

# Discriminative Sequence Labeling by Z-score Optimization

Elisa Ricci<sup>1</sup>, Tijl de Bie<sup>2</sup>, and Nello Cristianini<sup>2,3</sup>

<sup>1</sup> Dept. of Electronic and Information Engineering, University of Perugia, 06125, Perugia, Italy

`elisa.ricci@diei.unipg.it`,

<sup>2</sup> Dept. of Engineering Mathematics, University of Bristol, Bristol, BS8 1TR, UK  
`tijl.debie@gmail.com`,

<sup>3</sup> Dept. of Computer Science, University of Bristol, Bristol, BS8 1TR, UK  
`nello@support-vector.net`

**Abstract.** We consider a new discriminative learning approach to sequence labeling based on the statistical concept of the  $Z$ -score. Given a training set of pairs of hidden-observed sequences, the task is to determine some parameter values such that the hidden labels can be correctly reconstructed from observations. Maximizing the  $Z$ -score appears to be a very good criterion to solve this problem both theoretically and empirically. We show that the  $Z$ -score is a convex function of the parameters and it can be efficiently computed with dynamic programming methods. In addition to that, the maximization step turns out to be solvable by a simple linear system of equations. Experiments on artificial and real data demonstrate that our approach is very competitive both in terms of speed and accuracy with respect to previous algorithms.

## 1 Introduction

Sequence labeling is one of the most important tasks in many applications, such as in part-of-speech tagging or named entity recognition (NER) in the Natural Language Processing (NLP) field, and gene finding or protein homology detection in bioinformatics. This task represents a generalization of the standard classification problem since prediction is made not only to a single hidden variable, but to a sequence of mutually dependent hidden variables (the labels). Traditionally, Hidden Markov Models (HMMs) have been used for sequence labeling. The HMM conditional probabilities are trained using the maximum likelihood criterium, after which the HMM can be used for prediction by means of the Viterbi algorithm. However, the HMM approach is arguably suboptimal for this task, as it is designed for modeling, rather than for discrimination.

In the last few years, a number of discriminative methods have been proposed to improve the performance achieved by generative HMM based sequence labeling. Recently studied methods include Maximum Entropy Markov Models [6], Conditional Random Fields (CRFs) [5], Hidden Markov Perceptron (HMP) [3], boosting-based algorithms [1] and Maximal Margin (MM) methods [2, 7].

In particular MM algorithms have been shown to provide accurate labeling (see [2] for a comparison between different methods). These methods rely on the definition of a linear score function for observed-hidden sequence pairs. Parameter estimation is performed by imposing that for each observed sequence in the training set, the score of the pair with the correct given hidden sequence should be bigger than the score of all other possible hidden sequences. These conditions can be simply specified by a set of linear constraints. Subject to these constraints, the squared norm of the parameters is minimized, which guarantees that the minimal difference between the score of the correct pair and the closest runner-up is maximal. Clearly, a direct implementation of this strategy would be totally infeasible, as the number of possible hidden sequences associated to an observed one (and hence the number of constraints) is exponential in the length of the sequences. Altun *et al.* [2] have attacked this problem by adding constraints incrementally. With this approach, for each constraint to be added a Viterbi decoding needs to be performed, so it quickly becomes expensive for long sequences and large training sets. In [8] the number of constraints for this method is demonstrated to increase polynomially with the length of the sequences. An exponential number of constraints is required also by the algorithm proposed by Collins [4]. In [7] a different strategy is applied where the optimization problem is reparameterized in terms of marginal variables but a certain number of constraints (scaling linearly with the length of the sequences) is still required.

Since in practical applications data are often nonseparable, in MM methods slack variables (one for each training pair) are introduced to allow some constraints to be violated. With this approach the number of incorrectly reconstructed sequences is minimized. However in these cases, choosing other optimization criteria could be desirable. To this aim, in this paper, we approach the problem from a different perspective. We consider the full distribution of the scores for all possible observed-hidden sequence pairs and we compute the mean and the variance of this distribution as a function of the parameters. Then, we maximize the  $Z$ -score of the correct observed-hidden sequence pairs subject to the parameters, where the  $Z$ -score is defined as the number of standard deviations the score is away from its mean. In this way, the number of incorrect pairs which scores higher than the optimal ones is minimized. Moreover the score of the correct pair is optimally separated from the bulk of all possible scores without considering each of these separately. A crucial observation enabling this strategy is that the  $Z$ -score of any fixed pair can be computed exactly and efficiently as a function of the parameters. This is done by means of a dynamic program analogous to the classical forward algorithm for HMMs. Additionally, we show that the  $Z$ -score is a convex function of the scoring parameters, and consequently it can be maximized exactly by simply solving a linear system.

