Large Margin Methods for Label Sequence Learning

Yasemin Altun and Thomas Hofmann

Department of Computer Science,Brown University, Providence, RI

altun@cs.brown.edu, th@cs.brown.edu

Abstract

Label sequence learning is the problem of inferring a state sequence from an observation sequence, where the state sequence may encode a labeling, annotation or segmentation of the sequence. In this paper we give an overview of discriminative methods developed for this problem. Special emphasis is put on large margin methods by generalizing multiclass Support Vector Machines and AdaBoost to the case of label sequences. An experimental evaluation demonstrates the advantages over classical approaches like Hidden Markov Models and the competitiveness with methods like Conditional Random Fields.

1. Introduction

The problem of labeling, annotating or segmenting observation sequences is omnipresent in areas like natural language processing, speech recognition, information retrieval, and computational biology. Prominent examples include part-of-speech tagging, named entity classification, information extraction, continuous speech recognition, and secondary protein structure prediction. Formally, problems of this type can often be cast as the problem of inferring a label or state sequence $\mathbf{y} = (y^1, y^2, \ldots, y^T)$ with $y^t \in \Sigma$ from an observation sequence $\mathbf{x} = (x^1, x^2, \ldots, x^T)$.

Up to now, the predominant formalism for modeling label sequences has been based on Hidden Markov Models (HMMs) and variations thereof. Yet, despite its success HMMs have two major shortcomings. First, HMMs are typically trained in a non-discriminant manner using maximum likelihood estimation for a joint sampling model of observation and label sequences. Second, efficient inference and learning in this setting often requires to make questionable conditional independence assumptions. The first problem poses the challenge of finding more appropriate *objective functions*, i.e. alternatives to the log-likelihood that are more closely related to application-relevant performance measures. The second problem is one of developing more powerful *architectures*, for example, by allowing direct dependencies between a label and past/future observations (overlapping features) or by efficiently handling higher-order combinations of input features. At the same time, one would like to address these shortcomings without sacrificing some of the benefits that HMMs offer, namely a dynamic programming formulation for decoding, inference and learning.

In this paper we focus on the supervised learning framework and assume that a set of labeled training sequences is available from which the desired mapping is learned. Label sequence learning can then be thought of as a natural extension of supervised classification. In particular, we present generalizations of two of the most competitive large margin methods for classification, Support Vector Machines (SVMs) and AdaBoost, to the problem of label sequence learning.

2. Learning Architectures

We will work in the following setting: A *learning architecture* specifies a family of λ-parameterized discriminant functions $F(\mathbf{x}, \mathbf{y}; \lambda)$ that assign a numerical score to every pair of observation/label sequences. One can think of $F(\mathbf{x}, \mathbf{y}; \lambda)$ as measuring the *compatibility* between the observation sequence $\mathbf{x}$ and the label sequence $\mathbf{y}$. Each discriminant function F induces a mapping f,

$$f(\mathbf{x}; \lambda) = \arg\max_{\mathbf{y}} F(\mathbf{x}, \mathbf{y}; \lambda)\,, \tag{1}$$

where ties are arbitrarily broken. In this paper, we will restrict our attention to discriminant functions that are *linear* in some feature representation of $(\mathbf{x}, \mathbf{y})$. Hence, F has the following general functional form

$$F(\mathbf{x}, \mathbf{y}; \lambda) = \sum_k \lambda_k \psi_k(\mathbf{x}, \mathbf{y}) = \langle \lambda, \Psi(\mathbf{x}, \mathbf{y}) \rangle\,. \tag{2}$$

The remaining crucial ingredients of an architecture are thus the extracted features or statistics ψ_k.

