Discriminative Unsupervised Learning of Structured Predictors

Linli Xu L5XU@CS.UWATERLOO.CA
Dana Wilkinson D3WILKIN@CS.UWATERLOO.CA
School of Computer Science, University of Waterloo, Waterloo ON, Canada

Finnegan Southey FINNEGAN@CS.UALBERTA.CA
Dale Schuurmans DALE@CS.UALBERTA.CA
Department of Computing Science, University of Alberta, Edmonton AB, Canada

Abstract

We present a new unsupervised algorithm for training structured predictors that is discriminative, convex, and avoids the use of EM. The idea is to formulate an unsupervised version of structured learning methods, such as maximum margin Markov networks, that can be trained via semidefinite programming. The result is a discriminative training criterion for structured predictors (like hidden Markov models) that remains unsupervised and does not create local minima. To reduce training cost, we reformulate the training procedure to mitigate the dependence on semidefinite programming, and finally propose a heuristic procedure that avoids semidefinite programming entirely. Experimental results show that the convex discriminative procedure can produce better conditional models than conventional Baum-Welch (EM) training.

1. Introduction

There have recently been a number of advances in learning structured predictors from labeled data [17, 2, 15, 16]. Structured prediction extends the standard supervised learning framework to the multivariate setting, where complex, non-scalar predictions $\hat{\mathbf{y}}$ must be produced for inputs $\mathbf{x}$. The challenge is that each component $\hat{y}_i$ of $\hat{\mathbf{y}}$ should not depend only on the input $\mathbf{x}$, but instead should take into account correlations between $\hat{y}_i$ and its neighboring components $\hat{y}_j \in \hat{\mathbf{y}}$. It has been shown in many applications that structured predictors outperform models that do not directly enforce these relationships [17, 2, 15, 16]. However, recent progress on learning structured predictors has focused primarily on the *supervised* case, where the output labels are provided with the training data. Our goal is to extend these techniques to the *unsupervised* case.

Although our technique is general, we focus our exposition on the special case of hidden Markov models (HMMs) in this paper. HMMs have been a dominant method for sequence analysis since their inception and development over 40 years ago [14], and have continued to play a central role in speech recognition research [8, 9], natural language processing [12, 9], and biological sequence analysis [6].

HMMs are a special form of graphical model for sequence data of the form $\mathbf{x} = (x_1, ..., x_L)$ and $\mathbf{y} = (y_1, ..., y_L)$, where $\mathbf{x}$ is a vector of observations and $\mathbf{y}$ is a corresponding sequence of states. The model assumes that the observation x_k at time k is conditionally independent of all other variables given the state y_k at the same time, and moreover y_k is conditionally independent of all other variables given y_{k-1}, x_k, y_{k+1} (Figure 1). The parameters that define this model are the initial state potentials $p(y_1)$, the observation potentials $p(x_k|y_k)$, and the state transition potentials $p(y_{k+1}|y_k)$. Often one assumes a *stationary* model where the transition and emission potentials do not change as a function of time. For simplicity we assume finite alphabets for the state and observation variables.

An HMM expresses a joint distribution over an observation-sequence state-sequence pair $(\mathbf{x}, \mathbf{y})$. A significant appeal of these models is that nearly every operation one might wish to perform with them is tractable. For example, an HMM can be used to efficiently decode an observation sequence to recover an optimal hidden state sequence, $\mathbf{y}^* = \arg\max_{\mathbf{y}} p(\mathbf{y}|\mathbf{x})$, by the Viterbi algorithm [14]. Another example is

Appearing in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, 2006.

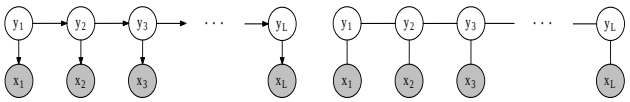

Figure 1. Equivalent directed and undirected representations of a hidden Markov model.

maximum likelihood training: if one is given *complete* training data expressed in paired sequences $(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_t, \mathbf{y}_t)$ then training a hidden Markov model to maximize joint likelihood is a trivial matter of setting the observation and transition potentials to the observed frequency counts of each state→state, state→observation, and initial state patterns.

More often, however, one is more interested in learning a *conditional model* $p(\mathbf{y}|\mathbf{x})$ rather than a joint model $p(\mathbf{x}, \mathbf{y})$, because the conditional model serves as the structured predictor of a joint labeling $\mathbf{y}$ for an input sequence $\mathbf{x}$. For conditional models it has long been observed that discriminative (conditional likelihood) training is advantageous compared to joint (maximum likelihood) training. In fact, the significant recent progress on learning structured predictors has been based on developing training procedures that exploit discriminative criteria, such as conditional likelihood or margin loss; for example, as in conditional random fields [10], discriminative sequence training [2, 17], and maximum margin Markov networks [15]. The decoding accuracy achieved by these techniques generally exceeds that of simple maximum likelihood.