## 2 Hidden Markov Models

We define a state alphabet set  $\Sigma_y = \{Y_1 \dots Y_{n_s}\}$  and an observation alphabet set  $\Sigma_x = \{X_1 \dots X_{n_o}\}$  and we consider an observed sequence  $\mathbf{x} =$

$(x_1, x_2, \dots, x_m)$ ,  $\mathbf{x} \in \mathcal{X} = \Sigma_x^m$  and the corresponding hidden sequence  $\mathbf{y} = (y_1, y_2, \dots, y_m)$ ,  $\mathbf{y} \in \mathcal{Y} = \Sigma_y^m$ . Formally an HMM is an object  $(E, T)$ . The emission matrix  $E$  stores the probability of observation  $i$  being produced from the state  $j$ , i.e. it is an  $n_o \times n_s$  matrix with elements  $e_{ij} = \log P(X_i|Y_j)$ ,  $1 \leq i \leq n_o$ ,  $1 \leq j \leq n_s$ . The transition matrix  $T$  is the matrix with elements  $t_{ij} = \log P(Y_i|Y_j)$ ,  $1 \leq i, j \leq n_s$ . The probability of a given observed-hidden sequence pair can be computed as:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^m (\log P(y_k|y_{k-1}) + \log P(x_k|y_k))$$

Defining  $\psi_{kk-1}^{ij} = \log P(y_k = Y_i|y_{k-1} = Y_j)$  and  $\psi_k^{ij} = \log P(x_k = X_i|y_k = Y_j)$ , the scoring function can be rewritten as:

$$s(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^m \left( \sum_{i,j=1}^{n_s} \psi_{kk-1}^{ij} I_{kk-1}^{ij} + \sum_{i=1}^{n_o} \sum_{j=1}^{n_s} \psi_k^{ij} I_k^{ij} \right) = \sum_{i,j=1}^{n_s} t_{ij} C_t^{ij} + \sum_{i=1}^{n_o} \sum_{j=1}^{n_s} e_{ij} C_e^{ij}$$

where  $I_{kk-1}^{ij}$  is equal to 1 if the  $k$ -th hidden label is  $Y_i$  and the  $(k-1)$ -th label is  $Y_j$  and analogously  $I_k^{ij} = 1$  means that the  $k$ -th observation is  $X_i$  and the associated label is  $Y_j$ . Therefore  $C_t^{ij} = \#(y_k = Y_i|y_{k-1} = Y_j)$  and  $C_e^{ij} = \#(x_k = X_i|y_k = Y_j)$  count the number of each of the possible transitions and emissions.

For notational convenience, define the vector of parameters  $\boldsymbol{\theta} \in R^d$ ,  $\boldsymbol{\theta} = [e_{11} \dots e_{n_o n_s} t_{11} \dots t_{n_s n_s}]^T$ , with  $d = n_s n_o + n_s n_s$ . Correspondingly, for a given pair of observed and hidden sequences  $(\mathbf{x}, \mathbf{y})$  define a vector  $\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) = [C_e^{11} \dots C_e^{n_o n_s} C_t^{11} \dots C_t^{n_s n_s}]^T \in R^d$  containing the sufficient statistics associated to each parameter. Then we can express the scoring function  $s(\mathbf{x}, \mathbf{y})$  as a linear function of the parameters  $\boldsymbol{\theta}$ :

$$s(\mathbf{x}, \mathbf{y}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y})$$

We call  $s(\mathbf{x}, \mathbf{y})$  the *score* associated to the observed-hidden sequence pair  $(\mathbf{x}, \mathbf{y})$ .

Once the parameters are fixed, the actual labeling task predicts the hidden state sequence  $\bar{\mathbf{y}}$  that is most likely in conjunction with the given observation sequence  $\mathbf{x}$ . Hence, labeling is done by solving  $h(\mathbf{x}) = \arg \max_{\mathbf{y}} s(\mathbf{x}, \mathbf{y})$ . A brute force calculation of  $h(\mathbf{x})$  is intractable for realistic problems, as the number  $N$  of possible assignments in  $\mathcal{Y}$  is exponential in the length of the sequences  $m$ . However such prediction can be done efficiently by the Viterbi algorithm. In the following the score of the optimal pair will be denoted by  $s(\mathbf{x}, \bar{\mathbf{y}})$ .

### 3 The Z-score

Given  $\mathbf{x}$ , we can consider the mean values  $\mu(\mathbf{x})$  of the scores of all possible  $N$  hidden sequences  $\mathbf{y}_j$  and show that it is also a linear function:

$$\mu(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^N \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}_j) = \boldsymbol{\theta}^T \boldsymbol{\mu}_\phi$$

where  $\boldsymbol{\mu}_\phi = [\mu_1 \dots \mu_d]^T$  is the vector with components given by the average values of the components of  $\phi(\mathbf{x}, \mathbf{y}_j)$ . Similarly, for the variance  $\sigma^2(\mathbf{x})$  we have:

$$\sigma^2(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^N (\boldsymbol{\theta}^T \phi(\mathbf{x}, \mathbf{y}_j) - \mu(\mathbf{x}))^2 = \boldsymbol{\theta}^T \mathbf{C} \boldsymbol{\theta}$$