For concreteness, let us assure ourselves that the HMM architecture can be handled as a special case in this setting. HMMs extract two types of features from a sequence pair $(\mathbf{x}, \mathbf{y})$. The first type of features deal with label-label interactions between neighboring labels:

$$\psi_k(\mathbf{x}, \mathbf{y}) = \sum_t [\![y^t = \sigma_m]\!] [\![y^{t+1} = \bar{\sigma}_m]\!], \tag{3}$$

where $\sigma_m, \bar{\sigma}_m \in \Sigma$ denote states and $[\![\cdot]\!]$ denotes the indicator function of the enclosed predicate. These features simply count how often a particular combination of labels occur at neighboring sites. The second type of features ψ_k conjunctively combine input attributes ϕ_l with states σ_m,

$$\psi_k(\mathbf{x}, \mathbf{y}) = \sum_t [\![y^t = \sigma_m]\!] \phi_l(x^t)\,. \tag{4}$$

For example, if each input is described by L attributes ϕ_l and if there are $K = |\Sigma|$ possible states, then one may extract a total of $K \cdot L$ features of this type by combining every input attribute with every state.

There are at least two ways for designing more powerful learning architectures. First, one may include direct dependencies of the type in Eq. (4) between a label variable y^t and input features $\phi_l(x^s)$ with $s \neq t$, for example, $s \in \{t-\delta, \ldots, t+\delta\}$. These are also sometimes called "overlapping" features, since the same input feature $\phi_l(x^s)$ is included in multiple statistics.

Second, in kernel-based architectures it may not be possible or efficient to work with the explicit input attributes Φ, but the data may instead be represented via kernel functions $K(x, \bar{x}) = \langle \Phi(x), \Phi(\bar{x}) \rangle$. Hence there is a need for methods that can work in a dual representation, where the data only enters through values of K on pairs of observations.

3. Loss Functions and Risks

There is no single objective function for label sequences that would be preferable in all situations, rather this choice will depend on the specific application. Consequently, we will discuss a number of reasonable alternatives.

First of all, based on a generalization of the standard zero-one classification loss to label sequences one can define the following empirical risk for n training instances

$$\mathcal{R}^{\text{zo}}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}[\![f(\mathbf{x}_i;\lambda) \neq \mathbf{y}_i]\!]\,. \tag{5}$$

A second risk function we consider is based on the rank loss [1, 2] which measures the fraction of incorrect label sequences that are ranked higher than the correct one. In order to account for varying sequence lengths, we include weights $w(T_i) > 0$ for every sequence, where T_i is the length of the i-th sequence,

$$\mathcal{R}^{\text{rk}}(\lambda, w) = \sum_{i=1}^{n} w(T_i) \sum_{\mathbf{y}\neq\mathbf{y}^i} [\![F(\mathbf{x}_i,\mathbf{y};\lambda) \geq F(\mathbf{x}_i,\mathbf{y}_i;\lambda)]\!]\,. \tag{6}$$

Since we expect the rank loss to scale exponentially with the sequence length, we have investigated weighting functions $\log w(T_i) = T_i \log \pi$ with $\pi \in (0;1]$ in the experiments.

Lastly, the Hamming risk [1, 2] measures the zero-one loss for individual labels and reduces to the standard empirical misclassification risk, if the sequential nature of the data is ignored,

$$\mathcal{R}^{\text{hm}}(\lambda) = \frac{1}{\sum_i T_i}\sum_{i=1}^{n}\sum_{t=1}^{T_i}[\![f^t(\mathbf{x}_i;\lambda) \neq y_i^t]\!]\,. \tag{7}$$

The three risk functions presented are discontinuous in λ and generally difficult to optimize. Moreover, minimizing the empirical risk alone is not sufficient to ensure good generalization performance. The methods discussed in the sequel can be understood as minimizing an upper bound on one of these risk functions, possibly combined with a regularization term.

4. Conditional Random Fields

Conditional random fields (CRFs) [3] can be considered the state-of-the-art in label sequence learning. CRFs are a natural generalization of logistic regression to label sequences. The starting point is to define a conditional probability of a label sequence given an observation sequence as

$$p(\mathbf{y}|\mathbf{x};\lambda) = \frac{1}{Z(\mathbf{x},\lambda)}\exp\left[F(\mathbf{x},\mathbf{y};\lambda)\right]\,, \tag{8}$$

where $Z(\mathbf{x},\lambda)$ is a normalization constant. One can interpret the weights λ as the canonical parameters and the $\psi_k(\mathbf{x},\mathbf{y})$ as the sufficient statistics of a conditional exponential family.