One major limitation of current discriminative training algorithms, however, is that they are all *supervised*. That is, these techniques require complete state sequences $\mathbf{y}$ to be provided with the observation sequences $\mathbf{x}$, which precludes the use of unsupervised or semi-supervised approaches. This is a serious limitation because in most application areas of sequence processing—be it speech, language, or biological sequence analysis—labeled state sequence information is very hard or expensive to obtain, whereas unlabeled observation sequences are very cheap and available in almost unlimited supply. Intuitively, much of the state-class structure of the domain can already be inferred from a massive collection of unlabeled observation sequences. Nevertheless, a generally effective technique for unsupervised learning of discriminative models has yet to be developed.[1]

For the *unsupervised* training of hidden Markov models, most researchers back off to a joint model view, and use EM to recover a conditional model as a side-effect of acquiring $p(\mathbf{x}, \mathbf{y})$. Even given the advantage of discriminative training for supervised learning, most research on unsupervised training of HMMs has had to drop the discriminative approach. However, there are several well-known problems with EM. First, EM is *not* an efficient optimization technique. That is, the marginal likelihood criterion it attempts to optimize is not concave, and EM only converges to local maxima. Thus, from the global optimization perspective, EM fails to guarantee a solution to the problem. Second, if we are interested in learning a discriminative model $p(\mathbf{y}|\mathbf{x})$, there is little reason to expect that the training criterion used by EM, which focuses on improving the input model $p(\mathbf{x})$, will recover a good decoder. In fact, given the experience with discriminative versus joint supervised training, there is every reason to expect that it will not.

The contribution of this paper is simple: we show that it is possible to optimize a *discriminative* training criterion even when learning an HMM model in the *unsupervised* case. We also show that it is possible to achieve this in a convex optimization framework, where at least in principle it is possible to compute optimal solutions in polynomial time [13]. Specifically, we base our approach on the discriminative margin criterion proposed in [15], and show that even without training labels we can still learn a discriminative model $p(\mathbf{y}|\mathbf{x})$ that postulates "widely separated" hidden state sequences for different input observation sequences.

2. Discriminative unsupervised training

To develop a discriminative unsupervised training approach for structured predictors, we first consider recent progress that has been made in the univariate case. Specifically, we build upon current ideas on how discriminative training criteria can still be optimized in the unsupervised setting. Our proposal is much easier to explain once a detailed understanding of these recent ideas has been established.

These recent approaches are based on the large margin criterion of support vector machines (SVMs), where new unsupervised training algorithms have been developed [5, 19, 20]. The idea is to treat the missing classification labels as variables to be optimized in a joint minimization with the underlying SVM parameters. That is, one formulates the SVM training objective (the *margin loss*) as a joint function of the training labels and the SVM parameters. The unsupervised learning principle then becomes finding a labeling that results in an SVM with minimal margin loss. Obviously, constraints need to be added to the labeling to

[1] An exception is [1], which considers a *semi-supervised* training approach. Unfortunately, the technique is not applicable to the unsupervised case we address in this paper.

avoid trivial results, such as having all labels set to the same value. The previous approaches have simply constrained the labeling so that the classes are approximately balanced. Despite its simplicity, this approach appears to yield good results.

To derive a structured form of these algorithms, we followed the same methodology outlined in previous work [5, 19]. Essentially this involves following a sequence of steps: First, one takes the dual of the supervised problem, yielding an expression that involves pairwise comparisons between the supervised y-labels. Second, one re-expresses the problem in terms of the comparisons, rather than the y-labels themselves, which yields a convex function in the comparison variables, even when the objective was not convex in the original y-labels. Third, one relaxes the comparison variables to real values, and possibly relaxes additional constraints, to obtain a convex optimization problem. Finally, the comparison variables in the solution can be used to recover a classification of the original data.

To derive an unsupervised training algorithm for the structured case, we briefly review some essential details from the 2-class and multi-class univariate case.

2.1. 2-class case

Suppose one is given unlabeled data $\mathbf{x}_1, .., \mathbf{x}_n$, and wishes to solve for a binary labeling $\mathbf{y} \in \{-1, +1\}^n$. The recent proposals [5, 19] suggest finding a labeling that minimizes the standard SVM margin loss, subject to constraint that the classes stay approximately balanced $-\epsilon \leq \mathbf{y}^\top \mathbf{e} \leq \epsilon$. That is, one writes the margin loss after the SVM parameters have been optimized as a function of $\mathbf{y}$. In the primal and dual forms this can be expressed

$$\begin{aligned} \omega(\mathbf{y}) &= \min_{\mathbf{w}} \frac{\beta}{2}\|\mathbf{w}\|^2 + \sum_i [1 - y_i \boldsymbol{\phi}(\mathbf{x}_i)^\top \mathbf{w}]_+ \quad (1) \\ &= \max_{0 \leq \boldsymbol{\lambda} \leq 1} \boldsymbol{\lambda}^\top \mathbf{e} - \frac{1}{2\beta} \langle K \circ \boldsymbol{\lambda}\boldsymbol{\lambda}^\top, \mathbf{y}\mathbf{y}^\top \rangle \quad (2) \end{aligned}$$

where $[u]_+ = \max(0, u)$, "$\mathbf{e}$" denotes the vector of all 1s, "$\circ$" denotes componentwise matrix multiplication, $\langle , \rangle$ denotes $\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}$, and K denotes the kernel matrix, $K_{ij} = \boldsymbol{\phi}(\mathbf{x}_i)^\top \boldsymbol{\phi}(\mathbf{x}_j)$.

