note that $c_{\mathbf{w}_t}$ always satisfies (15) by definition of $\mathbf{w}_t^*$, so that c_t also satisfies (15) in case there is no conflict.

Algorithm 2 guarantees that the new cutting plane c_t with parameters a_t and b_t satisfies condition (14) and (15). First it tries to solve the conflict by tuning b_t while fixing $a_t = a_{\mathbf{w}_t}$. The two conditions (14) and (15) may be rewritten as:

$$\begin{array}{ll} b_t & \leq R(\mathbf{w}_t^*) - \langle a_{\mathbf{w}_t}, \mathbf{w}_t^* \rangle = U \\ b_t & \geq f(\mathbf{w}_t^*) - \frac{\lambda}{2}\|\mathbf{w}_t\|^2 - \langle a_{\mathbf{w}_t}, \mathbf{w}_t \rangle = L \end{array} \tag{16}$$

which defines an upper bound U and a lower bound L of b_t. If $L \leq U$ any value in (L, U) works (in our implementation we set $b_t = L$). It may happen that $L > U$, then tuning b_t is not enough. We must adjust both b_t and the normal vector a_t to make sure that the conflict is solved (see Line 5 in Algorithm 2). The chosen values of a_t and b_t define a new cutting plane that trivially satisfies condition (15). It also satisfies condition (14) as we show now.

$$\begin{array}{ll} & \langle a_t, \mathbf{w}_t^* \rangle + b_t \\ = & \langle a_t, \mathbf{w}_t^* \rangle + f(\mathbf{w}_t^*) - \frac{\lambda}{2}\|\mathbf{w}_t\|^2 - \langle a_t, \mathbf{w}_t \rangle \\ = & R(\mathbf{w}_t^*) + \langle a_t + \frac{\lambda}{2}(\mathbf{w}_t^* + \mathbf{w}_t), \mathbf{w}_t^* - \mathbf{w}_t \rangle \end{array} \tag{17}$$

where we used the definition of objective function $f(\mathbf{w}_t^*) = \frac{\lambda}{2}\|\mathbf{w}_t^*\|^2 + R(\mathbf{w}_t^*)$. Then, we substitute $-\lambda \mathbf{w}_t^*$ for a_t (Cf. Line 5) and obtain

$$\begin{array}{ll} \langle a_t, \mathbf{w}_t^* \rangle + b_t & = R(\mathbf{w}_t^*) - \frac{\lambda}{2}\|\mathbf{w}_t^* - \mathbf{w}_t\|^2 \\ & \leq R(\mathbf{w}_t^*) \end{array} \tag{18}$$

Convergence Analysis

Our algorithm improves in two ways over previous standard non-convex bundle methods. First, it generates a sequence $\mathbf{w}_t$ that converges to a solution $\mathbf{w}^*$ where $f(\mathbf{w}^*) \leq f(\mathbf{w}_t) \forall t$, which is not guarantied with standard methods that may generate (stationary) cluster points, not necessarily better than previous solutions. Second, we provide below an upper bound on the convergence rate of our algorithm which converges with $O(1/\epsilon)$ rate, one cannot derive such a rate for standard bundle methods. Experimentally, after having reached "a moderate gap"(which is fast) no conflicts arise and our algorithm behaves like (Teo et al., 2007). One can assume $f(\mathbf{w})$ being locally convex then, which would make it converge to a local minimum.

Algorithm 1 Non-Convex Cutting Plane

1: **Input:** $\mathbf{w}_1$, λ, ϵ
2: **for** $t = 1$ to ∞ **do**
3: Define $c_{\mathbf{w}_t}$ according to Eq. (12)
4: $\mathbf{w}_t^* = \text{argmin}_{\mathbf{w}_j \in \{\mathbf{w}_1, \ldots \mathbf{w}_t\}} f(\mathbf{w}_j)$
5: **if** condition (14) is not satisfied **then**
6: $c_t = SolveConflict(\mathbf{w}_t^*, \mathbf{w}_t, c_{\mathbf{w}_t})$
7: **else** $c_t = c_{\mathbf{w}_t}$
8: Compute $\mathbf{w}_{t+1}$ and v_t according to Eq. (13)
9: $gap_t = f(\mathbf{w}_t^*) - v_t$
10: **if** $gap_t \leq \epsilon$ **then return** $\mathbf{w}_t^*$
11: **end for**