A number of algorithms have been proposed to estimate λ by maximizing the conditional likelihood, or equivalently by minimizing

$$\mathcal{R}^{\text{log}}(\lambda) = -\frac{1}{n}\sum_{i=1}^{n}\log p(\mathbf{y}_i|\mathbf{x}_i;\lambda)\,. \tag{9}$$

These include iterative scaling [3] and various flavors of conjugate gradient descent and second order methods [4, 5, 6] as well as approximation methods such as the voted perceptron [7]. Usually a regularization term proportional to the squared norm $\|\lambda\|^2$ is added, resulting in a penalized likelihood criterion [8]. The negative log-likelihood provides an upper bound on the empirical zero-one risk:

Proposition 1. $\mathcal{R}^{zo}\log 2 \leq \mathcal{R}^{log}$.

Proof. Distinguish two cases:
(i) $F(\mathbf{x}_i,\mathbf{y}_i;\lambda) > \max_{\mathbf{y}\neq\mathbf{y}_i} F(\mathbf{x}_i,\mathbf{y};\lambda)$ in which case $\log 2[\![f(\mathbf{x}_i;\lambda)\neq\mathbf{y}_i]\!] = 0 = -\log 1 \leq -\log p(\mathbf{y}_i|\mathbf{x}_i;\lambda)$.
(ii) $F(\mathbf{x}_i,\mathbf{y}_i;\lambda) \leq \max_{\mathbf{y}\neq\mathbf{y}_i} F(\mathbf{x}_i,\mathbf{y};\lambda)$ in which case $\log 2[\![f(\mathbf{x}_i;\lambda)\neq\mathbf{y}_i]\!] = -\log\frac{1}{2} \leq -\log p(\mathbf{y}_i|\mathbf{x}_i;\lambda)$, since $p(\mathbf{y}_i|\mathbf{x}_i;\lambda)\leq\frac{1}{2}$.
Summing over all i completes the proof. □

Following [9, 2] one can also use a modification of the logarithmic risk based on the marginal probabilities of the individual label variables y_i^t in estimating λ,

$$\mathcal{R}^{\text{mg}}(\lambda) = -\frac{1}{\sum_i T_i}\sum_{i=1}^{n}\sum_{t=1}^{T_i}\log p(y_i^t|\mathbf{x};\lambda) \tag{10}$$

This defines an upper bound on the Hamming risk $\mathcal{R}^{\text{hm}}$, if one uses a pointwise decoding function

Proposition 2. *With* $f^t(\mathbf{x};\lambda) = \arg\max_\sigma \Pr(Y^t=\sigma|\mathbf{x};\lambda)$, *the following bound holds:* $\log 2\cdot\mathcal{R}^{hm}(\lambda) \leq \mathcal{R}^{mg}(\lambda)$.

Proof. Omitted, analogous to Proposition 1. □

Again, standard numerical optimization procedures such as conjugate gradient can be used to optimize $\mathcal{R}^{\text{mg}}$, since the computation of the gradient can be carried out efficiently by dynamic programming as shown in [9].

5. Hidden Markov SVMs

We present a generalization of SVMs to label sequence learning. As a first step, we propose to generalize the multiclass separation margin [10, 11] and define the margin achieved by λ on an instance $(\mathbf{x},\mathbf{y})$ as

$$\gamma(\mathbf{x},\mathbf{y};\lambda) \equiv [F(\mathbf{x},\mathbf{y};\lambda) - \max_{\mathbf{y}'\neq\mathbf{y}} F(\mathbf{x},\mathbf{y}';\lambda)]/2\,. \tag{11}$$

Notice that $\gamma(\mathbf{x},\mathbf{y}) > 0$ implies that the correct label sequence receives the highest score. In general, we want the score of the correct output not only to be maximal, but also to be larger than the second best output by some margin, which is what γ measures. Then we propose to choose λ by maximizing the minimal margin, i.e. $\lambda^* = \arg\max_\lambda \min_i \gamma(\mathbf{x}_i,\mathbf{y}_i;\lambda)$.