The difficulty with the primal form of ω, however, is that it is not convex in $\mathbf{y}$. The dual is also not convex, but the labels appear only as the equivalence relation matrix $M = \mathbf{y}\mathbf{y}^\top$; where $M_{ij} = 1$ if $y_i = y_j$ and $M_{ij} = -1$ otherwise. The key observation of [5, 19] is that if one expresses the margin loss in terms of the equivalence relation M, it becomes a maximum of linear functions in M and is therefore convex [3]

$$\omega(M) = \max_{0 \leq \boldsymbol{\lambda} \leq 1} \boldsymbol{\lambda}^\top \mathbf{e} - \frac{1}{2\beta} \langle K \circ \boldsymbol{\lambda}\boldsymbol{\lambda}^\top, M \rangle$$

The class balance and the equivalence relation constraints can then be re-expressed in terms of M. For example, the class balance constraint can be encoded $-\epsilon\mathbf{e} \leq M\mathbf{e} \leq \epsilon\mathbf{e}$. The equivalence relation constraint can be encoded using a well-known result [7, 11] that asserts $M \in \{-1, +1\}^{n \times n}$ is a binary equivalence relation if and only if $M \succeq 0$ and $\text{diag}(M) = \mathbf{e}$. Thus, by relaxing the remaining integer constraint on M to $[-1, +1]$, one can obtain a convex training problem

$$\min_{M \succeq 0,\ \text{diag}(M) = \mathbf{e}} \omega(M) \qquad \text{subject to} \quad -\epsilon\mathbf{e} \leq M\mathbf{e} \leq \epsilon\mathbf{e}$$

This problem can be shown to be equivalent to the semidefinite program [5, 19]

$$\begin{aligned} \min_{M, \delta, \boldsymbol{\mu} \geq 0, \boldsymbol{\nu} \geq 0} \ & \delta \qquad \text{subject to} \quad (3) \\ & \begin{bmatrix} M \circ K & \mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\nu} \\ (\mathbf{e} + \boldsymbol{\mu} - \boldsymbol{\nu})^\top & \frac{2}{\beta}(\delta - \boldsymbol{\nu}^\top \mathbf{e}) \end{bmatrix} \succeq 0 \\ & \text{diag}(M) = \mathbf{e}, \ \ M \succeq 0, \ \ -\epsilon\mathbf{e} \leq M\mathbf{e} \leq \epsilon\mathbf{e} \end{aligned}$$

The result is a convex training criterion for SVMs that is completely unsupervised, yet discriminative.

2.2. Multi-class case

To tackle the structured prediction case, we will need to use a multi-class version of this training strategy [20]. Assume one is given unlabeled data $\mathbf{x}_1, ..., \mathbf{x}_n$, but now wishes to learn a labeling $\mathbf{y}$ such that $y_i \in \{1...\kappa\}$. A multi-class labeling $\mathbf{y}$ can be represented by an $n \times \kappa$ indicator matrix D, such that $D_{iy_i} = 1$ and $D_{iu} = 0$ for $u \neq y_i$. In a multi-class SVM, the feature functions $\boldsymbol{\phi}(\mathbf{x}, y)$ are also extended to include the y-labels, which provides a separate weight vector $\mathbf{w}_u$ for each class u. Once a weight vector has been learned, subsequent test examples $\mathbf{x}$ are classified according to $y^* = \arg\max_y \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}, y)$.

For unsupervised SVM training, the problem becomes finding a multi-class labeling $\mathbf{y}$ (or an indicator matrix D) to minimize the multi-class margin loss. Although the margin loss is not uniquely determined in this case, the most common choice is given by [4]

$$\begin{aligned} \omega(D) &= \min_{\mathbf{w}} \frac{\beta}{2}\|\mathbf{w}\|^2 + \sum_i \max_u \Big[1 - D_{iu} - \\ &\qquad \mathbf{w}^\top (\boldsymbol{\phi}(\mathbf{x}_i, y_i) - \boldsymbol{\phi}(\mathbf{x}_i, u))\Big]_+ \\ &= \max_{\Lambda \geq 0, \Lambda\mathbf{e} = \mathbf{e}} \ n - \langle D, \Lambda \rangle - \tfrac{1}{2\beta}\langle K, DD^\top \rangle \\ &\qquad + \tfrac{1}{\beta}\langle KD, \Lambda \rangle - \tfrac{1}{2\beta}\langle \Lambda\Lambda^\top, K \rangle \end{aligned}$$

As in the 2-class case, the primal form of the margin loss is not convex in D. The dual form is also not convex in D, but once again D appears conveniently in this case only as D itself and the equivalence relation matrix $M = DD^\top$; where $M_{ij} = 1$ if $y_i = y_j$ and $M_{ij} = 0$ otherwise. If one re-expresses the margin loss in terms of D and M, it once again becomes a maximum of linear functions of D and M and is therefore jointly convex in D and M [3]

$$\begin{aligned}\omega(D,M) &= \max_{\Lambda\geq 0, \Lambda\mathbf{e}=\mathbf{e}} \; n - \langle D,\Lambda\rangle - \tfrac{1}{2\beta}\langle K, M\rangle \\ &\quad + \tfrac{1}{\beta}\langle KD,\Lambda\rangle - \tfrac{1}{2\beta}\langle \Lambda\Lambda^\top, K\rangle\end{aligned}$$