Algorithm 2 SolveConflict

1: **Input:** $\mathbf{w}_t^*, \mathbf{w}_t, c_{\mathbf{w}_t}$ with parameters $(a_{\mathbf{w}_t}, b_{\mathbf{w}_t})$
2: **Output:** c_t with parameters (a_t, b_t)
3: Compute L, U according to (16)
4: **if** $L \leq U$ **then** $[a_t, b_t] = [a_{\mathbf{w}_t}, L]$ **else**
5: $[a_t, b_t] = [-\lambda \mathbf{w}_t^*, f(\mathbf{w}_t^*) - \frac{\lambda}{2}\|\mathbf{w}_t\|^2 - \langle a_t, \mathbf{w}_t \rangle]$

We first recall a result from (Teo et al., 2007) which is needed in our proof (Lemma 3.1). Then Lemma 3.2 determines a lower bound on the decrease of the gap every iteration. Finally, Theorem 3.1 proves that Algorithm 1 converges to a solution with accuracy ϵ with a rate $O(1/\lambda\epsilon)$.

Lemma 3.1. *(Teo et al., 2007) The minimum of* $\frac{1}{2}qx^2 - lx$ *with* $q, l > 0$ *and* $x \in [0, 1]$ *is bounded from above by* $-\frac{l}{2}\min(1, l/q)$.

Lemma 3.2. *Approximation gap decrease:*

$$gap_{t-1} - gap_t \geq \min\left(\frac{gap_{t-1}}{2}, \frac{(gap_{t-1})^2 \lambda}{8G^2}\right) \tag{19}$$

where the approximation gap is defined as $gap_t = f(\mathbf{w}_t^*) - v_t$, *and where* G *is an upper bound on the norm of cutting planes direction parameters* a_i.

Proof. The approximation problem (13) at iteration t can be rewritten as follows:

$$\begin{array}{lll} v_t = & \min_{\mathbf{w}, \xi} & \frac{\lambda}{2}\|\mathbf{w}\|^2 + \xi \\ & s.t. & \langle a_j, \mathbf{w} \rangle + b_j \leq \xi \;\; j = 1..t \end{array} \tag{20}$$

where ξ is a slack variable. The solution is given by a saddle point of the Lagrangian that must be minimized wrt. parameters $\mathbf{w}, \xi$ and maximized wrt. Lagrange multipliers. One gets easily the dual form:

$$\begin{array}{lll} v_t = & \max_{\alpha \in \mathbb{R}^t} & D_t(\alpha) = -\frac{\lambda}{2}\|\frac{\alpha A_t}{\lambda}\|^2 + \alpha B_t \\ & s.t & \alpha_i \geq 0 \;\; \forall i = 1..t \\ & & \sum_{i=1..t} \alpha_i = 1 \end{array} \tag{21}$$

where $A_t = [a_1; ...; a_t]$ and $B_t = [b_1; ...; b_t]$ and α stands for the vector of Lagrange multipliers (of length t at iteration t). Let α_t be the solution maximizing $D_t(\alpha)$[1]. The primal solution, which may be obtained using saddle point optimality conditions (of Lagrange duality), is: $\mathbf{w}_{t+1} = -\frac{\alpha_t A_t}{\lambda}$, with: $v_t = D_t(\alpha_t)$.

Unfortunately v_t cannot be computed explicitly since it is the result of a quadratic program. However, we may interestingly study the dual D_t for a subspace of α values, along the segment line between $\alpha^{start} = [\alpha_{t-1}, 0]$ and $\alpha^{end} = [0, .., 0, 1]$. It is easy to verify that any point along this segment

$$\alpha(\eta) = \eta\alpha^{start} + (1-\eta)\alpha^{end} \ \eta \in [0,1] \tag{22}$$

is feasible. Note also that $D_t(\alpha^{start}) = D_{t-1}(\alpha_{t-1}) = v_{t-1}$, which implies naturally that $v_t \geq v_{t-1}$.

