# Large margin methods for structured classification: Exponentiated gradient algorithms and PAC-Bayesian generalization bounds

Peter L. Bartlett[1], Michael Collins[2], David McAllester[3], and Ben Taskar[4]

[1] Division of Computer Science and Department of Statistics, U.C.Berkeley
367 Evans Hall #3860, Berkeley, CA 94720-3860
`bartlett@stat.berkeley.edu`

[2] MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139
`mcollins@ai.mit.edu`

[3] Toyota Technological Institute at Chicago
University Press Building 1427 East 60th Street, Second Floor Chicago, Illinois 60637
`mcallester@tti-c.org`

[4] Computer Science Department, Stanford University
Gates Building 1A, Stanford, CA 94305
`btaskar@cs.stanford.edu`

**Abstract.** We consider the problem of *structured classification*, where the task is to predict a label $y$ from an input $x$, and $y$ has meaningful internal structure. Our framework includes supervised training of both Markov random fields and weighted context-free grammars as special cases. We describe an algorithm that solves the large-margin optimization problem defined in [12], using an exponential-family (Gibbs distribution) representation of structured objects. The algorithm is efficient – even in cases where the number of labels $y$ is exponential in size – provided that certain expectations under Gibbs distributions can be calculated efficiently. The optimization method we use for structured labels relies on a more general result, specifically the application of exponentiated gradient (EG) updates [4, 5] to quadratic programs (QPs). We describe a new method for solving QPs based on these techniques, and give bounds on its rate of convergence. In addition to their application to the structured-labels task, the EG updates lead to simple algorithms for optimizing "conventional" binary or multiclass SVM problems. Finally, we give a new generalization bound for structured classification, using PAC-Bayesian methods for the analysis of large margin classifiers.

## 1 Introduction

Structured classification is the problem of predicting $y$ from $x$ in the case where $y$ has meaningful internal structure. For example $x$ might be a word string and $y$ a sequence of part of speech labels, or $x$ might be a Markov random field and $y$ a labeling of $x$, or $x$ might be a word string and $y$ a parse of $x$. In these examples the number of possible labels $y$ is exponential in the size of $x$. This paper presents a training algorithm and a generalization bound for a general definition of structured classification covering both Markov random fields and parsing.

We restrict our attention to linear discriminative classification. We assume that pairs $\langle x, y\rangle$ can be embedded in a linear feature space $\Phi(x, y)$, and that a predictive rule is determined by a direction (weight vector) $\mathbf{w}$ in that feature space. In linear discriminative prediction we select the $y$ that has the greatest value for the inner product $\langle \Phi(x, y), \mathbf{w} \rangle$. Linear discrimination has been widely studied in the binary and multiclass setting [3, 2]. However, the case of structured labels has only recently been considered [12, 1]. The structured-label case takes into account the internal structure of $y$ in the assignment of feature vectors, the computation of loss, and the definition and use of margins.

We focus on a formulation where each label $y$ is represented as a set of "parts", or equivalently, as a bit-vector. Moreover, we assume that the feature vector for $y$ and the loss for $y$ are both linear in the individual bits of $y$. This formulation has the advantage that it naturally covers both simple labeling problems, such as part-of-speech tagging, as well as more complex problems such as parsing.

This paper contains two results. The first concerns the large-margin optimization problem defined in [12] for selecting the classification direction $\mathbf{w}$ given a training sample. The starting-point for these methods is a primal problem that has one constraint for each possible labeling $y$; or equivalently a dual problem where each $y$ has an associated dual variable. A method is given in [12] for dealing with an exponential number of labelings, in the case where the label set has a Markov random field structure. We give a new approach to the problem that relies on an exponential-family (Gibbs distribution) representation of structured objects. Generally speaking, the algorithm is efficient – even in cases where the number of labels $y$ is exponential in size – provided that certain expectations under Gibbs distributions can be calculated efficiently. The computation of these expectations appears to be a natural computational problem for structured problems, and has specific polynomial-time dynamic programming algorithms for some important examples: specifically, the clique-tree belief propagation algorithm can be used in Markov random fields, and the inside-outside algorithm can be used in the case of weighted context-free grammars.

The optimization method we use for structured labels relies on a more general result, specifically the application of exponentiated gradient (EG) updates [4, 5] to quadratic programs (QPs). We describe a new method for solving QPs based on these techniques, and give bounds on its rate of convergence. The method leads to an optimization method which uses multiplicative updates on dual parameters in the problem.[5] In addition to their application to the structured-labels task, the EG updates lead to simple algorithms for optimizing "conventional" binary or multiclass SVM problems.

Our second result is a PAC-Bayesian margin bound for generalization loss in structured classification. This PAC-Bayesian bound improves on the bound in [12] in a variety of ways. Like PAC-Bayesian margin bounds for binary linear classification [7, 8], our bound for the collective case has a simple proof and is particularly tight in the case of low but nonzero empirical error. More interestingly, the new bound has a different formal structure. Let $H(y,\ y')$ be the Hamming distance between a labeling $y$ and a labeling $y'$. Consider a training pair $\langle x_i, y_i \rangle$ and two other possible values $y$ and $y'$. The new bound improves $H(y, y_i) + H(y_i, y')$ to $H(y,\ y')$. It is not yet clear how difficult it is to establish this triangle inequality improvement with other proof methods.

---

[5] Note that our updates are different from the multiplicative updates for support vector machines described in [10]. In particular, the updates in [10] do not factor in a way that allows algorithms for MRFs and WCFGs based on Gibbs-distribution representations, as described in this paper.

An open problem involves the significance of the new generalization bound. The new generalization bound, although strictly tighter than the bound in [12], is even further removed from the max-margin optimization problem formulated here or in [12]. It remains an open problem as to whether a direct optimization of this new bound, or some simplification of it, is feasible and whether such direct optimization would improve generalization performance.

## 2 The General Setting

We consider the problem of learning a function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ is some countable or uncountable set, and $\mathcal{Y}$ is a countable set. We assume a loss function $L : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$. The function $L(x, y, \hat{y})$ measures the loss when $y$ is the true label for $x$, and $\hat{y}$ is another label, typically the label proposed for $x$ by some function. In general we will assume that $L(x, y, \hat{y}) = 0$ for $y = \hat{y}$. Given some distribution $D(x, y)$ over examples in $\mathcal{X} \times \mathcal{Y}$, our aim is to find a function with low expected loss, or risk:

$$\mathbf{E}_{(x,y)\sim D} L(x, y, f(x)).$$

We consider functions $f$ which take a linear form. First, we assume a fixed function $\mathbf{G}$ which maps an input $x$ to a set of candidates $\mathbf{G}(x)$. For all $x$, we assume that $\mathbf{G}(x) \subseteq \mathcal{Y}$, and that $\mathbf{G}(x)$ is finite. A second component to the model is a feature-vector representation $\Phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ of dimension $d$. Given these definitions, the functions that we consider take the following form, [6] for some $\mathbf{w} \in \mathbb{R}^d$,

$$f_{\mathbf{w}}(x) = \arg \max_{y \in \mathbf{G}(x)} \langle \Phi(x, y), \mathbf{w} \rangle$$

Given training examples $(x_i, y_i)$ for $i = 1 \ldots n$, we will formalize a large-margin optimization problem that is a generalization of support vector machine methods for binary classifiers, and is essentially the same as the formulation in [12]. The optimal parameters are taken to minimize the following regularized empirical risk function:

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \left( \max_y \left( L(x_i, y_i, y) - m_{i,y}(\mathbf{w}) \right) \right)_+$$

where $m_{i,y}(\mathbf{w}) = \langle \mathbf{w}, \phi(x_i, y_i) \rangle - \langle \mathbf{w}, \phi(x_i, y) \rangle$ is the "margin" on example $y$, and $(x)_+ = \max\{x, 0\}$. Note that this optimization problem aims to separate each $y \in \mathbf{G}(x_i)$ from the target label $y_i$ by a margin that is proportional to the loss $L(x_i, y_i, y)$.

This optimization can be expressed as the primal problem in Figure 1. Following [12], the dual of this problem is also shown in Figure 1. The dual involves variables $\alpha_{i,y}$ for all $i = 1 \ldots n$, $y \in \mathbf{G}(x_i)$. The dual objective is a quadratic program in these variables. Note that the dual variables for each example are constrained to form a probability distribution.

### 2.1 Models for structured classification

The problems we are interested in concern structured labels, which have a natural decomposition into "parts". Examples of methods which lead to structured labels are

[6] Note that in the case that two members $y_1$ and $y_2$ have the same tied value for $\langle \Phi(x, y), \mathbf{w} \rangle$, we assume that there is some fixed, deterministic way for breaking ties. For example, one approach would be to assume some default ordering on the members of $\mathcal{Y}$.