The class balance and equivalence relation constraints are once again required. Class balance can be enforced by $\left(\frac{1}{\kappa}-\epsilon\right)n\mathbf{e} \leq M\mathbf{e} \leq \left(\frac{1}{\kappa}+\epsilon\right)n\mathbf{e}$. However, since D and M both now appear in the objective, they need to be constrained relative to each other. Unfortunately, the constraint $M = DD^\top$ is not convex. [20] proposes to relax this constraint to the one-sided version $M \succeq DD^\top$, $\mathrm{diag}(M) = \mathbf{e}$, which combined with relaxing the Boolean constraints on M and D yields a convex training problem

$$\begin{aligned}\min_{M\succeq 0,\ \mathrm{diag}(M)=\mathbf{e},\ D\geq 0} \omega(M) \qquad \text{subject to} \quad & M \succeq DD^\top, \\ & \left(\tfrac{1}{\kappa}-\epsilon\right)n\mathbf{e} \leq M\mathbf{e} \leq \left(\tfrac{1}{\kappa}+\epsilon\right)n\mathbf{e}\end{aligned}$$

This optimization can also be converted to a semidefinite program [20], resulting in a training criteria for multi-class SVMs that is convex, unsupervised and yet still discriminative.

3. Unsupervised M^{3}Ns

We can now attempt to extend this univariate approach to the structured prediction case. We focus our presentation on learning HMM predictors under the multivariate margin loss formulation of Taskar et al. [15], however the ideas easily extend to other structured prediction models and other training criteria.

To establish the representation, initially consider the supervised problem. Suppose we are given labeled training sequences $(\mathbf{x}_1,\mathbf{y}_1),...,(\mathbf{x}_n,\mathbf{y}_n)$, where each individual training sequence is of the form $(\mathbf{x}_i = (x_{i1},...,x_{iL}), \mathbf{y}_i = (y_{i1},...,y_{iL}))$. As before, we consider feature functions $\boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_i)$ of both the observation and the label sequence. If one assumes a stationary hidden Markov model, as we do, then the vector of features $\boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_i)$ can be re-expressed as a sum of feature vectors over the local pieces of the example

$$\boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_i) = \sum_{k=1}^{L} \boldsymbol{\phi}(x_{ik}y_{ik}y_{ik-1})$$

That is, each feature vector $\boldsymbol{\phi}(x_{ik}y_{ik}y_{ik-1})$ for a local sequence piece $(x_{ik}y_{ik}y_{ik-1})$ is just a sparse vector that indicates which particular configuration is true in each local table of the graphical HMM model. The fact that the feature vector depends only on a local subset of variables $(x_{ik}y_{ik}y_{ik-1})$ encodes the conditional independence assumption of the HMM. The fact that the total feature vector for a complete labeled sequence is just a sum of the feature vectors for each local sequence piece, independent of k, encodes the stationarity assumption of the HMM.

This would be the representation one would use in training a supervised M^{3}N given labeled training sequences [15]. In the supervised case, a discriminative structured predictor can be trained by solving a quadratic program with respect to weights on the feature vector $\boldsymbol{\phi}(\mathbf{x},\mathbf{y})$. The goal of this quadratic program is to minimize the multivariate margin loss

$$\begin{aligned}\omega(\mathbf{y}_1,...,\mathbf{y}_n) &= \min_{\mathbf{w}} \frac{\beta}{2}\|\mathbf{w}\|^2 + \sum_i \max_{\mathbf{u}_i} \Big[\, \Delta(\mathbf{u}_i,\mathbf{y}_i) - \\ &\qquad \mathbf{w}^\top\big(\boldsymbol{\phi}(\mathbf{x}_i,\mathbf{y}_i) - \boldsymbol{\phi}(\mathbf{x}_i,\mathbf{u}_i)\big)\Big]_+ \quad (4)\end{aligned}$$

where $\Delta(\mathbf{u}_i,\mathbf{y}_i)$ counts the disagreements between $\mathbf{u}_i$ and $\mathbf{y}_i$, thus encouraging a margin between the correct labeling $\mathbf{y}_i$ and an alternative labeling $\mathbf{u}_i$ that increases with Hamming distance [17].

We will need to work with the dual. Taskar et al. [15] observe that the dual of a structured problem (4), although of exponential size in a naive derivation, can be factored into marginal dual variables using the conditional independence structure of the y-labeling.[2] In the HMM representation we are assuming, a compact quadratic program that computes the same multivariate margin loss can be expressed as

$$\begin{aligned}\omega(\mathbf{y}_1,...,\mathbf{y}_n) &= \max_{\boldsymbol{\mu},\boldsymbol{\nu}} \sum_{i,k,u} \mu_{ik}(u) 1_{(u\neq y_{ik})} \quad (5) \\ &\quad -\frac{1}{2\beta}\sum_{ij,k\ell,uu',vv'} \nu_{ik}(uu')\nu_{j\ell}(vv')\ \Delta\boldsymbol{\phi}_{ik}(uu')^\top \Delta\boldsymbol{\phi}_{j\ell}(vv') \\ &\quad \text{subject to} \quad \mu_{ik}(u)\geq 0, \nu_{ik}(uu')\geq 0, \\ &\qquad \sum_{u'}\nu_{ik}(uu') = \mu_{ik}(u), \sum_u \mu_{ik}(u) = 1\end{aligned}$$

where $\Delta\boldsymbol{\phi}_{ik}(uu') = (\boldsymbol{\phi}_{ik}(y_{ikk-1}) - \boldsymbol{\phi}_{ik}(uu'))$. Here i,j index training cases, k,ℓ index locations in each training sequence, and u,u' index possible relabelings at these locations. There is a dual variable $\mu_{ik}(u)$ corresponding to each singleton relabeling, and a dual

[2] In fact, these marginal dual variables correspond to the canonical parameters for the conditional Markov random field defined on $\mathbf{y}$ [18].

variable $\nu_{ik}(uu')$ corresponding to each adjacent pair relabeling.