The matrix  $\mathbf{C}$  is a covariance matrix with elements:

$$c_{pq} = \frac{1}{N} \sum_{j=1}^N \phi_p(\mathbf{x}, \mathbf{y}_j) \phi_q(\mathbf{x}, \mathbf{y}_j) - \mu_p \mu_q = v_{pq} - \mu_p \mu_q \quad (1)$$

where  $1 \leq p, q \leq d$ . Based on this mean and variance, expressed in terms of the parameters  $\boldsymbol{\theta}$ , we can now define the  $Z$ -score parameterized by  $\boldsymbol{\theta}$ :

**Definition 1.** Let  $\mu(\mathbf{x})$  and  $\sigma^2(\mathbf{x})$  be the mean and the variance of the scores for all possible hidden sequences generating  $\mathbf{x}$ . We define the  $Z$ -score  $Z(\mathbf{x})$ :

$$Z(\mathbf{x}) = \frac{s(\mathbf{x}, \bar{\mathbf{y}}) - \mu(\mathbf{x})}{\sigma(\mathbf{x})} = \frac{\boldsymbol{\theta}^T \mathbf{b}}{\sqrt{\boldsymbol{\theta}^T \mathbf{C} \boldsymbol{\theta}}} \quad (2)$$

where the right expression is obtained with  $\mathbf{b} = \phi(\mathbf{x}, \bar{\mathbf{y}}) - \boldsymbol{\mu}_\phi$ .

In general, we are interested in computing the  $Z$ -score for a set of  $\ell$  pairs of sequences  $S = \{(\mathbf{x}_1, \bar{\mathbf{y}}_1)(\mathbf{x}_2, \bar{\mathbf{y}}_2) \dots (\mathbf{x}_\ell, \bar{\mathbf{y}}_\ell)\}$ . In such cases, we can define the global score as the sum of the scores for each sequence pair in the set. Its mean is the sum of the means for all sequence pairs  $(\mathbf{x}_i, \bar{\mathbf{y}}_i)$  separately, and can be summarized by  $\mathbf{b}^* = \sum_i \mathbf{b}_i$ . Similarly, the covariance of the sum of (independent) scores is the sum of the covariances:  $\mathbf{C}^* = \sum_i \mathbf{C}_i$ . Hence, the  $Z$ -score can be extended to the case where there is more than one given sequence pair by using for  $\mathbf{b}^*$  and  $\mathbf{C}^*$  instead of  $\mathbf{b}$  and  $\mathbf{C}$  in Eqn. 2 above.

We will now proceed to show that  $\mathbf{C}^*$  and  $\mathbf{b}^*$  can be computed efficiently. Then we will show that based on these the  $Z$ -score can be maximized efficiently, and that it represents a theoretically and empirically interesting criterium for discriminative sequence labeling.

## 4 Computing the $Z$ -score as a Function of the Parameters

In this section we show how the elements of  $\mathbf{b}$  and  $\mathbf{C}$  can be computed exactly and efficiently by dynamic programming (DP) routines. Therefore the  $Z$ -score can be fully determined, as a function of the parameter vector  $\boldsymbol{\theta}$ .

**Proposition 1.** Each element of the vector  $\mathbf{b}$  and of the matrix  $\mathbf{C}$  can be computed in a time  $O(mn_s^2)$ .

*Outline of proof.* We consider a pair of observed-hidden sequences  $(\mathbf{x}, \bar{\mathbf{y}})$ . The vector  $\mathbf{b}$  is given by  $\mathbf{b} = \phi(\mathbf{x}, \bar{\mathbf{y}}) - \boldsymbol{\mu}_\phi$ . The first term can be calculated computing the statistics associated to each parameter.

---

**Algorithm 1** Dynamic programming algorithm to compute  $\mu_k$ ,  $1 \leq k \leq n_s n_o$

---

```

1: Input:  $\mathbf{x} = (x_1, x_2, \dots, x_m)$ ,  $p$ ,  $q$ .
2:
3:  $\pi(i, 1) := 1 \quad \forall i$ 
4: if  $q = x_1 \wedge p = i$ ,  $\mu_{pq}^e(i, 1) := 1$ 
5: for  $j = 2$  to  $m$ 
6:   for  $i = 1$  to  $n_s$ 
7:      $M := 0$ 
8:      $\pi(i, j) := \sum_i \pi(i, j-1)$ 
9:     if  $q = x_j \wedge p = i$ ,  $M := 1$ 
10:     $\mu_{pq}^e(i, j) := \frac{\sum_i (\mu_{pq}^e(i, j-1) + M) \pi(i, j-1)}{\pi(i, j)}$ 
11:   end
12: end
13:
14: Output:  $\frac{\sum_i \mu_{pq}^e(i, m)}{\sum_i \pi(i, m)}$ 

```

---

The elements of the vector  $\boldsymbol{\mu}_\phi$  can be obtained with DP routines. We construct the vector  $\boldsymbol{\mu}_\phi$  such that its first  $n_s n_o$  elements contain the mean values associated with the emission probabilities. Each value can be determined by Algorithm 1. Here a  $n_s \times m$  DP table  $\mu_{pq}^e$  is considered and initialized to zero values. The index  $p$  denotes the hidden state ( $1 \leq p \leq n_s$ ) and  $q$  refers to the observation ( $1 \leq q \leq n_o$ ). For example the first component of  $\boldsymbol{\mu}_\phi$  corresponds to the DP matrix  $\mu_{11}^e$ . First a  $n_s \times m$  matrix  $\boldsymbol{\pi}$  is progressively filled. Each cell  $\pi(i, j)$  contains the number of all possible paths from the origin of the trellis to position  $(i, j)$ . Then a recursive relation is considered to compute each element of the matrix  $\mu_{pq}^e$ . Further details are shown in Algorithm 1.