**Primal problem:**

$\min_{\mathbf{w},\bar{\epsilon}} \left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \epsilon_i\right)$

Subject to the constraints:

$$\forall i, \forall y \in \mathbf{G}(x_i), \quad \langle \mathbf{w}, \Phi_{i,y}\rangle \geq L_{i,y} - \epsilon_i$$
$$\forall i, \quad \epsilon_i \geq 0$$

**Dual problem:**

$\max_{\bar{\alpha}} \left(C\sum_{i,y} \alpha_{i,y} L_{i,y} - \frac{1}{2}C^2 \sum_{i,y}\sum_{j,z} \alpha_{i,y}\alpha_{j,z}\langle \Phi_{i,y}, \Phi_{j,z}\rangle\right)$

Subject to the constraints:

$$\forall i, \sum_y \alpha_{i,y} = 1$$
$$\forall i, y, \ \alpha_{i,y} \geq 0$$

Relationship between optimal values for primal and dual problems: $\mathbf{w}^* = C\sum_{i,y} \alpha^*_{i,y}\Phi_{i,y}$ where $\mathbf{w}^*$ is the $\arg\min$ of the primal problem, and $\bar{\alpha}^*$ is the $\arg\max$ of the dual problem.

**Fig. 1.** The primal and dual problems. We use the definitions $L_{i,y} = L(x_i, y_i, y)$, and $\Phi_{i,y} = \Phi(x_i, y_i) - \Phi(x_i, y)$. Note that we also assume that for all $i$, $L_{i,y} = 0$ for $y = y_i$. The constant $C$ dictates the relative penalty for values of the slack variables $\epsilon_i$ which are greater than 0.

Markov random fields (MRFs), and weighted context-free grammars (WCFGs) (we elaborate more on these examples later in this section). Formally, we assume some countable set of parts, $\mathcal{R}$. We also assume a function $R$ which maps each object $(x, y) \in \mathcal{X} \times \mathcal{Y}$ to a finite subset of $\mathcal{R}$. Thus $R(x, y)$ is the set of parts belonging to a particular object. In addition we assume a feature-vector representation $\phi$ of parts: this is a function $\phi : \mathcal{X} \times \mathcal{R} \to \mathbb{R}^d$. The feature vector for an object $(x, y)$ is then a sum of the feature vectors for its parts:

$$\Phi(x, y) = \sum_{r \in R(x,y)} \phi(x, r)$$

In addition, we assume that the loss function $L(x, y, \hat{y})$ decomposes into a sum over parts, as follows:

$$L(x, y, \hat{y}) = \sum_{r \in R(x,\hat{y})} l(x, y, r)$$

Finally, for convenience we will also define indicator variables $I(x, y, r)$ which are 1 if $r \in R(x, y)$, 0 otherwise. We also define sets $R(x_i) = \cup_{y \in \mathbf{G}(x_i)} R(x_i, y)$ for the training examples $i = 1 \ldots n$. Thus $R(x_i)$ is the set of parts that is seen in at least one of the objects $\{(x_i, y) \ : \ y \in \mathbf{G}(x_i)\}$.

**Example 1: Markov Random Fields.** In this example we assume that the space of labels $\mathbf{G}(x)$, and their underlying structure, can be represented by a graph. The graph $G = (V, E)$ is a collection of vertices $V = \{v_1, v_2, \ldots v_l\}$ and edges $E$. Each vertex $v_i \in V$ has a set of possible labels, $\mathcal{Y}_i$. The set $\mathbf{G}(x)$ is then defined as $\mathcal{Y}_1 \times \mathcal{Y}_2 \ldots \times \mathcal{Y}_l$. Each possible labeling $y$ for the entire graph can be written as $y = \{y_1, y_2, \ldots y_l\}$.

We give a definition of the decomposition of each $y$ into a set of parts through the cliques in the graph. Each clique in the graph has a set of possible *configurations*: for example, if a particular clique contains vertices $\{v_3, v_5, v_6\}$, the set of possible configurations of this clique is $\mathcal{Y}_3 \times \mathcal{Y}_5 \times \mathcal{Y}_6$. We define $\mathcal{C}$ to be the set of cliques in the graph, and for any $c \in \mathcal{C}$ we define $\mathcal{Y}(c)$ to be the set of possible configurations for that clique. Finally, we define $\mathcal{R} = \{(c, a) \mid c \in \mathcal{C}, a \in \mathcal{Y}(c)\}$. Thus the number of possible parts in this decomposition is $|\mathcal{R}| = \sum_{c \in \mathcal{C}} |\mathcal{Y}(c)|$. In this case we define $R(x, y) = \{(c, a) \in \mathcal{R} \;:\; (c, a)$ is consistent with y$\}$. Thus $R(x, y)$ essentially tracks the assignment of values to each clique in the graph over $y$. Note that for any $y$, we have $|R(x, y)| = |\mathcal{C}|$, as only one configuration of each clique in $\mathcal{C}$ can be consistent with $y$.

All that remains is to define a feature vector representation and a loss function. The feature vector representation $\phi(x, c, a)$ for each part can essentially track any characteristics of the assignment $a$ for clique $c$, together with any features of the entire input $x$. Recall that the overall loss for a label $\hat{y}$ when compared to a true label $y$ is defined as $\sum_{(c,a) \in R(x,\hat{y})} l(x, y, (c, a))$, where $l(x, y, (c, a))$ is the loss for one clique assignment. As one example of a loss that can be expressed in this way, consider the Hamming loss used in [12], defined as follows: if $y = \{y_1, \ldots y_l\}$, and $\hat{y} = \{\hat{y}_1, \ldots, \hat{y}_l\}$, then $L(x, y, \hat{y}) = \sum_{i=1\ldots l} I_{y_i \neq \hat{y}_i}$. To achieve this: First, assign each vertex $v_i$ to a single one of the cliques in which it appears. Second, define $l(x, y, (c, a))$ to be the number of labels in the assignment $(c, a)$ which are both incorrect, and also correspond to vertices which have been assigned to the clique $c$. This definition leads to Hamming loss: note that assigning each vertex to a single clique avoids "double counting" of label errors.

**Example 2: Weighted Context-Free Grammars (WCFGs).** Our second example considers the case where $x$ is an input string, and $y$ is a "parse tree" for that string. More formally, we take $y$ to be a left-most derivation for $x$ under some context-free grammar. The set $\mathbf{G}(x)$ is the set of all left-most derivations for $x$ under the grammar. In general, ambiguity will lead to a given $x$ having many different possible derivations; our task is to learn a strategy for choosing between the members of $\mathbf{G}(x)$.

For convenience, we restrict the grammar to be in Chomsky-normal form, where all rules in the grammar are of the form $\langle A \to B\ C\rangle$ or $\langle A \to a\rangle$, where $A, B, C$ are non-terminal symbols, and $a$ is some terminal symbol. We take a part to be a CF-rule-tuple $\langle A \to B\ C, s, m, e, x\rangle$. The tuple specifies a particular rule $A \to B\ C$, and its position within the sentence $x$. Under this representation $A$ spans words $s \ldots e$ inclusive in $x$; $B$ spans words $s \ldots m$ inclusive; and $C$ spans words $(m+1) \ldots e$ inclusive. The set $\mathcal{R}$ is the set of all possible tuples of this form. The function $R(x, y)$ maps a derivation $y$ to the set of parts which it includes. Note that because all rules are in binary-branching form, $|R(x, y)|$ is constant across different derivations $y$ for the same input sentence $x$.

In WCFGs the function $\phi(x, r)$ can be any function mapping a rule production and its position in the sentence $x$, to some feature vector representation. For example, $\phi$ could include features which identify the rule used in the production, or features which track the rule identity together with features of the words at positions $s, m, e$ and neighboring positions in the string $x$. Now consider the loss function. One approach would be to define $l(x, y, r)$ to be 0 if $r$ is in the derivation $y$ and 1 otherwise. This definition would lead to $L(x, y, \hat{y})$ being the number of CF-rule-tuples in $\hat{y}$ which are not seen in $y$. Another, less strict, definition would be to define $l(x, y, r)$ to be 1 only if the non-terminal $A$ is not seen spanning words $s \ldots e$ in the derivation $y$. This would lead to $L(x, y, \hat{y})$ tracking the number of "constituent errors" in $\hat{y}$, where a constituent is a (non-terminal, start-point, end-point) tuple such as $(A, s, e)$.