To derive an unsupervised version of this training criterion, we now consider minimizing the multivariate margin loss ω as a function of the sequence labelings $\mathbf{y}_1, ..., \mathbf{y}_n$. Our task will be made considerably simpler by reformulating the multivariate margin loss in terms of the indicator and equivalence relation matrices that we will ultimately have to use. We first note that the quadratic terms in (5) can be reformulated as

$$\sum_{ij,k\ell,uv} \mu_{ik}(u)\mu_{j\ell}(v)\ \delta_1(iku, j\ell v)K(ik, j\ell) \qquad (6)$$
$$+ \sum_{ij,k\ell,uu',vv'} \nu_{ik}(uu')\nu_{j\ell}(vv')\delta_2(ikuu', j\ell vv')$$

where

$$\delta_1(iku, j\ell v) = 1_{(y_{ik}=y_{j\ell})} - 1_{(u=y_{j\ell})} - 1_{(y_{ik}=v)} + 1_{(u=v)}$$
$$\delta_2(ikuu', j\ell vv') = 1_{(y_{ikk-1}=y_{j\ell\ell-1})} - 1_{(uu'=y_{j\ell\ell-1})} - 1_{(y_{ikk-1}=vv')} + 1_{(uu'=vv')}$$

Here $K(ik, j\ell)$ is the inner product between the sub-feature vectors that omit the transition model features and set the current state values equal.

Next, define the indicator matrices M, N, C, D

$$\begin{aligned} M_{ik,j\ell} &= 1_{(y_{ik}=y_{j\ell})} \\ N_{ikk-1,j\ell\ell-1} &= 1_{(y_{ikk-1}=y_{j\ell\ell-1})} \\ C_{ik,u} &= 1_{(y_{ik}=u)} \\ D_{ikk-1,uu'} &= 1_{(y_{ikk-1}=uu')} \end{aligned} \qquad (7)$$

Note that by these definitions, M and N are equivalence relations on singleton and pairwise positions in the y-label sequences, respectively. Also, by these definitions $M = CC^\top$ and $N = DD^\top$. We can now also place the optimization variables into corresponding matrices $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ such that $\boldsymbol{\mu}_{ik,u} = \mu_{ik}(u)$ and $\boldsymbol{\nu}_{ik,uu'} = \nu_{ik}(uu')$. Given these definitions, we can then re-express an equivalent quadratic program to (5) in a compact matrix form. Letting p_1 be the number of singleton positions in the training data, and letting E denote the matrix of all 1's, we obtain

Theorem 1 *The multivariate margin loss (5) equals*

$$\begin{aligned} \omega(M, N, C, D) \ &= \ p_1 - \langle \boldsymbol{\mu}, C\rangle + \tfrac{1}{\beta}(\langle KC\rangle + \langle \boldsymbol{\nu}, ED\rangle) \\ &- \ \tfrac{1}{2\beta}\left(\langle M, K\rangle + \langle N, E\rangle + \langle \boldsymbol{\mu}\boldsymbol{\mu}^\top, K\rangle + \langle \boldsymbol{\nu}\boldsymbol{\nu}^\top, E\rangle\right) \end{aligned}$$

subject to $\sum_{u'} \nu_{ik}(uu') = \mu_{ik}(u) \quad \forall iku,$
$\boldsymbol{\mu} \geq 0$, $\boldsymbol{\nu} \geq 0$, $\boldsymbol{\nu}\mathbf{e} = \mathbf{e}$, $M = CC^\top$, $N = DD^\top$, *and*

$$N_{ikk-1,j\ell\ell-1} = M_{ik,j\ell}M_{ik-1,j\ell-1} \quad \forall ijk\ell \qquad (8)$$

The proof is just algebraic manipulation of (5) using (6) and the matrix definitions (7).

With this matrix form of the multivariate margin loss we can now develop a convex optimization objective for discriminative unsupervised training of a structured prediction model. In particular, given unlabeled sequences $\mathbf{x}_1, ..., \mathbf{x}_n$, we would like to solve for the sequence labelings—represented implicitly as indicator matrices M, N, C and D over the singleton and pair labelings—by minimizing the multivariate margin loss of the resulting M^{3}N predictor. In formulating the optimization problem, there are a number of constraints on the matrix variables that we need to impose.

First, one needs to impose class balance constraints to avoid trivial results. We impose class balance constraints on each local singleton labeling y_{ik} and each local pairwise labeling y_{ikk-1} by

$$\begin{aligned} \left(\tfrac{1}{\kappa} - \epsilon\right) p_1\mathbf{e} \leq \quad M\mathbf{e} \quad \leq \left(\tfrac{1}{\kappa} + \epsilon\right) p_1\mathbf{e} \\ \left(\tfrac{1}{\kappa^2} - \epsilon\right) p_2\mathbf{e} \leq \quad N\mathbf{e} \quad \leq \left(\tfrac{1}{\kappa^2} + \epsilon\right) p_2\mathbf{e} \end{aligned} \qquad (9)$$

where p_1 and p_2 are the number of singleton and pair positions in the data.