Substituting Eq. (22) into Eq. (21) and noting that $A_t = [A_{t-1}; a_t], B_t = [B_{t-1}; b_t]$, we get a quadratic form (in η) for $D_t(\alpha(\eta))$:

$$\begin{array}{rl} D_t(\alpha(\eta)) & = -\frac{1}{2\lambda}\|a_t - \alpha_{t-1}A_{t-1}\|^2\eta^2 \\ & + \big(\frac{1}{\lambda}\|\alpha_{t-1}A_{t-1}\|^2 - \alpha_{t-1}B_{t-1} \\ & + \frac{1}{\lambda}\langle a_t, \alpha_{t-1}A_{t-1}\rangle + b_t\big)\,\eta \\ & - \frac{1}{2\lambda}\|\alpha_{t-1}A_{t-1}\|^2 + \alpha_{t-1}B_{t-1} \end{array} \tag{23}$$

This expression may be simplified by using that α_{t-1} was the solution of the dual approximation problem at iteration $t-1$, hence $\mathbf{w}_t = -\frac{\alpha_{t-1}A_{t-1}}{\lambda}$ and $v_{t-1} = D_{t-1}(\alpha_{t-1})$. Then:

$$\begin{array}{rl} D_t(\alpha(\eta)) & = -\frac{1}{2\lambda}\|a_t + \lambda\mathbf{w}_t\|^2\eta^2 \\ & + \big(\frac{\lambda}{2}\|\mathbf{w}_t\|^2 + \langle a_t, \mathbf{w}_t\rangle + b_t - v_{t-1}\big)\,\eta \\ & + v_{t-1} \\ & = -\frac{1}{2}q\eta^2 + l\eta + v_{t-1} \end{array}$$

with $q = \frac{1}{\lambda}\|a_t + \lambda\mathbf{w}_{t-1}\|^2$ and $l = \frac{\lambda}{2}\|\mathbf{w}_t\|^2 + \langle a_t, \mathbf{w}_t\rangle + b_t - v_{t-1}$. Note that $Eq.(15) \Rightarrow l \geq f(\mathbf{w}_t^*) - v_{t-1}$.

We consider now two cases. If $l \leq 0$, and using that $v_{t-1} \leq v_t$, one gets $f(\mathbf{w}_t^*) \leq v_{t-1} \leq v_t$ which yields $gap_t \leq 0 \leq gap_{t-1} - \min(\frac{gap_{t-1}}{2}, \frac{(gap_{t-1})^2\lambda}{8G^2})$ assuming naturally that $gap_{t-1} > \epsilon > 0$ since convergence was not reached at iteration $t-1$ (otherwise algorithm would have stopped). Note that we never observed such a singular case $l \leq 0$ in our experiments but its study is required to complete the proof.

The case where $l > 0$ is not as simple and we rely on Lemma 3.1 for bounding the minimum of $v_{t-1} - D_t(\alpha(\eta)) \equiv \frac{1}{2}q\eta^2 - l\eta$:

$$\min_{\eta\in[0,1]} v_{t-1} - D_t(\alpha(\eta)) \quad \leq -\frac{l}{2}\min(1, l/q)$$

[1]We used solvers from the STPR toolbox available at http://cmp.felk.cvut.cz/cmp/software/stprtool/

Next, since $\forall\eta \in [0,1], v_t \geq D_t(\alpha(\eta))$:

$$-v_t \leq -v_{t-1} - \frac{l}{2}\min(1, l/q)$$

Adding $f(\mathbf{w}_t^*)$ to both sides, using $l \geq f(\mathbf{w}_t^*) - v_{t-1}$:

$$\begin{array}{rl} f(\mathbf{w}_t^*) - v_t & \leq f(\mathbf{w}_t^*) - v_{t-1} \\ & - \frac{f(\mathbf{w}_t^*) - v_{t-1}}{2}\min(1, \frac{f(\mathbf{w}_t^*) - v_{t-1}}{q}) \end{array} \tag{24}$$