### 2.2 A new dual in terms of marginals

We now begin to consider how to solve the optimization problem in Figure 1 when applied to the problem of labels with parts. As shown in [12], the dual in Figure 1 can be reframed in terms of "marginal" terms. We will also find it useful to consider this alternative formulation of the dual. We define the marginals $\mu_{i,r}(\bar{\alpha})$ for all $i, r$, given dual variables $\bar{\alpha}$, as follows:

$$\mu_{i,r}(\bar{\alpha}) = \sum_{y} \alpha_{i,y} I(x_i, y, r). \tag{1}$$

Now consider the dual objective in Figure 1, for convenience repeated here:

$$Q(\bar{\alpha}) = C \sum_{i,y} \alpha_{i,y} L_{i,y} - \frac{1}{2} C^2 \sum_{i,y} \sum_{j,z} \alpha_{i,y} \alpha_{j,z} \langle \Phi_{i,y}, \Phi_{j,z} \rangle \tag{2}$$

It can be shown that $Q(\bar{\alpha})$ is equal to $Q_m(\bar{\mu}(\bar{\alpha}))$, where $\bar{\mu}(\bar{\alpha})$ is the vector with components $\mu_{i,r}(\bar{\alpha})$, and $Q_m(\bar{\mu})$ is defined as follows:

$$\begin{aligned} Q_m(\bar{\mu}) = {} & C \sum_{i} \sum_{r \in R(x_i)} \mu_{i,r} l_{i,r} \\ & - \frac{1}{2} C^2 \sum_{i,j} \sum_{r \in R(x_i), s \in R(x_j)} \left[ I(x_i, y_i, r) - \mu_{i,r} \right] \left[ I(x_j, y_j, s) - \mu_{j,s} \right] \langle \phi_{i,r}, \phi_{j,s} \rangle \, , \end{aligned} \tag{3}$$

where $l_{i,r} = l(x_i, y_i, r)$, $\phi_{i,r} = \phi(x_i, r)$ and $\phi_{j,s} = \phi(x_j, s)$.

To see this, it is sufficient to use the definition in Eq. 1, and also to make use of the following equalities, which can be substituted into the definition for $Q(\bar{\alpha})$: first, $\Phi_{i,y} = \Phi(x_i, y_i) - \Phi(x_i, y) = \sum_{r \in R(x_i, y_i)} \phi_{i,r} - \sum_{r \in R(x_i, y)} \phi_{i,r}$; and second, $L_{i,y} = L(x_i, y_i, y) = \sum_{r \in R(x_i, y)} l(x_i, y_i, r)$.

Now let $\Delta_m$ be the space of marginal vectors which are feasible: $\Delta_m = \{\bar{\mu} : \exists \bar{\alpha} \in \Delta \text{ such that } \bar{\mu} = \bar{\mu}(\bar{\alpha})\}$. Our original optimization problem can be reframed as $\max_{\bar{\mu} \in \Delta_m} Q_m(\bar{\mu})$. Note that $Q_m(\bar{\mu})$ is again a quadratic loss function. In some cases, such as MRFs with reasonable tree width, or WCFGs, it turns out that $\bar{\mu}$ is of polynomial size, and that the domain $\Delta_m$ can be formulated with a polynomial number of linear constraints. See [12] for discussion of the MRF case, for example.

## 3 Exponentiated gradient (EG) updates for large margin problems

### 3.1 General form of the updates

We now turn to a general algorithm for optimizing quadratic programs (QPs), which relies on Exponentiated Gradient (EG) updates. We assume a positive semi-definite matrix $\mathbf{A}$ of dimension $m$, and a vector $\mathbf{b} \in \mathbb{R}^m$, specifying a loss function $Q(\bar{\alpha}) = \mathbf{b}'\bar{\alpha} + \frac{1}{2}\bar{\alpha}'\mathbf{A}\bar{\alpha}$. Here $\bar{\alpha}$ is an $m$-dimensional vector of reals. We assume that $\bar{\alpha}$ is formed by the concatenation of $n$ vectors $\bar{\alpha}_i \in \mathbb{R}^{m_i}$ for $i = 1 \ldots n$, where $\sum_i m_i = m$. Moreover, we assume that each $\bar{\alpha}_i$ is within a simplex of dimension $m_i$, so that the feasible set is

$$\Delta = \{\bar{\alpha} : \bar{\alpha} \in \mathbb{R}^m ; \text{for } i = 1 \ldots n, \sum_{j=1}^{m_i} \alpha_{i,j} = 1; \text{for all } i, j, \, \alpha_{i,j} \geq 0\} \tag{4}$$

Our aim is to find $\arg\min_{\bar{\alpha} \in \Delta} Q(\bar{\alpha})$. Figure 2 gives an algorithm for finding the solution to this minimization problem.

**Inputs:**

- An $(m \times m)$ positive semi-definite matrix $\mathbf{A}$, and an $m$-dimensional vector $\mathbf{b}$, specifying a loss function $Q(\bar{\alpha}) = \mathbf{b} \cdot \bar{\alpha} + \frac{1}{2}\bar{\alpha}'\mathbf{A}\bar{\alpha}$
- Each vector $\bar{\alpha}$ is an $m$-dimensional vector of reals in $\Delta$, where $\Delta$ is defined in Eq. 4.

**Algorithm:**

- Initialize parameter values $\bar{\alpha}^1$ to some point in the interior of $\Delta$ (for example, $\alpha^1_{i,j} = 1/m_i$ for all $i, j$).
- Choose a learning rate $\eta > 0$
- For $t = 1 \dots T$
  - Calculate $\bar{s}^t = \nabla Q(\bar{\alpha}^t) = \mathbf{b} + \mathbf{A}\bar{\alpha}^t$.
  - Calculate $\bar{\alpha}^{t+1}$ from $\bar{\alpha}^t$ using the following updates:

$$\forall i, j, \quad \alpha^{t+1}_{i,j} = \frac{\alpha^t_{i,j} \exp\{-\eta s^t_{i,j}\}}{\sum_k \alpha^t_{i,k} \exp\{-\eta s^t_{i,k}\}}$$

**Output:** Return $\bar{\alpha}^{T+1}$.

**Fig. 2.** The Exponentiated Gradient (EG) algorithm for quadratic programs.

### 3.2 Convergence of the exponentiated gradient QP algorithm

The following theorem shows how the optimization algorithm converges to an optimal solution. The theorem compares the value of the objective function for the algorithm's vector $\bar{\alpha}^t$ to the value for a comparison vector $u \in \Delta$. (For example, consider $u$ as the solution of the QP.) The convergence result is in terms of several properties of the algorithm and the comparison vector $u$. The distance between $u$ and $\bar{\alpha}_1$ is measured using the Kullback-Liebler divergence (KL-divergence). Recall that the KL-divergence between two probability vectors $\bar{u}, \bar{v}$ is defined as $D(\bar{u}, \bar{v}) = \sum_i u_i \log \frac{u_i}{v_i}$. For sequences of probability vectors, $\bar{u} \in \Delta$ with $\bar{u} = (\bar{u}_1, \dots, \bar{u}_n)$ and $\bar{u}_i = (u_{i,1}, \dots, u_{i,m_i})$, we can define a divergence as the sum of KL-divergences: for $\bar{u}, \bar{v} \in \Delta$, $\bar{D}(\bar{u}, \bar{v}) = \sum_{i=1}^n D(\bar{u}_i, \bar{v}_i)$. Two other key parameters are $\lambda$, the largest eigenvalue of the positive semidefinite symmetric matrix $A$, and

$$B = \max_{\bar{\alpha} \in \Delta} \left( \max_i \left(\nabla Q(\bar{\alpha})\right)_i - \min_i \left(\nabla Q(\bar{\alpha})\right)_i \right) \le 2 \left( n \max_{ij} |A_{ij}| + \max_i |b_i| \right).$$

**Theorem 1.** *For all $\bar{u} \in \Delta$,*

$$\frac{1}{T} \sum_{t=1}^{T} Q(\bar{\alpha}^t) \le Q(\bar{u}) + \frac{\bar{D}(\bar{u}, \bar{\alpha}^1)}{\eta T} + \frac{e^{\eta B} - 1 - \eta B}{\eta^2 B^2 \left(1 - \eta(B + \lambda) e^{\eta B}\right)} \frac{Q(\bar{\alpha}^1) - Q(\bar{\alpha}^{T+1})}{T}.$$

*Choosing $\eta = 0.4/(B + \lambda)$ ensures that*

$$Q\left(\bar{\alpha}^{T+1}\right) \le \frac{1}{T} \sum_{t=1}^{T} Q(\bar{\alpha}^t) \le Q(\bar{u}) + 2.5(B + \lambda) \frac{\bar{D}(\bar{u}, \bar{\alpha}^1)}{T} + 1.5 \frac{Q(\bar{\alpha}^1) - Q(\bar{\alpha}^{T+1})}{T}.$$

The first lemma we require is due to Kivinen and Warmuth [5].

**Lemma 1.** *For any $\bar{u} \in \Delta$,*

$$\eta Q(\bar{\alpha}^t) - \eta Q(\bar{u}) \leq \bar{D}(\bar{u}, \bar{\alpha}^t) - \bar{D}(\bar{u}, \bar{\alpha}^{t+1}) + \bar{D}(\bar{\alpha}^t, \bar{\alpha}^{t+1}).$$

We focus on the third term. Define $\nabla_{(i)} Q(\bar{\alpha})$ as the segment of the gradient vector corresponding to the component $\bar{\alpha}_i$ of $\bar{\alpha}$, and define the random variable $X_{i,t}$, satisfying $\Pr\left(X_{i,t} = -\left(\nabla_{(i)} Q(\bar{\alpha}^t)\right)_j\right) = \alpha_{i,j}$.