Second, we need to respect the quadratic constraints between matrices, such as $M = CC^\top$ and $N = DD^\top$. Unfortunately, these are non-convex. However, using the same approach as [20] we can relax these to the convex one-sided constraints

$$M \succeq CC^\top, N \succeq DD^\top, \mathrm{diag}(M) = \mathbf{e}, \mathrm{diag}(N) = \mathbf{e} \quad (10)$$

We also need to relax the quadratic constraints (8) that relate M and N (the equivalence relations on singleton and pairwise assignments). These are also non-convex, but can be approximated by linear constraints

$$\begin{aligned} N_{ikk-1,j\ell\ell-1} &\leq M_{ik,j\ell} \\ N_{ikk-1,j\ell\ell-1} &\leq M_{ik-1,j\ell-1} \\ N_{ikk-1,j\ell\ell-1} &\geq M_{ik,j\ell} + M_{ik-1,j\ell-1} - 1 \end{aligned} \qquad (11)$$

Finally, to pose the training problem, we need to relax the $\{0, 1\}$ integer constraints to $[0, 1]$. Putting these pieces together we get a relaxed training criterion for structured predictors that is entirely convex

$$\min_{M,N,C,D \geq 0} \omega(M, N, C, D) \quad \text{subject to (9), (10), (11)}$$

To solve this training problem in the same manner as above, one would then have to re-express this convex problem as a semidefinite program. Unfortunately, we find that the semidefinite program that results is too large to be practical. Therefore, our initial attempt

Table 1. Prediction error with different methods. Results averaged over 10 repeats, for each, EM given 10 re-starts.

DATA SET	CDHMM	EM
SYTH. DATA1 (95%)	3.38 ±0.75	15.09 ±1.92
SYTH. DATA2 (90%)	8.12 ±1.57	17.49 ±1.81
SYTH. DATA3 (80%)	22.12 ±1.40	30.06 ±1.24
SYTH. DATA4 (70%)	31.50 ±1.46	39.90 ±0.86
PROTEIN DATA1	51.75 ±1.80	58.11 ±0.47
PROTEIN DATA2	50.38 ±2.04	57.23 ±0.39

at optimizing this training criterion has taken a different approach: We can obtain a more compact training technique by using a constraint generation method

$$\min_{M,N,C,D,\delta} \delta \text{ s.t. } \delta \geq \omega(M,N,C,D;\boldsymbol{\mu}_c,\boldsymbol{\nu}_c),\ \forall c \in \mathcal{C} \quad (12)$$

Here we keep a finite set of constraints $\mathcal{C}$, where each $\boldsymbol{\mu}_c, \boldsymbol{\nu}_c$ corresponds to a set of dual parameters for an M^{3}N. Then given a current, M, N, C, D, an M^{3}N training algorithm (QP) is used to maximize $\omega(M, N, C, D; \boldsymbol{\mu}, \boldsymbol{\nu})$ as a function of μ and ν; hence adding a new constraint to (12). Then (12) can be solved for a new M, N, C, D by a smaller semidefinite program. By convexity, a fixed point must yield a global solution to the convex problem (12). Unfortunately, this training algorithm is still quite expensive. Therefore, in Section 5 below we propose some principled alternatives that are much faster, but no longer guaranteed to find a global solution.

4. Experimental results

As a proof of concept, we implemented the training technique proposed above, using CPLEX for constraint generation and SDPT3 for the outer semidefinite optimizations. Our goal in this section is not to assert that we have a practical technology, yet, that one can easily apply to real problems immediately. However, we believe the fundamental idea is important and we first want to demonstrate that the principle works, regardless of computational cost. We will then revisit the question of computational efficiency and propose some faster but approximate alternatives below.

Since we were initially limited in the sizes of the problems we could consider, we investigated six small data sets: four synthetic data sets generated from a 2-state HMM (see Figure 1), and two reduced versions of a real protein secondary structure data set obtained from the UCI repository (protein-secondary-structure). In each case, we gathered a sample of labeled sequences, removed the labels, trained the unsupervised HMM models, and used these to relabel the sequences using Viterbi decoding. We measured accuracy by first optimizing the map between predicted and possible state labels for each method, and then counting the number of misclassified positions in all training sequences. We compared the performance of the proposed convex discriminative HMM training (CDHMM) to standard EM training (EMHMM). The regularization parameter β for CDHMM was set to 0.01 for the synthetic experiments and 1.0 for the protein experiments. EM was run from a random set of initial parameters 10 times. Smoothing had little noticeable effect on EM.

The synthetic data sets were generated from a simple 2-state HMM, where each state emits either 0 or 1 according to emission probability. Emission noise was set equal to the probability of transitioning to the other state. In synthetic data set 1, there is a 5% chance of staying in the same state and a 95% chance of moving to another state; similarly, in synthetic data set 2, 3, and 4 there is a 10%, 20%, and 30% chance respectively of staying in the same state. Thus, the synthetic data sets are incrementally noisier from synthetic data set 1 to 4. To generate training data, we sampled 10 sequences of length 8 from the 2-state HMM.