Notice that the discriminant function F is linear in the feature representation ψ_k. Hence, if a minimal margin of $\gamma > 0$ can be achieved, then the margin can be made arbitrary large by scaling λ. Using the standard trick of fixing the functional margin at 1, one can hence equivalently minimize the squared norm $\|\lambda\|^2$ subject to the margin constraints.

In order to accommodate for margin violations one can generalize this formulation in two ways. First one may add one slack variable ξ_i for every training sequence. A soft-margin SVM problem can than be formulated as

$$\textbf{SVM}_1\textbf{:}\ \min_{\lambda,\xi}\ \frac{1}{2}\|\lambda\|^2 + C\sum_{i=1}^{n}\xi_i, \quad \text{s.t. } \xi_i\geq 0,\ \forall i$$
$$\frac{1}{2}[F(\mathbf{x}_i,\mathbf{y}_i;\lambda) - F(\mathbf{x}_i,\mathbf{y};\lambda)] \geq 1-\xi_i, \quad \forall i,\mathbf{y}\neq\mathbf{y}_i\,.$$

Notice that the optimal solution of the slack variables is implicitly determined by the weights λ, $\xi_i(\lambda) = \max\{0, 1-\gamma_i(\lambda)\}$.

Proposition 3. *The risk* $\mathcal{R}^{svm}(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\xi_i(\lambda)$ *is an upper bound on the sequence classification loss.*

Proof. (i) If $\xi_i(\lambda) < 1$, then one gets $F(\mathbf{x}_i, \mathbf{y}_i; \lambda) - \max_{\mathbf{y} \neq \mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y}; \lambda) = \gamma(\mathbf{x}_i, \mathbf{y}_i) > 0$ which means the data point is correctly classified and $[\![f(\mathbf{x}_i; \lambda) \neq y_i]\!] = 0 \leq \xi_i(\lambda)$.
(ii) If $\xi_i(\lambda) \geq 1$, then is automatically an upper bound, since $[\![f(\mathbf{x}_i; \lambda) \neq y_i]\!] \leq 1 \leq \xi_i(\lambda)$. □

As an alternative to SVM_1, one can also introduce one slack variable for every training instance and every sequence $\mathbf{y}$, leading to a similar QP, SVM_2, with slack variables $\xi_{i\mathbf{y}}(\lambda) = \max\{0, 1 - [F(\mathbf{x}_i, \mathbf{y}_i) - F(\mathbf{x}_i, \mathbf{y})]/2\}$. This provides an upper bound on the rank loss:

Proposition 4. $\frac{1}{n} \sum_{i=1}^{n} w(T_i) \sum_{\mathbf{y} \neq \mathbf{y}_i} \xi_{i\mathbf{y}}(\lambda) \geq \mathcal{R}^{rk}(\lambda, \mathbf{w})$.

Proof. (i) If $\xi_{i\mathbf{y}}(\lambda) < 1$, then $F(\mathbf{x}_i, \mathbf{y}_i; \lambda) > F(\mathbf{x}_i, \mathbf{y}; \lambda)$ which implies that $\mathbf{y}$ is ranked lower than $\mathbf{y}_i$, in which case $\xi_{i\mathbf{y}}(\lambda) \geq 0$ establishes the bound.
(ii) If $\xi_{i\mathbf{y}}(\lambda) \geq 1$, then the bound holds trivially, since the contribution of every pair $(\mathbf{x}_i, \mathbf{y})$ to $\mathcal{R}^{\text{rk}}$ can be at most 1. □

Comparing SVM_1 and SVM_2, notice that we expect the number of active inequalities in SVM_1 to be much smaller compared to SVM_2, since SVM_1 only penalizes the *largest* margin violation for each example. While this is a data-dependent, i.e. empirical assertion, it is of great practical relevance and has led us to focus on the sparser SVM_1 formulation.