**Lemma 2.**

$$\bar{D}(\bar{\alpha}^t, \bar{\alpha}^{t+1}) = \sum_{i=1}^{n} \log \mathbf{E}\left[e^{\eta(X_{i,t} - \mathbf{E}X_{i,t})}\right] \leq \left(\frac{e^{\eta B} - 1 - \eta B}{B^2}\right) \sum_{i=1}^{n} \operatorname{var}(X_{i,t}).$$

*Proof.*

$$\begin{aligned}
\bar{D}(\bar{\alpha}^t, \bar{\alpha}^{t+1}) &= \sum_{i=1}^{n} \sum_j \alpha_{ij}^t \log \frac{\alpha_{ij}^t}{\alpha_{ij}^{t+1}} \\
&= \sum_{i=1}^{n} \sum_j \alpha_{ij}^t \left(\log\left(\sum_k \alpha_{ik}^t \exp(-\eta \nabla_{i,k})\right) + \eta \nabla_{i,j}\right) \\
&= \sum_{i=1}^{n} \log\left(\sum_k \alpha_{ik}^t \exp\left(-\eta \nabla_{i,k} + \eta \bar{\alpha}_i^t \cdot \nabla_i\right)\right) \\
&= \sum_{i=1}^{n} \log\left(\mathbf{E}\left[e^{\eta(X_{i,t} - \mathbf{E}X_{i,t})}\right]\right) \\
&\leq \frac{e^{\eta B} - 1 - \eta B}{B^2} \sum_{i=1}^{n} \operatorname{var}(X_{i,t}).
\end{aligned}$$

This last inequality is at the heart of the proof of Bernstein's inequality; e.g., see [9].

The second part of the proof of the theorem involves bounding this variance in terms of the loss. The following lemma relies on the fact that this variance is, to first order, the decrease in the quadratic loss, and that the second order term in the Taylor series expansion of the loss is small compared to the variance, provided the steps are not too large. The lemma and its proof require several definitions. For any $d$, let $\sigma : \mathbb{R}^d \to (0,1)^d$ be the softmax function, $\sigma(\bar{\theta})_i = \exp(\theta_i) / \sum_{j=1}^{d} \exp(\theta_j)$, for $\bar{\theta} \in \mathbb{R}^d$. We shall work in the exponential parameter space: let $\bar{\theta}^t$ be the exponential parameters at step $t$, so that the updates are $\bar{\theta}^{t+1} = \bar{\theta}^t - \eta \nabla Q(\bar{\alpha}^t)$, and the QP variables satisfy $\bar{\alpha}_i^t = \sigma(\bar{\theta}_i^t)$. Define the random variables $X_{i,t,\bar{\theta}}$, satisfying $\Pr\left(X_{i,t,\bar{\theta}} = -\left(\nabla_{(i)} Q(\bar{\alpha}^t)\right)_j\right) = \left(\sigma(\bar{\theta}_i)\right)_j$. This takes the same values as $X_{i,t}$, but its distribution is given by a different exponential parameter ($\bar{\theta}_i$ instead of $\bar{\theta}_i^t$). Define $\left[\bar{\theta}^t, \bar{\theta}^{t+1}\right] = \left\{a\bar{\theta}^t + (1-a)\bar{\theta}^{t+1} : a \in [0,1]\right\}$.

**Lemma 3.** *For some $\bar{\theta} \in [\bar{\theta}^t, \bar{\theta}^{t+1}]$,*

$$\eta \sum_{i=1}^{n} \operatorname{var}(X_{i,t}) - \eta^2 (B + \lambda) \sum_{i=1}^{n} \operatorname{var}(X_{i,t,\bar{\theta}}) \leq Q(\bar{\alpha}^t) - Q(\bar{\alpha}^{t+1}),$$

*but for all $\bar{\theta} \in [\bar{\theta}^t, \bar{\theta}^{t+1}]$, $\operatorname{var}(X_{i,t,\bar{\theta}}) \leq e^{\eta B} \operatorname{var}(X_{i,t})$. Hence,*

$$\sum_{i=1}^{n} \mathrm{var}(X_{i,t}) \leq \frac{1}{\eta\,(1-\eta(B+\lambda)e^{\eta B})}\left(Q(\bar{\alpha}^t) - Q(\bar{\alpha}^{t+1})\right).$$

*Thus, for* $\eta < 0.567/(B+\lambda)$, $Q(\bar{\alpha}^t)$ *is non-increasing in t.*

The proof is in Appendix A. Theorem 4 follows from an easy calculation.

## 4 EG updates for structured objects

We now consider applying the algorithm in Figure 2 to find $\bar{\alpha}^* = \arg\max_{\bar{\alpha}\in\Delta} Q(\bar{\alpha})$, where $Q(\bar{\alpha})$ is the dual form of the maximum margin problem, as defined in Eq. 2 or Figure 1. In particular, we are interested in the optimal values of the primal form parameters, which are related to $\bar{\alpha}^*$ by $\mathbf{w}^* = C\sum_{i,y}\alpha^*_{i,y}\Phi_{i,y}$. A key problem is that in many of our examples, the number of dual variables $\alpha_{i,y}$ precludes dealing with these variables directly. For example, in the MRF case or the WCFG cases, the set $\mathbf{G}(x)$ is exponential in size, and the number of dual variables $\alpha_{i,y}$ is therefore also exponential.

This section shows how the algorithm in Figure 2 can be implemented in a more efficient form, for certain examples of structured objects such as MRFs or WCFGs. Instead of representing the $\alpha_{i,y}$ variables explicitly, we will instead manipulate a vector $\bar{\theta}$ of variables $\theta_{i,r}$ for $i = 1\dots n, r \in R(x_i)$. Thus we have one of these "mini-dual" variables for each part seen in the training data. Each of the variables $\theta_{i,r}$ can take any value in the reals. We now define the dual variables $\alpha_{i,y}$ as a function of the vector $\bar{\theta}$, which takes the form of a Gibbs distribution:

$$\alpha_{i,y}(\bar{\theta}) = \frac{\exp(\sum_{r\in R(x_i,y)}\theta_{i,r})}{\sum_y \exp(\sum_{r\in R(x_i,y)}\theta_{i,r})}.$$

We shall see that the EG algorithm in Figure 2 can be implemented efficiently, independently of the dimensionality of $\bar{\alpha}$, provided that there is an efficient algorithm for computing

$$\mu_{i,r}(\bar{\theta}) = \sum_{i,y}\alpha_{i,y}(\bar{\theta})I(x_i,y,r) \tag{5}$$

for all $i = 1\dots n, r \in R(x_i)$, for any value of $\bar{\theta}$ in $\mathbb{R}^{|\bar{\theta}|}$.

Note that the values of $\mu_{i,r}$ as defined in Eq. 5 can be calculated for MRFs and WCFGs in many cases, using standard algorithms, even in cases where the $\bar{\alpha}$ vectors are exponential in size. For example, in the WCFG case, the inside-outside algorithm can be used to calculate $\mu_{i,r}$, provided that each part $r$ is a context-free rule production, as described in Example 2 of Section 2.1. In the MRF case, the $\mu_{i,r}$ values can be calculated efficiently providing that the tree-width of the underlying graph remains manageable.[7]

Figure 3 gives an algorithm for solving the dual form of the maximum margin problem, which makes use of the Gibbs form $\bar{\alpha}(\bar{\theta})$. The algorithm defines a sequence of values for the $\bar{\theta}$ values, $\bar{\theta}^1, \bar{\theta}^2, \dots, \bar{\theta}^{T+1}$. These values implicitly define a sequence of dual variable values, $\bar{\alpha}^1, \bar{\alpha}^2, \dots \bar{\alpha}^{T+1}$, where $\bar{\alpha}^t = \bar{\alpha}(\bar{\theta}^t)$. It can be verified that the updates to the $\bar{\theta}$ variables take the form

[7] The calculation of marginals as defined in Eq. 5 is also directly related to optimization of Conditional Random Fields (CRFs) [6], in that for these models the gradient of the log-likelihood of the training data can be calculated providing that expectations of this form can be calculated.

**Inputs:** A learning rate $\eta$. A quadratic loss $Q_m(\bar{\mu}(\bar{\alpha}))$ with the structure in Eq. 3.
**Data structures:** A vector $\bar{\theta}$ of parts variables, $\theta_{i,r}$ for $i = 1 \ldots n, r \in R(x_i)$.
**Assumptions:** We define $\alpha_{i,y}(\bar{\theta})$ for $i = 1 \ldots n, y \in \mathbf{G}(x_i)$ to be $\alpha_{i,y}(\bar{\theta}) = \exp(\sum_{r \in R(x_i,y)} \theta_{i,r})/Z_i$ where $Z_i = \sum_y \exp(\sum_{r \in R(x_i,y)} \theta_{i,r})$. We assume we have an efficient procedure for computing $\mu_{i,r} = \sum_y \alpha_{i,y}(\bar{\theta}) I(x_i, y, r)$ for any value of $\bar{\theta}$, for all $i = 1 \ldots n, r \in R(x_i)$.