Now note that $x - \frac{x}{2}\min(1, x/q)$ is monotonically increasing $\forall q > 0$. Also $f(\mathbf{w}_t^*)$ is monotonically decreasing so that $f(\mathbf{w}_t^*) - v_{t-1} \leq f(\mathbf{w}_{t-1}^*) - v_{t-1} = gap_{t-1}$. Putting this together:

$$gap_t \quad \leq gap_{t-1} - \frac{gap_{t-1}}{2}\min(1, gap_{t-1}/q) \tag{25}$$

Finally we show that $q \leq 4G^2/\lambda$. Actually, $q = \frac{1}{\lambda}\|a_t + \lambda\mathbf{w}_{t-1}\|^2 = \frac{1}{\lambda}\|a_t + \alpha_{t-1}A_{t-1}\|^2$ where $\|\alpha_{t-1}A_{t-1}\| \leq G$ because $\forall i \|a_i\| \leq G$ and $\sum_{i=1..t-1}\alpha_{t-1}^i = 1$. Finally $\|a_t - \alpha_{t-1}A_{t-1}\|^2 \leq 4G^2$ and we get $q \leq 4G^2/\lambda$. □

Theorem 3.1. *Algorithm 1 produces an approximation gap below ϵ after T steps where :*

$$\begin{array}{l} T \leq T_0 + 8G^2/\lambda\epsilon - 2 \\ with\ T_0 = 2log_2\frac{\lambda\|\mathbf{w}_1 + a_1/\lambda\|}{G} - 2 \end{array} \tag{26}$$

and converges with a rate $O(1/\lambda\epsilon)$.

Proof. Consider the two quantities occurring in Eq. (19), $gap_{t-1}/2$ and $\lambda gap_{t-1}^2/8G^2$. We first show that the situation where $gap_{t-1}/2 > \lambda gap_{t-1}^2/8G^2$ (i.e. $gap_{t-1} > 4G^2/\lambda$) may only happen a finite number of iterations, T_0. Actually if $gap_{t-1} > 4G^2/\lambda$ Lemma 3.2 shows that $gap_t \leq gap_{t-1}/2$ and the gap is at least divided by two every iteration. Then $gap_{t-1} > 4G^2/\lambda$ may arise for at most $T_0 = log_2(\lambda gap_1/4G^2)+1$. Since $gap_1 = \frac{\lambda}{2}\|\mathbf{w}_1 + a_1/\lambda\|^2$ (it may be obtained analytically since the approximation function in the first iteration is quadratic), $T_0 = 2log_2\frac{\lambda\|\mathbf{w}_1 + a_1/\lambda\|}{G} - 2$.

Hence after at most T_0 iterations the gap decrease obeys $gap_t - gap_{t-1} \leq -gap_{t-1}^2/8G^2 \leq 0$. To estimate the number of iterations required to reach $gap_t \leq \epsilon$, we follow an idea from (Teo et al., 2007) and introduce a function $u(t)$ which is an upper bound of gap_t. Solving differential equation $u'(t) = -\frac{\lambda}{8G^2}u^2(t)$ with boundary condition $u(T_0) = 4G^2/\lambda$ gives $u(t) = -\frac{8G^2}{\lambda(t+2-T_0)} \geq gap_t\ \forall t \geq T_0$. Solving $u(t) \leq \epsilon \iff t \geq 8G^2/\lambda\epsilon + T_0 - 2$, the solution is reached with accuracy ϵ within $\big[T_0 + 8G^2/\lambda\epsilon - 2\big]$ iterations. □

4. Experiments

We provide experimental results on speech and on online handwritten digit recognition and analyze experimentally the convergence behavior of our method.

Automatic Speech Recognition

We performed experiments on the TIMIT database with the standard splitting into train, development and test data. The signal was preprocessed using the procedure described in (Sha & Saul, 2007). There are 3696 utterances and over 1 million frames in the training set. A left-right HMM with one to three states and Gaussian mixture probability densities was build for each of 48 phonetic classes. We followed standard conventions in mapping the 48 phonetic labels down to 39 broader phone categories and error rates were computed as the sum of substitution, deletion, and insertion error rates from the alignment process.