A similar algorithm is adopted to compute the mean values for the transition probabilities. A DP matrix  $\mu_{pz}^t$  is filled, where  $1 \leq p, z \leq n_s$ . The only difference is the recursive formula that can be found in Algorithm 2.

Analogously the elements of the covariance matrix  $\mathbf{C}$  can be computed. We have five sets of values: variances of emission probabilities ( $c_{pq}^e$ ,  $1 \leq p \leq n_s, 1 \leq q \leq n_o$ ), variances of transition probabilities ( $c_{pz}^t$ ,  $1 \leq p, z \leq n_s$ ), covariances of emission probabilities ( $c_{pp'q'q'}^e$ ,  $1 \leq p, p' \leq n_s, 1 \leq q, q' \leq n_o$ ), covariances of transition probabilities ( $c_{pp'z'z'}^t$ ,  $1 \leq p, p', z, z' \leq n_s$ ) and mixed covariances ( $c_{pp'z}^{et}$ ,  $1 \leq p, p', z \leq n_s, 1 \leq q \leq n_o$ ). To determine each of them we consider Eqn. 1. It suffices to calculate the values  $v_{pq}$  since the mean values are already known. The computation of  $v_{pq}$  is again performed following Algorithm 1 but with recursive relations given in Algorithm 2.

**Computational Cost Analysis.** In general the calculation of the matrices  $\mathbf{b}$  and  $\mathbf{C}$  requires running a DP algorithm like Algorithm 1 respectively  $d$  times for mean values and  $d^2$  times for the covariance matrix. Hence the overall computational cost considerably increases for large  $d$ . However, most of the DP routines are redundant since many cells of  $\mathbf{b}$  and  $\mathbf{C}$  have the same values. In particular:

---

**Algorithm 2** Extra formulas for computing mean and variance values
 

---

$$\begin{aligned}
 &9: \text{if } z = i, M := 1 \\
 &10: \mu_{pz}^t(i, j) := \frac{\sum_i \mu_{pz}^t(i, j-1) \pi(i, j-1) + M \pi(p, j-1)}{\pi(i, j)} \\
 &4: \text{if } q = x_1 \wedge p = i, v_{pq}^e(i, 1) := 1 \\
 &9: \text{if } q = x_j \wedge p = i, M := 1 \\
 &10: v_{pq}^e(i, j) := \frac{\sum_i (v_{pq}^e(i, j-1) + 2M \mu_{pq}^e(i, j-1) + M) \pi(i, j-1)}{\pi(i, j)} \\
 &9: \text{if } q' = x_j \wedge p' = i, M_1 := 1 \\
 &\text{if } q = x_j \wedge p = i, M_2 := 1 \\
 &10: v_{pp'q'q'}^e(i, j) := \frac{\sum_i (v_{pp'q'q'}^e(i, j-1) + M_1 \mu_{pq}^e(i, j-1) + M_2 \mu_{p'q'}^e(i, j-1)) \pi(i, j-1)}{\pi(i, j)} \\
 &4: \text{if } p = i, v_{pz}^t(i, 2) = 1 \\
 &9: \text{if } p = i, M := 1 \\
 &10: v_{pz}^t(i, j) := \frac{\sum_i v_{pz}^t(i, j-1) \pi(i, j-1) + (2M \mu_{pz}^t(p, j-1) + M) \pi(p, j-1)}{\pi(i, j)} \\
 &9: \text{if } p' = j, M_1 := 1 \\
 &\text{if } p = j, M_2 := 1 \\
 &10: v_{pzp'z'}^t(i, j) := \frac{\sum_i v_{pzp'z'}^t(i, j-1) \pi(i, j-1) + M_1 \mu_{pz}^t(p', j-1) \pi(p', j-1) + M_2 \mu_{p'z'}^t(p, j-1) \pi(p, j-1)}{\pi(i, j)} \\
 &9: \text{if } z' = i, M_1 := 1 \\
 &\text{if } q = x_j \wedge p = i, M_2 := 1 \\
 &10: v_{pqp'z}^{et}(i, j) := \frac{\sum_i v_{pqp'z}^{et}(i, j-1) \pi(i, j-1) + M_1 \mu_{pq}^e(p', j-1) \pi(p', j-1) + M_2 \mu_{p'z'}^t(p, j) \pi(p, j)}{\pi(i, j)}
 \end{aligned}$$


---

**Proposition 2.** *The number of dynamic programming routines required to calculate  $\mathbf{b}$  and  $\mathbf{C}$  increases linearly with the size  $n_o$  of the observation alphabet.*

*Outline of proof.* In the mean vector  $\boldsymbol{\mu}_\phi$  there are  $n_o + 1$  different values. All the elements associated to transition probabilities assume the same values while for emission probability  $\mu_{pq}^e = \mu_{ef}^e, \forall q = f$ .