**Algorithm:**

- Choose initial values $\bar{\theta}^1$ for the $\theta_{i,r}$ variables, thus implicitly defining an initial set of dual variables $\alpha_{i,y}(\bar{\theta}^1)$.
- For $t = 1 \ldots T$
  - For each $i = 1 \ldots n, r \in R(x_i)$, calculate $\mu_{i,r} = \sum_y \alpha_{i,y}(\bar{\theta}^t) I(x_i, y, r)$.
  - For each $i = 1 \ldots n, r \in R(x_i)$, calculate new values for the $\theta_{i,r}$ variables as

$$\theta_{i,r}^{t+1} = \theta_{i,r}^t + \eta C \left( l_{i,r} + C \sum_{j,s \in R(x_j)} \left[ I(x_j, y_j, s) - \mu_{j,s} \right] \langle \phi_{i,r}, \phi_{j,s} \rangle \right)$$

**Output:** Dual variables $\bar{\theta}^{T+1}$, or primal parameters $\mathbf{w}^*$ defined as

$$\mathbf{w}^* = \sum_{i,y} \alpha_{i,y}(\bar{\theta}^{T+1}) \Phi_{i,y} = C \left( \sum_{i,r \in R(x_i,y_i)} \phi_{i,r} - \sum_{i,r \in R(x_i)} \mu_{i,r} \phi_{i,r} \right)$$

where $\mu_{i,r} = \sum_y \alpha_{i,y}(\bar{\theta}^{T+1}) I(x_i, y, r)$.

**Fig. 3.** The Exponentiated Gradient (EG) algorithm for structured classification problems.

$$\theta_{i,r}^{t+1} = \theta_{i,r}^t + \eta \frac{\partial Q_m(\bar{\mu}(\bar{\alpha}(\bar{\theta}^t)))}{\partial \mu_{i,r}}$$

because $\partial Q_m(\bar{\mu})/\partial \mu_{i,r} = C \left( l_{i,r} + C \sum_{j,s \in R(x_j)} \left[ I(x_j, y_j, s) - \mu_{j,s} \right] \langle \phi_{i,r}, \phi_{j,s} \rangle \right)$. The following theorem justifies this approach. (Note that there is a sign change in the $\bar{\theta}$ updates, because we wish to *maximize* the negative semidefinite quadratic forms $Q(\bar{\alpha})$ and $Q_m(\bar{\alpha})$, and the algorithm in Figure 2 is a minimization problem. Thus we implicitly redefine our problem to be minimization of $-Q(\bar{\alpha})$ and $-Q_m(\bar{\alpha})$.)

**Theorem 2.** *Take a QP $Q(\bar{\alpha})$ where $\bar{\alpha} \in \Delta$, for $\Delta$ defined in Eq. 4. Assume that there is some function $I(i,j,k)$, and a function $Q_m(\bar{\mu})$, such that if we define $\mu_k(\bar{\alpha}) = \sum_{i,j} \alpha_{i,j} I(i,j,k)$, then $Q(\bar{\alpha}) = Q_m(\bar{\mu}(\bar{\alpha}))$ for all $\bar{\alpha} \in \Delta$. Assume we apply the algorithm in Figure 2 with initial values $\bar{\alpha}^1$, generating a sequence of values $\bar{\alpha}^1, \ldots, \bar{\alpha}^T$. Define $\alpha_{i,j}(\bar{\theta}) = \exp(\sum_k \theta_k I(i,j,k))/Z_i$, where $Z_i = \sum_j \exp(\sum_k \theta_k I(i,j,k))$. Then the updates*

$$\theta_k^{t+1} = \theta_k^t - \eta \frac{\partial Q_m(\bar{\mu}(\bar{\alpha}(\bar{\theta}^t)))}{\partial \mu_k}$$

*for $t = 1 \ldots (T-1)$ give $\bar{\alpha}(\bar{\theta}^t) = \bar{\alpha}^t$ for $t = 1 \ldots T$.*

*Proof.* See Appendix B.

The main storage requirements of the algorithm in Figure 3 concern the vector $\bar{\theta}$. This is a vector which has as many components as there are parts in the training set. This number can become extremely large. Appendix C gives a "primal form" algorithm which in some cases can be much more efficient in terms of space requirements. The new algorithm requires $O(d + p)$ space, where $d$ is the dimensionality of the primal form parameter vector, and $p$ is the number of parts on a single training example. This algorithm is more efficient in cases where $d$ is much less than the total number of parts in the training set, in particular in cases where kernels are not used.

## 5 Generalization Bound

In this section we use a bit vector formulation of structured classification. More specifically, assume a distribution $D$ over pairs $\langle x, y \rangle$ with $x \in \mathcal{X}$ and $y \in \{0,1\}^{\ell(x)}$. We also assume a known "feasible set" $\mathcal{Y}(x) \subseteq \{0,1\}^{\ell(x)}$ such that with probability 1 we have $y \in \mathcal{Y}(x)$. We also assume a given sample $\langle x_1, y_1 \rangle, \ \ldots, \ \langle y_n, x_n \rangle$ drawn independently from $D$. We will use $y$ to range over the training labels $y_i$ and $\hat{y}$ to range over elements of sets of the form $\mathcal{Y}(x)$ (which includes $y_i$). We write $y_k$ for the $k$th bit in $y$. As in the part formulation we assume a feature vector function $\Phi$ with $\Phi(x, \hat{y}) \in \mathbb{R}^d$ and a loss function $L(x, y, \hat{y}) \in [0, 1]$ satisfying the following linearity conditions.

$$\Phi(x, \hat{y}) = \sum_{k=1}^{\ell(x)} \hat{y}_k \Phi(x, k) \qquad L(x, y, \hat{y}) = \sum_{k=1}^{\ell(x)} \hat{y}_k l(x, y, k)$$

It is sometimes useful to distinguish feature bits from loss bits. A bit position $k$ will be called a feature bit for $x$ if $\Phi(x, k)$ is nonzero. A bit position $k$ will be called a loss bit for $x$ and $y$ if $l(x, y, k)$ is nonzero. The generalization bound given here depends only on the number of feature bits — any number of loss bits can be used. Also, the generalization bound does not require a linear loss function, the bound applies to any loss function $L$ with $L(x, y, \hat{y}) \in [0, 1]$. The generalization bound will involve the following distance measure between two bit vectors $\hat{y}$ and $\hat{y}'$.

$$E(x, \hat{y}, \hat{y}') = \sum_{k: \ \hat{y}_k \neq \hat{y}'_k} ||\Phi(x, k)||$$

This is a weighted Hamming distance on bit strings and satisfies the triangle inequality. For any weight vector $w \in \mathbb{R}^d$ we define the classifier $F_w(x)$ in the obvious way as the bit vector $\hat{y} \in \mathcal{Y}(x)$ maximizing $w \cdot \Phi(x, \hat{y})$. Rather than bound the loss of $F_w$ we will bound the loss of a Gibbs classifier. Let $Q$ be a function taking a weight vector $w$ and a margin $\gamma \geq 0$ and returning a distribution $Q(w, \ \gamma)$ on a second weight vector $w'$. We define the loss of $Q(w, \ \gamma)$, denoted $L(Q(w, \ \gamma), D)$ to be the expectation of $L(x, y, F_{w'}(x))$ when $w'$ is drawn from $Q(w, \gamma)$ and $\langle x, y \rangle$ is drawn from $D$. For $q, p \in [0, 1]$ we define $KL(q||p)$ to be $q(\ln q / \ln p) + (1 - q)(\ln(1 - q)/ln(1 - p)$. For $q \in [0, 1]$ and $\epsilon \geq 0$ we define $KL^{-1}(q, \epsilon)$ to be the supremum of the set of values $p$ satisfying $KL(q||p) \leq \epsilon$. The function $KL^{-1}$ satisfies the following.

$$KL^{-1}(q, \ \epsilon) \leq q + \sqrt{2q\epsilon} + 2\epsilon \tag{6}$$

Let $\ell_{\max}$ be maximum over $i$ from 1 to $n$ of the number of feature bits for $x_i$. Note that $\ell_{\max}$ is a random variable that depends on the sample. Our main generalization result is that there exists a function $Q$ such that with probability at least $1-\delta$ over the choice of the sample we have the following simultaneously for all $w$ and $\gamma$ with $||w||=1$ and $\gamma \in (0,1]$.