For the protein sequence experiments, we created two small data sets of 10 subsequences of length 10, with endpoints randomly selected in the data. In the first experiment (protein data 1) a simple HMM was used (Figure 1), where y_i is the secondary structure tag (one of 3 values) and x_i is the amino acid tag (one of 20 values). In the second experiment (protein data 2), each observation x_i was set to a window of 3 adjacent amino acid tags (one of 20^3 values).

Table 1 shows the classification accuracies achieved by CDHMM and EMHMM on these problems. Here we see that CDHMM learns far more accurate prediction models than EMHMM. In fact, these results are quite strong, supporting our contention that the discriminative training criterion, based on M^{3}Ns, might provide a fundamental improvement over EM for learning structured predictors. Of course, one source of the advantage might simply be that the convex training criterion avoids getting stuck in local minima. However, independent of local minima, we still argue that even in the unsupervised setting, optimizing a discriminative criterion that focuses on $p(\mathbf{y}|\mathbf{x})$ is superior to optimizing a criterion that focuses solely on improving the model of $p(\mathbf{x})$ (which in fact is what EM is trying to do in this case). Below we present further evidence to attempt to support the second contention.

Unfortunately, the computational cost of the convex training is quite high (hours versus seconds) and we do not yet have a efficient optimization strategy that is able to guarantee global minimization, even though

there are no local minima. To try and address this issue, we attempt to formulate more computationally attractive versions of the proposed training criterion.

5. Efficient approximation techniques

Our main idea, currently, for a computationally efficient approximate training method is based on an equivalent reformulation of the training objective. The reformulation we propose sacrifices convexity, but permits more efficient local optimization.

The easiest way to illustrate the idea is to consider the simple 2-class univariate case from Section 2.1. Equations (1) and (2) give two equivalent expressions for the margin loss as a function of the labeling $\mathbf{y}$, expressed as primal and dual quadratic programs. It is well-known that the primal and the dual solutions are related by $\mathbf{w} = \frac{1}{\beta}\sum_j \lambda_j y_j \boldsymbol{\phi}(\mathbf{x}_j)$. This relationship is, in fact, not accidental, and one can establish several alternative quadratic programs that give the same solution as the standard primal and dual forms.

Proposition 1 *The margin loss, (1) and (2), equals*

$$\begin{aligned} \omega(M) \quad &= \quad \min_{0\le\boldsymbol{\lambda}\le 1,\boldsymbol{\xi}\ge 0} \frac{1}{2\beta}\langle K\circ\boldsymbol{\lambda}\boldsymbol{\lambda}^\top, M\rangle + \boldsymbol{\xi}^\top\mathbf{e} \\ &\textit{subject to} \quad \xi_i \ge 1 - \frac{1}{\beta}\sum_j M_{ij}\lambda_j K(\mathbf{x}_i,\mathbf{x}_j) \;\; \forall i \\ &\textit{where} \quad M = \mathbf{y}\mathbf{y}^\top \end{aligned}$$

That is, the normal *maximization* over the dual variables can be replaced by an equivalent *minimization* over the dual and slack variables. This allows one to re-express the problem of maximizing margin loss as a joint *minimization* between the dual variables $\boldsymbol{\lambda}$ and the equivalence relation M as

$$\min_M \min_{0\le\boldsymbol{\lambda}\le 1,\boldsymbol{\xi}\ge 0} \omega(M;\boldsymbol{\lambda},\boldsymbol{\xi}) = \boldsymbol{\lambda}^\top(K\circ M)\boldsymbol{\lambda}/2\beta + \boldsymbol{\xi}^\top\mathbf{e}$$
$$\text{subject to convex constraints}$$

Unfortunately, $\omega(M;\boldsymbol{\lambda},\boldsymbol{\xi})$ is not jointly convex in M and $\boldsymbol{\lambda}$, meaning that global minimization cannot be easily guaranteed. Nevertheless, the objective is marginally convex in M and $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$. This suggests an alternating minimization approach—first solve a semidefinite program in M given $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$, then solve a quadratic program in $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$ given M, and so on. A key fact about alternating minimization is that it must make monotonic progress in the objective ω. Given that the loss ω is bounded below, such a procedure is guaranteed to converge to a local minimum.

Although alternating minimization yields superior intermediate solutions to the constraint generation method used above, it still involves a semidefinite program at each alternation, making it still too expensive for large practical problems. However, our key observation is that one can now in fact sidestep semidefinite programming entirely.

Table 2. Prediction error including alternating method.