We naturally compared our algorithms with a non discriminant system (MLE) (trained with the HTK Toolkit). In addition this MLE system is used during the training of discriminant systems both for initialization and for regularization. Actually we used the regularization term $\frac{\lambda}{2}\|\mathbf{w}-\mathbf{w}^{MLE}\|^2$ which experimentally performs slightly better than $\frac{\lambda}{2}\|\mathbf{w}\|^2$. Moreover, using $\|\mathbf{w}-\mathbf{w}^{MLE}\|^2$ for regularization term leads to a bigger optimal value of λ than using $\|\mathbf{w}\|^2$, which reduces considerably the number of training steps (and also the number of cutting planes required to approximate well the objective function). We implemented two variants of our method (non-convex optimization or NCO), one uses the hard-max (NCO-H) and the other one (NCO-S) uses the soft-max version over all possible labellings (see Eq. (4),(9)), this latter version is implemented with a Forward-Backward procedure. Also, we used the hamming distance for $\Delta(\mathbf{y}^i, \mathbf{y})$.

Experimentally NCO-S is about 10 times slower than NCO-H which is 2 times slower than MLE training, we give hints now. Actually learning cost mainly decomposes into two terms, computing frames probabilities and dynamic programming. In exprimental settings such as in speech recognition the first term dominates and is similar for NCO and MLE methods if training sentences include many different phones. Besides, to reach a $gap < 1\%$, NCO-H requires about two times more iteration than MLE requires to converge. Finally NCO-H training is 2 times slower than MLE.

We compared our methods to three competitive discriminant methods, the large margin convex formulation of (Sha & Saul, 2007) (named Oracle), and two benchmark discriminant methods, Conditional Maximum Likelihood (CML) and Minimum Classification Error (MCE). Table 1 shows phone error rates of all these methods for one-state HMM per phone. Note that Oracle, MCE and MMI results are taken from (Sha, 2006) and correspond to the same experimental setting. These results clearly show first that discriminant approaches significantly outperform MLE training, and second that large margin approaches (NCO and Oracle) significantly outperform the two other discriminant methods. Note also that the two variants of our method NCO-H and NCO-S perform similarly. Since NCO-H is much faster we report only NCO-H results in the following. Table 2 shows results with a few states per left-right phone HMM, for the two most efficient techniques (NCO and Oracle) only. Note that (Sha, 2006) only report results for 3 states HMM with a small number of gaussians. As may be seen in these experiments the oracle method is not able to exploit the increasing complexity of the models while our method can take advantage of the number of states to reach lower error rates. We believe that this success comes from the original non-convex formulation.

Table 1. Phone error rates with single state phone HMMs ($N = 1$) and mixtures of M Gaussian laws per state.

M	MLE	NCO-H	NCO-S	Oracle	CML	MCE
1	44.75	31.44	31.02	31.2	36.4	35.6
2	39.54	29.70	30.21	30.8	34.6	34.5
4	36.06	29.13	29.30	29.8	32.8	32.4
8	34.46	28.29	29.11	28.2	31.5	30.9

Table 2. Phone error rates with multi-state phone HMMs.

N	M	MLE	NCO-H	Oracle(Sha, 2006)
2 states	1	38.21	29.57	Not Available
2 states	2	34.14	27.99	NA
2 states	4	32.00	27.67	NA
2 states	8	31.25	27.58	NA
3 states	1	36.70	28.70	37.8
3 states	2	31.92	27.93	32.6
3 states	4	30.28	27.40	NA
3 states	8	29.55	27.61	NA

On-line Handwriting Recognition

On-line handwriting signals are temporal sequences of the position of an electronic pen captured through a digital tablet. We used a part of the Unipen international database with a total of 15k digit samples, 5k samples for training and 10k samples for testing. We trained a five states left-right CDHMM for each digit.