The covariance matrix  $\mathbf{C}$  is a symmetric block matrix made basically by three components: the block associated to emission probabilities, that of transition probabilities and that relative to mixed terms. To compute it  $6n_o + 5$  DP routines are required. In the emission part there are  $2n_o$  possible different values since  $c_{pq}^e = c_{ef}^e, \forall q = f, c_{pp'q'q'}^e = 0, \forall q \neq q'$  and  $c_{pp'q'q'}^e = c_{efef'}^e, \forall q = q' = f = f'$ . In the transition block there are only 5 possible different values. In particular for the variances, it is  $c_{pz}^t = c_{eg}^t, \forall p = z = e = g$  and  $c_{pz}^t = c_{eg}^t, \forall p = e, z = g$  and  $p \neq z$ . The other three values are associated to covariances since  $c_{pzp'z'}^t = 0, \forall p \neq p', z \neq z', c_{pzp'z'}^t = c_{egeg'}^t, \forall p = p', z \neq z', e = e', g \neq g'$  and  $c_{pzp'z'}^t = c_{egeg'}^t, \forall p \neq p', z = z', e \neq e', g = g'$ . The block relative to mixed terms is made of  $4n_o$  possible different value. In fact there are  $n_o$  values  $c_{pqp'z}^{et}$  with  $p = p' = z'$ ,  $n_o$  values  $c_{pqp'z}^{et}$ , with  $p = p', p' \neq z', n_o$  values  $c_{pqp'z}^{et}$ , with  $p = z', p' \neq z'$  and  $n_o$  values  $c_{pqp'z}^{et}$ , with  $p \neq p', z \neq z'$ .

#### 4.1 Dealing with Arbitrary Features

A nice property of our method is that it can be easily extended to the case of arbitrary features. In general the vector  $\phi(\mathbf{x}, \mathbf{y})$  contains not only statistics associated to transition and emission probabilities but also any feature that reflects the properties of the objects represented by the nodes of the HMM. For example in most of the NLP tasks, observations are words and the problem is to assign opportune labels to them. In this case, feature vectors can contain information about the occurrence of a certain word in a sentence as well as about its spelling properties (e.g. if the word is capitalized, if it contains numerical symbols). Sometimes also overlapping features [5] are needed, i.e. features that indicate relations between observations and some previous and future labels. It means that at instant  $k$  all the indicator functions  $I_s^{ij}$  are considered, with  $k - w \leq s \leq k + w$  where  $w$  is the size of a window around the  $k$ -th observation. In this way it is possible to deal with high order dependencies between labels.

To compute the  $Z$ -score in these situations the derivation of appropriate formulas similar to those of  $\mu_{pq}^e$ ,  $c_{pq}^e$  and  $c_{pp'z}^{et}$  is straightforward. It suffices to set the values  $M$ ,  $M_1$  and  $M_2$  equal to 1 when the considered features are active. For example, if  $\mathbf{x}$  represents a sequence of words and we want to compute the mean for the feature “*The word is capitalized*”, we use Algorithm 1 with the parameter  $M$  equal to 1 in correspondence to observation  $x_i$  if the first letter of  $x_i$  is an upper case letter. Unfortunately the computational cost increases with the number of features since the number of different parameters in the matrix  $\mathbf{C}^*$  scales quadratically with the observations alphabet size  $n_o$ . However we show that in this case approximate algorithms can be used to obtain close estimates of the mean and the variance values with a significantly reduced computational cost. The experiments reported in the last section support this claim.

## 5 Z-score Optimization

Suppose we have a training set of pairs of observed and hidden sequences  $S = \{(\mathbf{x}_1, \bar{\mathbf{y}}_1)(\mathbf{x}_2, \bar{\mathbf{y}}_2) \dots (\mathbf{x}_\ell, \bar{\mathbf{y}}_\ell)\}$ . One often considers the task to find the parameter values  $\theta$  such that the optimal sequence of hidden states  $\bar{\mathbf{y}}_i$  can be reconstructed from  $\mathbf{x}_i$ ,  $\forall 1 \leq i \leq \ell$ . In formulas this condition can be expressed as:

$$\theta^T \phi(\mathbf{x}_i, \bar{\mathbf{y}}_i) \geq \theta^T \phi(\mathbf{x}_i, \mathbf{y}_i) \quad \forall \mathbf{y}_i \neq \bar{\mathbf{y}}_i, \quad \forall 1 \leq i \leq \ell \quad (3)$$

This set of constraints defines a convex set in the parameter space and its number is exponential in the length of the sequences. To obtain an optimal vector  $\theta$  that successfully fulfills (3) an optimization problem can be considered, i.e. a suitable objective function must be chosen. Several choices are possible. For example in [1, 7] a maximal margin solution is considered: between all possible values of  $\theta$  such that (3) is verified, they pick the values such that the highest scoring sequence is maximally separated from the others. Interestingly, with this approach, an upper bound on the zero-one error (i.e. on the number of incorrectly reconstructed

sequences) is minimized. Similarly CRFs [5] use a conditional likelihood criterion to minimize a different upper bound on this loss.