DATA SET	CDHMM	ACDHMM	EM
SYTH1	3.38 ±0.75	14.46 ±1.78	15.09 ±1.92
SYTH2	8.12 ±1.57	17.34 ±1.52	17.49 ±1.81
SYTH3	22.12 ±1.40	26.56 ±1.06	30.06 ±1.24
SYTH4	31.50 ±1.46	38.58 ±0.96	39.90 ±0.86
PROT1	51.75 ±1.80	56.67 ±0.47	58.11 ±0.47
PROT2	50.38 ±2.04	53.65 ±0.57	57.23 ±0.39

Proposition 2 *Given an SVM solution specified by a fixed $\boldsymbol{\lambda}$, the labeling $\mathbf{y}$ (and its equivalence relation $M = \mathbf{y}\mathbf{y}^\top$) that minimizes margin loss is given by the labeling that is consistent with the SVM's predictions.*

That is, the best labeling for a fixed SVM, in terms of minimizing margin loss, is simply the SVM's predictions, which in fact is a fairly obvious statement. Nevertheless, one can obtain a very fast approximate training procedure as a result: initialize the labeling, train an SVM, relabel according to the SVM's discriminant, re-train the SVM, and so on. Although this sounds like a naive heuristic, it is in fact a principled coordinate descent method: each iteration, either re-training or re-labeling, is guaranteed to be non-increasing in the margin loss. Thus the alternation must make monotonic progress in the objective and cannot oscillate, except at a fixed point, which corresponds to a local minimum. We have experimented with this approach below and found that it requires a nontrivial number of iterations (typically more than 1) to reach a fixed point—so the procedure is not completely vacuous as one might fear—but on the other hand the number of iterations rarely exceeds 5-10, so the training time is not significantly worse than training a supervised M^3N. Quite obviously, however, this heuristic only finds local minima and will be dependent on good initialization. Surprisingly, however, we have found that this heuristic alternation technique can achieve good results when applied to large scale sequence data, and is still able to surpass EM in the quality of the structured predictors it learns from unlabeled data.

5.1. Experimental evaluation

In order to gauge the impact of the approximation, the alternating heuristic (ACDHMM) was run on the same small data sets as the constraint generation method.

Table 3. Prediction error for larger data sets.

DATA SET	ACDHMM	EM
20×2-SEQ	43.12 ±2.20	46.27 ±1.51
10×5-SEQ	44.33 ±2.30	48.67 ±1.51
5×10-SEQ	46.44 ±2.12	48.67 ±1.82

Note that ACDHMM needs some initial labeling so it was seeded using the Viterbi labeling from a model learned on a single run of EM. The results shown in Table 2 show that ACDHMM is not as accurate as the exact CDHMM procedure, but generally offers better results than EM, especially with the complex model of PROT2. However, ACDHMM scales better than CDHMM so we are able to present results on much larger data sets. To demonstrate this, we use the same protein secondary structure data set but now use complete sequences (from the available set of 110) instead of sampling short segments. We show results in Table 3 for 20 samples of 2 sequences (20×2-SEQ), 10 samples of 5 sequences (10×5-SEQ), and 5 samples of 10 sequences (5×10-SEQ) taken randomly from the data set. These data sets are much larger than our earlier examples, having, on average, 337, 628, and 1214 structure observations respectively. In all cases, the observations x_i were set to a window of 7 adjacent amino acids. The results show an improvement over EM in a more realistic context that is quite infeasible using CDHMM.

6. Conclusion

We have presented a new discriminative approach to the unsupervised training of hidden Markov models. Our technique combines current ideas in discriminative sequence prediction with those in discriminative unsupervised training. To the best of our knowledge this is the first technique to formulate a convex criterion for discriminative unsupervised training.

Our experimental results, although preliminary, mirror the experience in supervised learning that, from the perspective of learning a decoder $p(\mathbf{y}|\mathbf{x})$, it is better to use a discriminative training criterion than a joint criterion. We can offer an exact but expensive training method for this criterion, or fast but inexact training methods, but cannot yet attain both.

There are many directions for future research. One of the most significant issues is overcoming the computational burden of semidefinite programming. Even though this problem is polynomial time in principle [13], current solvers limit the size of the problems we can practically handle. Generalizing the approach to arbitrary graphical models is not hard, although the usual limits on graph topology are required to ensure tractability. A more interesting issue that we have not made much progress on is dealing with continuous variables and continuous time. Finally, it would be interesting to try our technique on semi-supervised data to see if improvements over current discriminative classification techniques can be achieved.

References

[1] Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi-supervised learning for structured variables. In *Proceedings NIPS 18*, 2005.

[2] Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings ICML*, 2003.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge U. Press, 2004.

[4] K. Crammer and Y. Singer. On the algorithmic interpretation of multiclass kernel-based vector machines. *JMLR*, 2, 2001.

[5] T. De Bie and N. Cristianini. Convex methods for transduction. In *Proceedings NIP 16*, 2003.

[6] B. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge U. Press, 1998.

[7] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical report, 2000.

[8] F. Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, 1998.

[9] D. Jurafsky and J. Martin. *Speech and Language Processing*. Prentice Hall, 2000.

[10] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings ICML*, 2001.

[11] M. Laurent and S. Poljak. On a positive semidefinite relaxation of the cut polytope. *Linear Algebra and its Applications*, 223/224, 1995.

[12] C. Manning and H. Schutze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[13] Y. Nesterov and A. Nimirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1994.

[14] Lawrence Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of IEEE*, 77(2):257–286, 1989.

[15] B. Taskar, C. Guestrin, and D. Koller. Max-margin Markov networks. In *Proceedings NIPS 16*, 2003.

[16] B. Taskar, D. Klein, M. Collins, D. Koller, and C. Manning. Max-margin parsing. In *Proceedings EMNLP*, 2004.

[17] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings ICML*, 2004.

[18] M. Wainwright and M. Jordan. Graphical models, exponential families, and variational inference. Technical report, 2003.

[19] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *NIPS 17*, 2004.

[20] L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *Proceedings AAAI*, 2005.