$$L(Q(w,\ \gamma),\ D) \\ \leq KL^{-1}\left(\mathcal{L}(w,\ \gamma,\ S)+\frac{1}{n\gamma^2}, \frac{1}{n-1}\left(\frac{\ln\left(2\ell_{\max}n\gamma^2\right)}{\gamma^2}+\ln\left(\frac{n}{\delta}\right)\right)\right) \quad (7)$$

$$\mathcal{L}(w,\ \gamma,\ S)=\frac{1}{n}\sum_{i=1}^{n} \max_{\hat{y}\in\mathcal{C}(x_i,\ w,\ \gamma)} L(x, y_i,\ \hat{y})$$

$$\mathcal{C}(x,\ w,\ \gamma)=\left\{\hat{y}\in\mathcal{Y}(x) : w\cdot\Phi(x,\hat{y}) \geq \max_{\hat{y}'\in\mathcal{Y}(x)}\left(w\cdot\Phi(x,\hat{y}')-\gamma E(x,\hat{y},\hat{y}')\right)\right\} .$$

Taskar et al. prove a generalization bound for Markov random fields. The goal in that setting is to assign labels to the nodes of a Markov random field. We let $z$ range over assignments of labels to nodes. A bit vector representations is discussed below. Recall that $H(z,z')$ is the Hamming distance between two assignments, i.e., the number of nodes on which they differ. Let $N$ be the number of nodes, $V$ the number of values per node, and $q$ be the maximum over all nodes of the degree of that node (the number of edges incident on that node). Let $R_{\max}$ be the maximum of the norm of any feature vector associated with any edge. Under similar conditions to (7), Taskar et al. proved the following for the Hamming loss $L(x_i,z_i,z)=H(z_i,z)/\ell$.

$$L(F_w,\ D) \leq \mathcal{L}'(w,\ \gamma,\ S)+O\left(\sqrt{\frac{1}{n}\left(\frac{\ln(VNn\gamma)}{\gamma^2}+\ln\frac{1}{\delta}\right)}\right). \quad (8)$$

$$\mathcal{L}'(w,\ \gamma,\ S)=\frac{1}{n}\sum_{i=1}^{n} \max_{z\in\mathcal{C}'(i,\ w,\ \gamma)} L(x,z_i,\ z)$$

$$\mathcal{C}'(i,\ w,\ \gamma)=\Big\{z : w\cdot\Phi(x_i,z)\geq \\ \max_{z'}\left(w\cdot\Phi(x_i,z')-2\gamma q R_{\max}(H(z,z_i)+H(z',z_i))\right)\Big\} .$$

To compare this with the bound in [12] we use a bit vector $y(z)$ to represent node assignment $z$ where there is a loss bit to represent each assignment of a value to a node and a feature bit to represent each possible pair of assignments to the two nodes connected by any edge of the Markov random field. Under this representation we have $NV$ loss bits and at most $N^2V^2$ feature bits. The new bound (7) improves on (8) in a variety of ways, some more significant than others. First, (7) explicitly states the constants. Second, (6) implies that $KL^{-1}(q,\epsilon)$ can be arbitrarily smaller than $q+\sqrt{\epsilon}$. Third, and probably most significantly, the set $\mathcal{C}(i,w,\gamma)$ improves on the set $\mathcal{C}'(i,w,\gamma)$. This improvement comes from the fact that $E(y(z),y(z'))$ can be no larger than $2qR_{\max}H(z,z')$ and $H(z,z')$ can be no larger than $H(z,z_i)+H(z_i,z')$.

**Proof of (7).** We give a PAC-Bayesian proof. In applying the PAC-Bayesian theorem we take the "prior" distribution to be a unit-variance isotropic Gaussian on weight vectors defined as $p(w) = e^{-\|w\|^2/2}/Z$. For a given weight vector $w$ and margin $\gamma$ we define the "posterior" $Q(w, \gamma)$ with density

$$q(w' \mid w, \gamma) = \frac{1}{Z} e^{-(1/2)\|w'-\mu\|^2} \text{ with } \mu = \left(\frac{\sqrt{2 \ln\left(2\ell_{\max}\, n\gamma^2\right)}}{\gamma}\right) w.$$

The PAC-Bayesian theorem now implies that with probability at least $1 - \delta$ over the choice of the sample we have that the following holds simultaneously for all $w$ and $\gamma$.

$$L(Q(w,\ \gamma), D) \leq KL^{-1}\left(L(Q(w,\ \gamma), S),\ \ \frac{KL(Q(w,\ \gamma)\|P) + \ln \frac{n}{\delta}}{n-1}\right) \tag{9}$$

For $Q$ and $P$ unit variance Gaussians we have the following.

$$KL\left(Q(w,\ \gamma) \parallel P\right) = \frac{\|\mu\|^2}{2} = \frac{\ln\left(2\ell_{\max} n\gamma^2\right)}{\gamma^2}$$

To derive (7) from (9) it now suffices to show the following.

$$L(Q(w,\ \gamma),\ S) \leq \mathcal{L}(w,\ \gamma) + \frac{1}{n\gamma^2} \tag{10}$$

To prove (10) we first note the following for any vector $\Psi \in \mathbb{R}^d$ with $\|\Psi\| = 1$.

$$\mathrm{P}_{w' \sim Q(w,\ \gamma)}\left[|(w' - \mu) \cdot \Psi| \geq \epsilon\right] \leq 2e^{\frac{-\epsilon^2}{2}} \tag{11}$$

If we replace $\Psi$ by $\Phi(x,k)/\|\Phi(x,k)\|$ and $\epsilon$ by $\|\mu\|\gamma$ in (11) we get the following.

$$\mathrm{P}_{w' \sim Q(w,\ \gamma)}\left[\left|\frac{w'}{\|\mu\|} \cdot \Phi(x,k) - w \cdot \Phi(x,k)\right| \geq \|\Phi(x,k)\|\gamma\right] \leq \frac{1}{\ell_{\max} n\gamma^2}$$

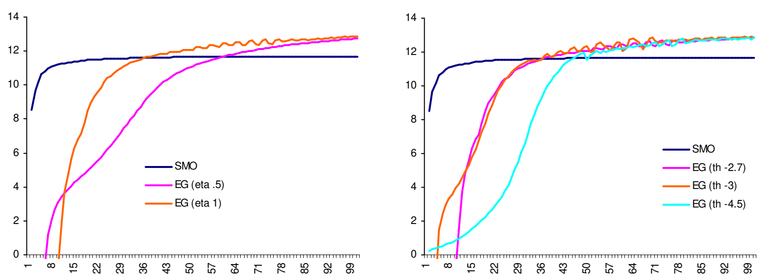
By taking a union bound over the feature bits we get the following.

**Lemma 4.** *For any fixed value of $x$ we have that with probability at least $1 - 1/(n\gamma^2)$ over the selection of $w'$ from $Q(w,\ \gamma)$ the following holds simultaneously for all $k$ in 1 to $\ell(x)$.*

$$\left|\frac{w'}{\|\mu\|} \cdot \Phi(x,k) - w \cdot \Phi(x,k)\right| \leq \|\Phi(x,k)\|\gamma \tag{12}$$

In the case where (12) holds we have the following for any $\hat{y}, \hat{y}' \in \mathcal{Y}(x)$.

$$\begin{aligned}
&\frac{w'}{\|\mu\|} \cdot \Phi(x,\hat{y}) - \frac{w'}{\|\mu\|} \cdot \Phi(x,\hat{y}') \\
&= \sum_{k:\ \hat{y}_k \neq \hat{y}'_k} \hat{y}_k \left(\frac{w'}{\|\mu\|} \cdot \Phi(x,k)\right) - (1 - \hat{y}_k)\left(\frac{w'}{\|\mu\|} \cdot \Phi(x,k)\right) \\
&\leq \sum_{k:\ \hat{y}_k \neq \hat{y}'_k} \hat{y}_k \left(w \cdot \Phi(x,k)\right) - (1 - \hat{y}_k)\left(w \cdot \Phi(x,k)\right) + \gamma\|\Phi(x,k)\| \\
&= w \cdot \Phi(x,\hat{y}) - w \cdot \Phi(x,\hat{y}') + \gamma E(x,\hat{y},\hat{y}')
\end{aligned}$$




(a) (b)

**Fig. 4.** Graph of number of iterations over training set vs. dual objective on a named entity tagging task, for the SMO and EG optimization methods. (a) Comparison of SMO and EG with different $\eta$ parameter; (b) Comparison of SMO and EG with $\eta = 1$ and different initial $\theta$ parameters.

This implies that if $F_{w'}(x) = \hat{y}$ then we have the following.

$$w \cdot \Phi(x, \hat{y}) \geq \max_{\hat{y}' \in \mathcal{Y}(x)} (w \cdot \Phi(x, \hat{y}') - \gamma E(x, \hat{y}, \hat{y}')) \tag{13}$$

But (13) now yields the following for any fixed $x$.