The main computational challenge in optimizing SVM_1 is posed by the extremely large number of linear inequalities, scaling exponentially with the length of the sequences. However, one can reasonably expect that only a very tiny fraction of inequalities will be active at the solution. Hence, we propose to use a row selection or working set procedure to incrementally add inequalities to the problem. Along these lines, we first derive the dual QP with Lagrange multipliers $\alpha_{i\mathbf{y}}$ for every margin inequality.

$$\textbf{DSVM}_1 \quad \max_{\alpha} \frac{1}{2} \sum_{i,\mathbf{y}} \alpha_{i\mathbf{y}} \left[1 - \sum_{j,\bar{\mathbf{y}}} \alpha_{j\bar{\mathbf{y}}} z_{i\mathbf{y}} z_{j\bar{\mathbf{y}}} K_{i,j}(\mathbf{y}, \bar{\mathbf{y}}) \right]$$
$$\text{s.t. } \alpha_{i\mathbf{y}} \geq 0, \ \forall i, \mathbf{y}; \ \sum_{\mathbf{y}} \alpha_{i\mathbf{y}} \leq C, \ \sum_{\mathbf{y}} z_{i\mathbf{y}} \alpha_{i\mathbf{y}} = 0, \forall i$$

Here $z_{i\mathbf{y}}$ denotes a binary pseudo-label, i.e. $z_{i\mathbf{y}} = 1$ for $\mathbf{y} = \mathbf{y}_i$ and $z_{i\mathbf{y}} = -1$, otherwise. $K_{i,j}(\mathbf{y}, \bar{\mathbf{y}})$ denotes the inner product between training sequences defined as $K_{i,j}(\mathbf{y}, \bar{\mathbf{y}}) = \sum_k \psi_k(\mathbf{x}_i, \mathbf{y}) \psi_k(\mathbf{x}_j, \bar{\mathbf{y}})$.

It is important to point out that for features that just involve a single label, one can combine this with an implicit feature representation for observation vectors. Hence with $\psi_k(\mathbf{x}, \mathbf{y}) = \sum_t [\![y_t = \sigma_m]\!] \phi_l(x_{t+r})$ and $k \in I$, where I is the index set over features that combine input attributes with states, one can exploit the identity

$$\sum_{k \in I} \psi_k(\mathbf{x}, \mathbf{y}) \psi_k(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = \sum_{s,t} [\![y_s = \bar{y}_t]\!] K(x_s, \bar{x}_t) \,. \qquad (12)$$

The implications are significant, since this allows to carry over all the advantages of non-linear SVMs to the label sequence case.

As a first step towards solving DSVM_1, we observe that the constraints only couple $\alpha_{i\mathbf{y}}$ for the same training instance i. Hence we can adapt the strategy proposed in [11] and optimize over a subspace associated with a particular training instance, while keeping the remaining variables fixed. Secondly, we propose to maintain an active set of label sequences, S_i, for every instance. The full algorithm is shown in Algorithm 1. In order to perform step 4, we use a two-best Viterbi decoding.

Algorithm 1 Working set optimization for DSVM_1.

1: $S_i \leftarrow \{\mathbf{y}_i\}, \alpha_i = 0, \forall i$
2: **repeat**
3: **for** $i = 1, \ldots, n$ **do**
4: compute $\hat{\mathbf{y}} = \arg\max_{\mathbf{y} \neq \mathbf{y}_i} F(\mathbf{x}_i, \mathbf{y}; \alpha)$
5: **if** $F(\mathbf{x}_i, \mathbf{y}_i; \alpha) - F(\mathbf{x}_i, \hat{\mathbf{y}}; \alpha) < 2$ **then**
6: $S_i \leftarrow S_i \cup \{\hat{\mathbf{y}}\}$
7: optimize DSVM_1 over $\{\alpha_{i\mathbf{y}} | \mathbf{y} \in S_i\}$
8: **end if**
9: remove $\mathbf{y} \in S_i$ with $\alpha_{i\mathbf{y}} < \epsilon$
10: **end for**
11: **until** converged

6. Label Sequence AdaBoost

As a second large margin method, we present a generalization of AdaBoost to label sequence learning. Following [12], our starting point will be the following exponential risk function

$$\mathcal{R}^{\text{exp}}(\lambda, w) \equiv \sum_{i=1}^{n} w(T_i) \sum_{\mathbf{y} \neq \mathbf{y}_i} e^{F(\mathbf{x}_i, \mathbf{y}; \lambda) - F(\mathbf{x}_i, \mathbf{y}_i; \lambda)} \qquad (13)$$