Here a different philosophy is investigated, which we believe to be more appropriate in the cases where there exists no parameter setting for which the given pairs are optimal. Our purpose is to minimize the number of incorrect pairs that are ranked higher than the correct one. To this aim we choose as objective function the  $Z$ -score since we are motivated by statistical reasoning. The  $Z$ -score can be regarded as a measure of ranking quality. To give an intuition, a pair of sequences  $(\mathbf{x}, \mathbf{y})$  corresponds to a high  $Z$ -score if few other pairs have probability of having a higher score. On the other hand, a small  $Z$ -score means a low position of the given  $(\mathbf{x}, \mathbf{y})$  in the ranking associated with that scoring model. Interestingly, under normality assumptions, this  $Z$ -score is directly equivalent to a  $p$ -value. Hence, maximizing the  $Z$ -score can be interpreted as maximizing the significance of the correct pair score: the larger the  $Z$ -score, the more significant it is, or the more different it is from the majority of others pairs. Intriguingly, we can interpret the  $Z$ -score maximization as a special case of Fisher’s discriminant analysis (FDA), where one class reduces to a single data point: we consider the distribution of all possible scores and contrast this with the ‘distribution’ of the score for the given training example (which is obviously non-zero only for one value). Therefore learning theory applicable to FDA would be directly translated to our algorithm. (We will not go into this aspect in the present paper).

Following the definition of  $Z$ -score given at the end of Section 3, the optimization problem we are interested in is:

$$\max_{\boldsymbol{\theta}} \frac{\boldsymbol{\theta}^T \mathbf{b}^*}{\sqrt{\boldsymbol{\theta}^T \mathbf{C}^* \boldsymbol{\theta}}} \quad (4)$$

We note that,  $\mathbf{C}^*$  being a positive semidefinite matrix, the objective function is convex and the problem admits a global solution. We can find the optimal  $\boldsymbol{\theta}$  simply by inverting the covariance matrix ( $\boldsymbol{\theta} = \mathbf{C}^{*-1} \mathbf{b}^*$ ). Alternatively, considering the invariance of the problem to positive rescaling and the monotonicity of the square root, (4) becomes:

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{2} \boldsymbol{\theta}^T \mathbf{C}^* \boldsymbol{\theta} \\ \text{s.t.} \quad & \boldsymbol{\theta}^T \mathbf{b}^* \geq 1 \end{aligned} \quad (5)$$

Classical Lagrangian duality enables the primal problem (5) to be transformed into the associated dual, which can be easier to solve for large values of  $d$ . It is simple to verify that the dual problem is:

$$\max_{\boldsymbol{\alpha} \geq 0} \quad -\frac{1}{2} \boldsymbol{\alpha}^T \mathbf{J} \boldsymbol{\alpha} + \mathbf{h} \boldsymbol{\alpha} \quad (6)$$

where we have defined  $\mathbf{J} = \mathbf{b}^{*T} \mathbf{C}^{*-1} \mathbf{b}^*$  and  $\mathbf{h} = 1$ . The solution of the primal is given by  $\boldsymbol{\theta} = \mathbf{C}^{*-1} \mathbf{b}^* \boldsymbol{\alpha}$ . The computational cost in the optimization phase is dominated by the inversion of the matrix  $\mathbf{C}^*$ . General matrix inversions usually take  $O(d^3)$  time. However since  $\mathbf{C}^*$  is a symmetric positive definite matrix the use of iterative methods as conjugate gradient greatly speed up the computation.

### 5.1 Incorporating Hamming Loss Function

Imposing constraints (3) a zero-one loss  $\ell_{0/1}(\mathbf{y}, \mathbf{y}') = I(\mathbf{y} \neq \mathbf{y}')$  is implicitly considered: unfortunately  $\ell_{0/1}(\mathbf{y}, \mathbf{y}')$  is 1 if the complete sequence is not labeled correctly, both when the entire sequence is wrong and when only one label is predicted incorrectly. A better loss function that discriminates between similar pairs of sequences and very different ones, is the Hamming loss  $\ell_H(\mathbf{y}, \mathbf{y}') = \sum_i I(y_i \neq y'_i)$ . Originally proposed in [7] for MM algorithms it can be also used in our method. For each pair of sequences  $(\mathbf{x}, \mathbf{y})$  we consider the score:

$$s(\mathbf{x}, \mathbf{y}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) + \ell_H(\mathbf{y}, \bar{\mathbf{y}}) = \boldsymbol{\theta}'^T \boldsymbol{\phi}'(\mathbf{x}, \mathbf{y})$$