**Lemma 5.** *With probability at least* $1 - 1/(n\gamma^2)$ *over the selection of* $w'$ *from* $Q(w,\ \gamma)$ *we have* $F_{w'}(x) \in \mathcal{C}(x,\ w,\ \gamma)$.

Formula (10) now follows by applying Lemma 5 to each $x_i$ in the sample.

## 6 Experiments

We compared the Exponentiated Gradient algorithm with the factored Sequential Minimal Optimization (SMO) algorithm in [12] on a sequence segmentation task. We selected the first 1000 sentences (12K words) from the CoNLL-2003 named entity recognition challenge data set for our experiment. The goal is to extract (multi-word) entity names of people, organizations, locations and miscellaneous entities. Each word is labelled by 9 possible tags (beginning of one of the four entity types, continuation of one of the types, or not-an-entity). We trained a first-order Markov chain over the tags, where our cliques are just the nodes for the tag of each word and edges between tags of consecutive words. The feature vector for each node assignment consists of the word itself, its capitalization and morphological features, etc., as well as the previous and consecutive words and their features. Likewise, the feature vector for each edge assignment consists of the two words and their features as well as surrounding words.

Figure 4 shows the growth of the dual objective function for each pass through the data for SMO and EG, for several settings of the learning rate parameter $\eta$ and the initial setting of the $\theta$ parameters. Note that SMO starts up very quickly but converges to a suboptimal plateau, while EG lags at the start, but overtakes SMO and achieves a larger than 10% increase in the value of the objective. These preliminary results suggest that a hybrid algorithm could get the benefits of both, by starting out with several SMO updates and then switching to EG. The key issue is to switch from the marginal $\mu$

representation SMO maintains to the Gibbs $\theta$ representation that EG uses. We can find $\theta$ that produces $\mu$ by first computing conditional "probabilities" that correspond to our marginals (e.g. dividing edge marginals by node marginals in this case) and then letting $\theta$'s be the logs of the conditional probabilities.

## References


1. M. Collins. Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. To appear.
2. K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2(5):265–292, 2001.
3. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.
4. J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.
5. J. Kivinen and M. Warmuth. Relative loss bounds for multidimensional regression problems. *Journal of Machine Learning Research*, 45(3):301–329, 2001.
6. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.
7. J. Langford and J. Shawe-Taylor. Pac-Bayes and margins. In *NIPS*, 2002.
8. D. McAllester. Simplified PAC-Bayesian margin bounds. In *COLT*, 2003.
9. D. Pollard. *Convergence of Stochastic Processes*. Springer-Verlag, 1984.
10. F. Sha, L. Saul, and D. Lee. Multiplicative updates for non-negative quadratic programming in support vector machines. Technical report, University of Pennsylvania, 2002.
11. J. Szarski. *Differential Inequalities*. Monografie Matematyczne, 1965.
12. B. Taskar, C. Guestrin, and D. Koller. Max margin Markov networks. In *NIPS*, 2003.


## A Proof of Lemma 3

Consider the mapping from $\bar{\theta}$ parameters to the value $Q(\bar{\alpha})$: $\bar{\theta} \mapsto Q(\bar{\sigma}(\bar{\theta}))$, where $\bar{\sigma}$ is defined by $\bar{\alpha} = \bar{\sigma}(\bar{\theta})$ iff $\bar{\alpha}_i = \sigma(\bar{\theta}_i)$. Now consider the Taylor expansion of the loss $Q$ about $\bar{\theta}^t$. Taylor's theorem implies there is a $\bar{\theta} \in [\bar{\theta}^t, \bar{\theta}^{t+1}]$ for which

$$Q(\bar{\sigma}(\bar{\theta}^{t+1})) = Q(\bar{\sigma}(\bar{\theta}^t)) + \nabla_{\bar{\theta}} Q(\bar{\sigma}(\bar{\theta}^t))'(\bar{\theta}^{t+1} - \bar{\theta}^t) + (\bar{\theta}^{t+1} - \bar{\theta}^t)' \nabla^2_{\bar{\theta}} Q(\bar{\sigma}(\bar{\theta}))(\bar{\theta}^{t+1} - \bar{\theta}^t).$$

It is easy to verify that for $\bar{\alpha} = \bar{\sigma}(\bar{\theta})$ we have $\nabla \bar{\sigma}(\bar{\theta}) = \mathrm{diag}(\bar{\alpha}) - \mathrm{diag}(\bar{\alpha}_1 \bar{\alpha}_1', \ldots, \bar{\alpha}_n \bar{\alpha}_n')$, which is a block diagonal matrix, and the $i$th block is the variance operator for the probability distribution $\bar{\alpha}_i$. Applying the chain rule, we see that the first order term in the Taylor series is

$$\begin{aligned} \nabla_{\bar{\theta}} Q(\bar{\sigma}(\bar{\theta}^t))'(\bar{\theta}^{t+1} - \bar{\theta}^t) &= -\eta \sum_i \left(\nabla_{(i)} Q(\bar{\alpha}^t)\right)' \left(\mathrm{diag}(\bar{\alpha}_i) - \bar{\alpha}_i \bar{\alpha}_i'\right) \left(\nabla_{(i)} Q(\bar{\alpha}^t)\right) \\ &= -\eta \sum_i \mathrm{var}(X_{i,t}). \end{aligned}$$

We next consider second derivatives. Notice that for $i \neq j$, $\nabla_{\bar{\theta}_i} \nabla'_{\bar{\theta}_j} Q = 0$. Thus, the Hessian is block diagonal. We have

$$\begin{aligned}
\frac{\partial}{\partial \theta_{ij} \partial \theta_{ik}} Q(\bar{\sigma}(\bar{\theta})) &= \frac{\partial}{\partial \theta_{ij}} \frac{\partial \sigma(\bar{\theta}_i)'}{\partial \theta_{ik}} \nabla_{(i)} Q(\bar{\alpha}) \\
&= \frac{\partial}{\partial \theta_{ij}} \left( \alpha_{ik} (e_k - \bar{\alpha}_i)' \nabla_{(i)} Q(\bar{\alpha}) \right) \\
&= \left( \alpha_{ij} \delta_{jk} (e_j - \alpha_i) - \alpha_{ij} \alpha_{ik} (e_j + e_k - 2\alpha_i) \right)' \nabla_{(i)} Q(\bar{\alpha}) \\
&\quad + \alpha_{ik} (e_k - \bar{\alpha}_i)' \nabla^2_{(i)} Q(\bar{\alpha}) \alpha_{ij} (e_j - \bar{\alpha}_i),
\end{aligned}$$

where $e_j = (\delta_{j1}, \ldots, \delta_{jm_i})$ and $\delta_{jk}$ is the indicator function for $j = k$. Defining $\bar{\alpha} = \bar{\sigma}(\theta)$, we have

$$\begin{aligned}
\nabla_{\bar{\theta}} Q(\bar{\sigma}(\bar{\theta})) = \operatorname{diag}(\bar{\alpha}) \operatorname{diag} \left( \nabla Q(\bar{\alpha}) - \bar{1} \bar{\alpha}' \nabla Q(\bar{\alpha}) \right) \\
- \nabla \bar{\sigma}(\bar{\theta}) \nabla Q(\bar{\alpha}) \bar{\alpha}' - \bar{\alpha} \nabla Q(\bar{\alpha})' \nabla \bar{\sigma}(\bar{\theta}) + \nabla \bar{\sigma}(\bar{\theta}) \nabla^2 Q(\bar{\alpha}) \nabla \bar{\sigma}(\bar{\theta}),
\end{aligned}$$

where $\bar{1}$ is the all ones vector. If the random variable $Y_i$ has $\Pr\left( Y_i = \left( \nabla_{(i)} Q(\bar{\alpha}) \right)_j \right) = \alpha_{ij}$, we see that

$$\begin{aligned}
&(\bar{\theta}^{t+1} - \bar{\theta}^t)' \nabla^2_{\bar{\theta}} Q(\bar{\sigma}(\bar{\theta})) (\bar{\theta}^{t+1} - \bar{\theta}^t) \\
&= \eta^2 \nabla Q(\bar{\alpha}^t)' \nabla^2_{\bar{\theta}} Q(\bar{\sigma}(\bar{\theta})) \nabla Q(\bar{\alpha}^t) \\
&= \eta^2 \sum_i \mathbf{E} \left[ X^2_{i,t,\bar{\theta}} (Y_i - \mathbf{E} Y_i) - 2 (X_{i,t,\bar{\theta}} Y_i - \mathbf{E} X_{i,t,\bar{\theta}} \mathbf{E} Y_i) \mathbf{E} X_{i,t,\bar{\theta}} \right] \\
&\qquad + \eta^2 \nabla Q(\bar{\alpha}^t)' \nabla \bar{\sigma}(\bar{\theta}) \nabla^2 Q(\bar{\alpha}) \nabla \bar{\sigma}(\bar{\theta}) \nabla Q(\bar{\alpha}^t) \\
&= \eta^2 \sum_i \mathbf{E} \left[ (X_{i,t,\bar{\theta}} - \mathbf{E} X_{i,t,\bar{\theta}})^2 (Y_i - \mathbf{E} Y_i) \right] + \eta^2 \left\| \operatorname{diag}(\sqrt{\lambda}) U \nabla \bar{\sigma}(\bar{\theta}) \nabla Q(\bar{\alpha}^t) \right\|^2 \\
&\leq \eta^2 B^2 \sum_i \operatorname{var}(X_{i,t,\bar{\theta}}) + \eta^2 \lambda \left\| \nabla \bar{\sigma}(\bar{\theta}) \nabla Q(\bar{\alpha}^t) \right\|^2 \\
&\leq \eta^2 (B^2 + \lambda) \sum_i \operatorname{var}(X_{i,t,\bar{\theta}}),
\end{aligned}$$