Proposition 5. *The exponential risk is an upper bound on the rank risk, $\mathcal{R}^{rnk} \leq \mathcal{R}^{exp}$.*

Proof. (i) If $F(\mathbf{x}_i, \mathbf{y}_i; \lambda) > F(\mathbf{x}_i, \mathbf{y}; \lambda)$ then $[\![F(\mathbf{x}_i, \mathbf{y}; \lambda) \geq F(\mathbf{x}_i, \mathbf{y}_i; \lambda)]\!] = 0 \leq e^z$ for any z.
(ii) Otherwise, $[\![F(\mathbf{x}_i, \mathbf{y}; \lambda) \geq F(\mathbf{x}_i, \mathbf{y}_i; \lambda)]\!] = 1 = e^0 \leq e^{F(\mathbf{x}_i, \mathbf{y}; \lambda) - F(\mathbf{x}_i, \mathbf{y}_i; \lambda)}$.
Performing a weighted sum over all instances and label sequences $\mathbf{y}$ completes the proof. □

One way of minimizing the exponential loss in Eq. (13) is by gradient-based methods [2], but here we will also outline the derivation of a boosting algorithm that generalizes the AdaBoost.MR algorithm for multiclass classification [1]. We identify Σ^{T_i} with a set of possible super-labels for $\mathbf{x}_i$ and define a sequence of distributions $D_r(i, \mathbf{y})$ over $(\mathbf{x}_i, \mathbf{y})$ pairs recursively as follows:

$$D_{r+1}(i, \mathbf{y}) \equiv \frac{D_r(i, \mathbf{y})}{Z_r} e^{\triangle\lambda_k (\psi_k(\mathbf{x}_i, \mathbf{y}) - \psi_k(\mathbf{x}_i, \mathbf{y}_i))} . \qquad (14)$$

Here $k = k(r)$ denotes the feature selected in the r-th round, $\triangle\lambda_k$ is the corresponding update increment and Z_r the normalization constant. We initialize $D_0(i, \mathbf{y}) = \frac{w(T_i)}{(|\Sigma|^{T_i} - 1) \sum_j w(T_j)}$. After R rounds of boosting, the parameters vector is given by $\lambda_k = \sum_{r=1}^{R} \triangle\lambda_{k(r)}$.

Proposition 6. *For any number of rounds R, $\mathcal{R}^{rk} \leq \prod_{r=1}^{R} Z_r$.*

Proof. [1, Theorem 6] □

Hence, one may greedily optimize the upper bound by selecting at every round a feature k leading to the minimal Z_r. As discussed in [2] the parallel computation of Z_r for all k using dynamic programming is usually inefficient. Instead, we propose to compute an upper bound on Z_r and use these upper bounds for selecting features in every round of boosting.

The basic idea is to use the following inequality valid for x, $x_0 \leq x \leq x_1$, leading to a bound that is linear in x,

$$e^x \leq e^{x_0} \frac{x_1 - x}{x_1 - x_0} + e^{x_1} \frac{x - x_0}{x_1 - x_0} \,, \qquad (15)$$

which results in

$$Z_{r+1} \leq a_k e^{\triangle\lambda_k \psi_{ik}^{min}} + (1 - a_k) e^{\triangle\lambda_k \psi_{ik}^{max}} \tag{16a}$$

$$a_k = \sum_{i,\mathbf{y}\neq\mathbf{y}_i} D_r(i,\mathbf{y}) \frac{\psi_{ik}^{max} - \psi_k(\mathbf{x}_i,\mathbf{y})}{\psi_{ik}^{max} - \psi_{ik}^{min}}, \tag{16b}$$

where ψ_{ik}^{max} and ψ_{ik}^{min} are an upper and lower bound on the value of feature $\psi_k(\mathbf{x}_i,\mathbf{y})$ taken over all $\mathbf{y}$. The left hand side of Eq. (16a) can be minimized analytically with respect to $\triangle\lambda_k$ to give the tightest bound. The index k achieving the smallest upper bound is selected.

We would like to point out that all the quantities involved (such as a_k) can be computed for all features simultaneously with a single dynamic programming run per sequence [2].

Algorithm 2 Label sequence AdaBoost.MR.