where we define the vectors  $\boldsymbol{\theta}'^T = [\boldsymbol{\theta} \ 1]^T$  and  $\boldsymbol{\phi}'(\mathbf{x}, \mathbf{y})^T = [\boldsymbol{\phi}(\mathbf{x}, \mathbf{y}) \ \ell_H(\mathbf{y}, \bar{\mathbf{y}})]^T$ . It is easy to verify that the associate  $Z$ -score turns out into a dual convex optimization problem with the same form of (6), with  $\mathbf{h} = 1 - \mu_\ell - \mathbf{b}^{*T} \mathbf{C}^{*-1} \mathbf{c}_\ell$ . The optimal vector of parameters in the primal is  $\boldsymbol{\theta} = \mathbf{C}^{*-1}(\mathbf{b}^* \boldsymbol{\alpha} - \mathbf{c}_\ell)$ , which is used to perform decoding with Viterbi algorithm. Here  $\mu_\ell$  represents the mean value of the terms  $\ell_H(\mathbf{y}, \bar{\mathbf{y}})$  computed along all possible paths while  $\mathbf{c}_\ell$  is the vector containing the covariance values between the loss term and all the other parameters. The computation of  $\mu_\ell$  and  $\mathbf{c}_\ell$  is realized with Algorithm 1 and recursive relations similar to those in Algorithm 2. For example the value  $\mu_\ell$  is computed with Algorithm 1 with the only difference that at line 11  $M = 1$  if  $y_j \neq \bar{y}_j$ . It is worth noting that in general every loss function that can be computed by DP can be used in our method.

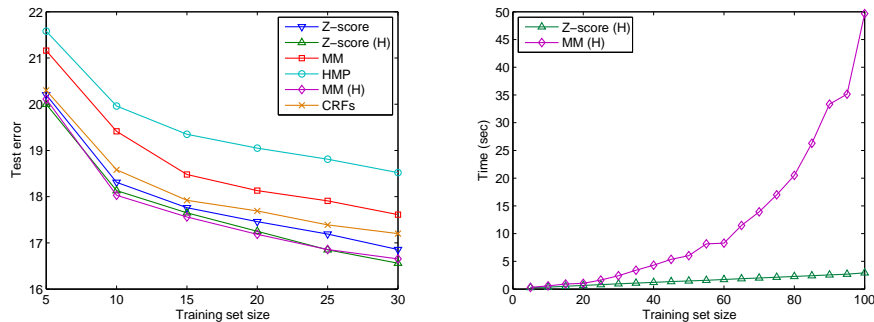
## 6 Experimental Results

**Artificial Data.** In the first series of experiments our method has been tested with artificial data. An HMM with  $n_s = 3$ ,  $n_o = 4$  has been considered. The parameters to be determined are the transition and emission probabilities. Sequences with length  $m = 10$  have been generated randomly, so that the optimal parameter vector may not exist. In the experiments the size of training set varies while the number of pairs in the test set is fixed to 100. We compared the performance of our approach with CRFs, and the HMP and the MM method in [2] with linear kernel. For the latter and for the  $Z$ -score a formulation with a Hamming loss is also considered. For MM algorithms the soft margin parameter  $C$  has been set to 0.1 and 1 for the standard and the rescaled margin version respectively, a constant  $\epsilon = 10^{-12}$  specifies the accuracy for constraints to be satisfied. The maximum number of iterations of the HMP is  $T = 100$ . The CRFs have been optimized using a conjugate gradient method. Parameter values have been determined by cross-validation. The performance are evaluated in terms of labeling error on test set (average number of misclassified labels) (Fig 1.a). Results are averaged over 100 runs. The simple  $Z$ -score maximization is only slightly outperformed by  $Z$ -score and MM methods with Hamming loss which have comparable performance. For the two latter algorithms we also examine the computational cost. Figure 1.b shows, for the same experiment, the training time

as function of the training set size: our approach is definitely faster, especially for larger datasets.

We performed similar experiments for different datasets and HMMs and we observed that maximizing the  $Z$ -score the performance is comparable or better than MM method for nonseparable data (which is the more common situation with real-life data). In the separable case, we can also impose the constraints (3) with an iterative algorithm similar to that proposed in [2]. In this way labeling accuracy is comparable to MM method but much less constraints are used. We do not report associated simulation results due to lack of space. In terms of computation time our approach is generally much faster.

For very large numbers of parameters, however, the time required to compute  $\mathbf{b}^*$  and  $\mathbf{C}^*$  may exceed the computation time of competing MM approaches. However, in this case, good approximations for  $\mathbf{b}$  and  $\mathbf{C}$  can be used by considering a randomly sampled subset of paths in the trellis, rather than using DP. Results in Table 6 demonstrate this is a valid approach. Here, sequences of length 100 have been considered, the training and test set sizes are 50 and 100. Various HMM models have been used: the hidden alphabet size is fixed to  $n_s = 2$ , while  $n_o$  varies. The average test error and the computation time are reported for the  $Z$ -score method with exact  $\mathbf{b}$  and  $\mathbf{C}$ , when they are computed on a set of 100 random paths, and for the MM method with Hamming loss.