Table 3 reports classification error rates of three systems, namely MLE, the Oracle method and NCO-H. Again, one can see that our method reaches the best results whatever M the number of Gaussian in Gaussian mixtures. NCO-H is shown to significantly outperform the Oracle based method showing that our algorithm has been able here too to efficiently learn from partially labeled training samples.

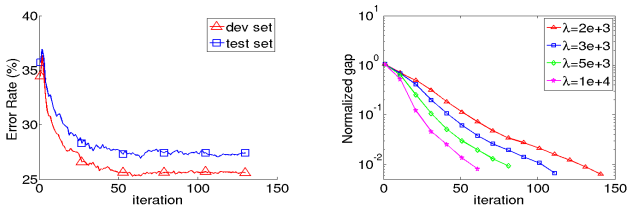
Table 3. Handwritten digit recognition error rates ($N = 5$)

M	MLE	Oracle	NCO-H
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

Convergence

Finally, we analyze experimentally the convergence rate of our algorithms (on speech recognition experiments). In these experiments learning is performed until the approximation gap becomes less than 1% of the objective function, which is enough to reach an optimal error-rate. Figure 2a plots the evolution of the error rate on the development set and on the test set against the learning iteration number. It is seen that the error rate remains stable after about 50 iterations.

Figure 2b shows the evolution of the approximation gap (in log scale) as a function of the iteration number, with different values of the regularization parameter λ. For clarity reasons we plot a normalized gap which is computed by dividing the gap by the number of frames in the training set. Note that the convergence rate depends directly on the value of λ as observed in (Teo et al., 2007), which is not the case in traditional bundle methods. This comes naturally from the use of a regularization term. More importantly, this figure shows that the convergence rate is actually closer to $O(log(1/\epsilon))$ than to our theoretical proven $O(1/\epsilon)$.



(a) (b)

Figure 2. Training LMHMM with 3 states and 4 Gaussians for speech recognition

5. Conclusion

We described a new method for maximum margin learning of CDHMMs, that allows learning with partially labeled training sets, which is still an open problem. We showed how this optimization problem may be cast as a non-convex optimization problem for which we propose a method based on bundle methods and cutting planes. We provided a convergence proof and reported experimental results on speech and handwritten digit recognition showing improved results over state of the art algorithms.

References

Hiriart-Urruty, J., & Lemarechal, C. (1993). *Convex analysis and minimization algorithms, i and ii.* Springer-Verlag.

Jiang, H., & Li, X. (2007). Incorporating training errors for large margin hmms under semi-definite programming framework. *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing*, IV–629–632.

Joachims, T. (2006). Training linear SVMs in linear time. *ACM International Conference On Knowledge Discovery and Data Mining* (pp. 217–226).

Juang, B., & Katagiri, S. (1992). Discriminative learning for minimum error classification. *IEEE Trans. Signal Processing, Vol.40, No.12* (pp. 3043–3054).

Kiwiel, K. C. (1985). *Methods of descent for nondifferentiable optimization.* Springer-Verlag.

Sha, F. (2006). Large margin training of acoustic models for speech recognition. Phd Thesis.

Sha, F., & Saul, L. K. (2007). Large margin hidden markov models for automatic speech recognition. *Neural Information Processing Systems* (pp. 1249–1256). Cambridge, MA: MIT Press.

Taskar, B., Guestrin, C., & Koller, D. (2004). Max-margin markov networks. *Neural Information Processing Systems* (pp. 25–32). Cambridge, MA.

Teo, C. H., Le, Q. V., Smola, A., & Vishwanathan, S. V. (2007). A scalable modular convex solver for regularized risk minimization. *ACM SIGKDD International Conference On Knowledge Discovery and Data Mining* (pp. 727–736). New York, USA.

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. *Proc. Intl. Conf. on Machine Learning* (pp. 104–112).

Woodland, P., & Povey, D. (2002). Large scale discriminative training of hidden markov models for speech recognition. *Computer Speech and Language*, *16*, 25–47.

Yu, D., & Deng, L. (2007). Large-margin discriminative training of hidden markov models for speech recognition. *Proceedings of the International Conference on Semantic Computing* (pp. 429–438).