1: initialize $D_0(i,\mathbf{y}) = \frac{w(T_i)}{(|\Sigma|^{T_i}-1)\sum_j w(T_j)}$, $D_0(\mathbf{x}_i,\mathbf{y}_i) = 0$
2: initialize $\lambda = 0$
3: **for** $r = 1,\ldots,R$ **do**
4: perform dynamic programming to compute $\{a_k\}$
5: select k minimizing the upper bound in Eq. (16a)
6: compute optimal increment $\triangle\lambda_k$
7: update weight $\lambda_k \leftarrow \lambda_k + \triangle\lambda_k$
8: update D_{r+1} using Eq. (14)
9: **end for**

7. Applications and Experiments

We report experiments on two applications: named entity recognition (NER) and part-of-speech tagging (POS). For the first task we generated a sub-corpus consisting of 300 sentences from the Spanish news wire article corpus which was provided for the Special Session of limited to CoNLL2002 on NER. The label set in this corpus consist of non-name and the beginning and continuation of person names, organizations, locations and miscellaneous names, resulting in a total of $|\Sigma| = 9$ different labels. For the tagging application we extracted a corpus consisting of 300 sentences from the Penn TreeBank corpus. The total number of function tags was $|\Sigma| = 45$. All input features are simple binary features. Most features are indicator functions for a word occurring within a fixed size window centered on the word being labeled. In addition there are features that encode morphological properties, e.g. spelling features.

Table 1 summarizes the experimental results. We have trained standard HMMs as well as HMMs with overlapping features. As can be seen, all discriminative methods outperform HMMs. HM-SVMs overall achieve the best accuracy values. Boosting performs somewhat worse than the other methods, but has the advantage to lead to a sparse solution, which may have additional advantages in real-time settings.

8. Conclusion

We surveyed discriminative methods for label sequence learning and presented generalizations of large margin methods, SVM and AdaBoost. These methods combine the advantages of large margin methods with the elegance and efficiency of HMMs. Our experiments prove the competitiveness of these methods on two benchmark sets. We are currently working on a large-scale experimental evaluation.

	HMM	CRF	MRF	SVM	EXP	Boost
NER	89.13 (91.15)	94.55	94.56	94.92	94.14	92.26
POS	73.60 (77.22)	87.12	87.55	88.16	86.47	85.89

Table 1: Prediction accuracies of various method. HMM: Hidden Markov Model, overlapping features in brackets, CRF: Conditional Random Fields, MRF: Marginal Random Fields, minimizing $\mathcal{R}^{\text{mg}}$, SVM: Hidden Markov Support Vector Machine, EXP: minimizing exponential loss $\mathcal{R}^{\text{exp}}$, Boost: generalization of AdaBoost.

Acknowledgments: This work was sponsored by an NSF-ITR grant, award number IIS-0085940.

9. References

[1] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[2] Y. Altun, T. Hofmann, and M. Johnson, "Discriminative learning for label sequences via boosting," in *Advances in Neural Information Processing Systems (NIPS*15)*, 2003.

[3] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th International Conf. on Machine Learning*. Morgan Kaufmann, San Francisco, CA, 2001, pp. 282–289.

[4] H. Wallach, "Efficient training of conditional random fields," Master's thesis, University of Edinburgh, 2002.

[5] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," in *Proceedings of Human Language Technology-NAACL*, 2003.

[6] T. Minka, "Algorithms for maximum-likelihood logistic regression," CMU, Department of Statistics, TR 758, Tech. Rep., 2001.

[7] M. Collins, "Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms," in *EMNLP*, 2002.

[8] S. Chen and R. Rosenfeld, "A Gaussian prior for smoothing maximum entropy models," Carnegie Mellon University, Tech. Rep. CMUCS-99-108, 1999.

[9] S. Kakade, Y. W. Teh, and S. Roweis, "An alternate objective function for Markovian fields," in *ICML*, 2002.

[10] J. Weston and C. Watkins, "Support vector machines for multi-class pattern recognition," in *Proceedings European Symposium on Artificial Neural Networks*, 1999.

[11] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.

[12] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," Stanford, Statistics Department, Tech. Rep., 1998.