**Fig. 1.** (a) Average number of uncorrect labels and (b) computational time as function of the training set size for an HMM with  $n_s = 3$  and  $n_o = 4$ .

**Named Entity Recognition.** NER is a subtask of information extraction which deals with finding phrases containing person, organization and locations names or temporal and numerical expressions. The experimental setup is similar to [2]. We considered 300 sentences extracted from the Spanish news wire article corpus used for the Special Session of CoNLL-2002 on NER. Our subset contains more than 7000 tokens (about 2000 unique) and each sentence has an average length of 30 words. The hidden alphabet is limited to  $n_s = 9$  different labels,

**Table 1.** Test error. Time (sec) in parenthesis.

$n_o$	Z-SCORE	Z-SCORE (100)	MM (H)
3	15.82 (0.88)	15.91 (0.43)	15.81 (5.63)
5	10.10 (1.28)	10.13 (0.63)	10.02 (8.29)
7	7.49 (1.80)	7.58 (0.68)	7.22 (10.68)
9	4.99 (2.48)	4.99 (0.69)	4.94 (14.24)
11	4.74 (3.29)	4.78 (0.72)	4.58 (16.03)

since the expression types are only persons, organizations, locations and miscellaneous names. We use only a small subset of CoNLL-2002 since our aim here is simply to compare  $Z$ -score with previous methods and not to compete with large scale NER systems. We performed experiments with 5-fold crossvalidation and two different sets of binary features:  $\mathcal{S}_1$  (HMM features) and  $\mathcal{S}_2$  ( $\mathcal{S}_1$  and HMM features for the previous and the next word). We compared the performance of our approach with CRFs and with the HMP and the MM method with linear kernel. As for artificial datasets, we also report results for our method and MM with Hamming loss. For MM algorithms the soft margin parameter  $C$  has been set to 1, while the required accuracy for constraints to be satisfied is given by  $\epsilon = 0.01$ . The number of iterations of the HMP is  $T = 200$ .

The results shown in Table 6 demonstrate the competitiveness of the proposed method. Here the test error is reported. Optimizing the  $Z$ -score, we obtain performance very close to MM-methods. Admittedly, since the length of feature vectors is large, our approach results generally slower than the other methods. However experiments have been performed with a naive implementation of our algorithm based on a conjugate gradient method for inverting  $\mathbf{C}^*$ . Perhaps more sophisticated iterative methods exploiting the sparseness of  $\mathbf{C}^*$  and the use of approximate matrices can speed up the computation. For example the average running times with features  $\mathcal{S}_1$  is about 9465.34 sec for  $Z$ -score, while the MM approach (SVM-struct implementation [8],  $\epsilon = 0.01$ ) takes 1043.16 sec. However computing  $\mathbf{b}^*$  and  $\mathbf{C}^*$  with sampling (150 paths) the time required by  $Z$ -score optimization decreases to 607.45 sec.

**Table 2.** Classification error on test set on NER.

	Z-SCORE	Z-SCORE (H)	MM	MM (H)	HMP	CRFs
$\mathcal{S}_1$	11.66	11.07	13.94	10.97	20.99	12.01
$\mathcal{S}_2$	8.01	7.89	9.04	8.11	13.78	8.29

## 7 Conclusions

In this paper a new discriminative algorithm for sequence labeling has been proposed. The algorithm is fast and easy to implement. It relies on DP to compute the  $Z$ -score as a function of the parameters, and a simple linear system to maximize it. Similar to recent discriminative methods, the learning problem is a convex optimization with an objective function that takes into account arbitrary dependencies between input and output labels and penalizes incorrect decoded sequences based on the Hamming distance from the given output. Our approach avoids the need to explicitly consider the exponential number of constraints that arise in this kind of problems and, unlike previous works, naturally and adequately deals with the infeasible case where there exists no parameter setting for which the correct given pairs are optimal. Moreover the proposed algorithm does not rely on any parameter that needs to be tuned with time-consuming procedures as cross-validation. We are currently developing a kernelized version of our algorithm which will enable us to circumvent the computational problems when the size of the features vectors becomes large. A further investigation includes the analysis of approximate algorithms to obtain mean and covariance matrices in order to reduce the computational cost.

## Acknowledgments

This work was partially supported by NIH grant R33HG003070-01, and the EU Project SMART.

## References

1. Altun Y., Hofmann T., Johnson M. Discriminative learning for label sequences via boosting. In *Advances in Neural Information Processing Systems (NIPS)*, 2003.
2. Altun Y., Tsochantaridis I., Hofmann T. Hidden markov support vector machines. In *20th International Conference on Machine Learning (ICML)*, 2003.
3. Collins M. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2002.
4. Collins M. Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. In *IWPT*, 2001.
5. Lafferty J., Pereira F., McCallum A. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*, 2001.
6. McCallum A., Freitag D., Pereira F. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2000.
7. Taskar B., Guestrin C., Koller D. Max-margin markov networks. In *Neural Information Processing Systems (NIPS)*, 2003.
8. Tsochantaridis I., Hofmann T., Joachims T., Altun Y. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.