where we represent the positive semidefinite symmetric Hessian matrix as $\nabla^2 Q(\bar{\alpha}) = A = U' \operatorname{diag}(\lambda) U$ for an orthonormal eigenvector matrix $U$ and vector of eigenvalues $\lambda$. Combining with the first order term gives the first inequality of the lemma. For the second, define $x = \nabla_{(i)} Q(\bar{\alpha}^t)$ and notice that we can write

$$\begin{aligned}
\frac{\partial}{\partial \bar{\theta}_{ik}} \operatorname{var}(X_{i,t,\bar{\theta}}) &= \frac{\partial}{\partial \bar{\theta}_{ik}} x' (\operatorname{diag}(\bar{\alpha}_i) - \bar{\alpha}_i \bar{\alpha}'_i) x \\
&= \sigma_k \left( (x_k - \mathbf{E} X_{i,t,\bar{\theta}})^2 - \operatorname{var}(X_{i,t,\bar{\theta}}) \right),
\end{aligned}$$

Thus, for $v = (\bar{\theta}_i - \bar{\theta}^t_i)$ for some $\bar{\theta} \in [\theta^t, \theta^{t+1}]$, we have

$$v' \nabla_{\bar{\theta}_i} \operatorname{var}(X_{i,t,\theta}) = (v - \bar{1} \min_k v_k)' \nabla_{\bar{\theta}_i} \operatorname{var}(X_{i,t,\theta}) \leq \eta B \operatorname{var}(X_{i,t,\theta}).$$

Viewing this as a differential inequality in one variable, we see that if $\sum_i \mathrm{var}(X_{i,t}) > 0$ then $\sum_i \mathrm{var}(X_{i,t,\theta}) \leq e^{\eta B} \sum_i \mathrm{var}(X_{it})$. (See, for example, Lemma 6.1 in [11].) Lemma 2 shows that if $\sum_i \mathrm{var}(X_{it}) = 0$ then $\bar{\alpha}^t = \bar{\alpha}^{t+1}$, and in that case the lemma is trivially true. So we may assume that $\sum_i \mathrm{var}(X_{it}) > 0$. The lemma follows.

## B Proof of theorem 2

We now give a proof of theorem 2, re-stated here:

**Theorem 3.** *Take a QP $Q(\bar{\alpha})$ where $\bar{\alpha} \in \Delta$ and $\Delta$ is as defined in Eq. 4. Assume that there is some function $I(i,j,k)$, and a function $Q_m(\bar{\mu})$, such that if we define $\mu_k(\bar{\alpha}) = \sum_{i,j} \alpha_{i,j} I(i,j,k)$, then $Q(\bar{\alpha}) = Q_m(\bar{\mu}(\bar{\alpha}))$ for all $\bar{\alpha} \in \Delta$. Assume we apply the algorithm in Figure 2 with initial values $\bar{\alpha}^1$, generating a sequence of values $\bar{\alpha}^1, \ldots, \bar{\alpha}^T$. Define $\alpha_{i,j}(\bar{\theta}) = \exp(\sum_k \theta_k I(i,j,k))/Z_i$, where $Z_i = \sum_j \exp(\sum_k \theta_k I(i,j,k))$. Then the updates*

$$\theta_k^{t+1} = \theta_k^t - \eta \frac{\partial Q_m(\bar{\mu}(\bar{\alpha}(\bar{\theta}^t)))}{\partial \mu_k}$$

*for $t = 1 \ldots (T-1)$ give $\bar{\alpha}(\bar{\theta}^t) = \bar{\alpha}^t$ for $t = 1 \ldots T$.*

*Proof.* The algorithm in Figure 1 starts with some initial dual values $\bar{\alpha}^1$. For $t = 1 \ldots T$, the updated parameters $\bar{\alpha}^{t+1}$ are defined as

$$\forall i,j, \quad \alpha_{i,j}^{t+1} = \frac{\alpha_{i,j}^t \exp\{-\eta \nabla_{i,j}\}}{\sum_j \alpha_{i,j}^t \exp\{-\eta \nabla_{i,j}\}}$$

where $\nabla_{i,j} = \frac{\partial Q(\bar{\alpha})}{\partial \alpha_{i,j}}$. Given that $Q(\bar{\alpha}) = Q_m(\bar{\mu}(\bar{\alpha}))$, and that $\mu_k(\bar{\alpha}) = \sum_{i,j} \alpha_{i,j} I(i,j,k)$, by the chain rule we have

$$\nabla_{i,j} = \frac{\partial Q(\bar{\alpha})}{\partial \alpha_{i,j}} = \frac{\partial Q_m(\bar{\mu}(\bar{\alpha}))}{\partial \alpha_{i,j}} = \sum_k \frac{\partial Q_m(\bar{\mu}(\bar{\alpha}))}{d\mu_k} \frac{\partial \mu_k}{d\alpha_{i,j}} = \sum_k \delta_k I(i,j,k)$$

where $\delta_k = \frac{\partial Q_m(\bar{\mu}(\bar{\alpha}))}{d\mu_k}$.

We now prove that $\bar{\alpha}^t = \bar{\alpha}(\bar{\theta}^t)$ for $t = 1 \ldots T+1$ by induction over $t$. The base case, for $t = 1$, holds by assumption. To prove the inductive case, assume $\bar{\alpha}(\bar{\theta}^t) = \bar{\alpha}^t$, and note that by the definition of the updates, $\theta_k^{t+1} = \theta_k^t - \eta \delta_k$. Then for all $i, j$,

$$\begin{aligned}
\alpha_{i,j}(\bar{\theta}^{t+1}) &= \frac{\exp(\sum_k I(i,j,k)\theta_k^{t+1})}{\sum_j \exp(\sum_k I(i,j,k)\theta_k^{t+1})} = \frac{\exp(\sum_k I(i,j,k)(\theta_k^t - \eta\delta_k)}{\sum_j \exp(\sum_k I(i,j,k)(\theta_k^t - \eta\delta_k)} \\
&= \frac{\alpha_{i,j}(\bar{\theta}^t)\exp(-\eta \sum_k I(i,j,k)\delta_k)}{\sum_j \alpha_{i,j}(\bar{\theta}^t)\exp(-\eta \sum_k I(i,j,k)\delta_k)} = \frac{\alpha_{i,j}(\bar{\theta}^t)\exp(-\eta\nabla_{i,j})}{\sum_j \alpha_{i,j}(\bar{\theta}^t)\exp(-\eta\nabla_{i,j})} \\
&= \frac{{\alpha_{i,j}}^t \exp(-\eta\nabla_{i,j})}{\sum_j {\alpha_{i,j}}^t \exp(-\eta\nabla_{i,j})} = \alpha_{i,j}^{t+1}
\end{aligned}$$

Thus we have improved the inductive case, that $\alpha_{i,j}(\bar{\theta}^{t+1}) = \alpha_{i,j}^{t+1}$.

**Input:** See algorithm in Figure 3.
**Assumptions:** See algorithm in Figure 3.
**Data structures:** A vector $\mathbf{z} \in \mathbb{R}^d$, where $d$ is the dimensionality of the feature vectors $\phi(r)$ over parts.

**Algorithm:**

- Choose initial values $\mathbf{z}^1$.
- For $t = 1 \ldots T$
  - Initialize $\mathbf{z}^{t+1} = \mathbf{z}^t$.
  - For $i = 1 \ldots n$
    * For $r \in R(x_i)$, calculate $\mu_{i,r} = \sum_y \alpha_{i,y}(\bar{\theta}^t) I(x_i, y, r)$ where

$$\theta^t_{i,r} = \eta C(t-1) l_{i,r} + \eta C^2 \langle \mathbf{z}^t, \phi_{i,r} \rangle$$

    * Set $\mathbf{z}^{t+1} = \mathbf{z}^t + \sum_{r \in R(x_i)} \left[ I(x_i, y_i, r) - \mu_{i,r} \right] \phi_{i,r}$

**Output:** Dual variables $\bar{\theta}^{T+1}$, or primal parameters $\mathbf{w}^*$ defined as

$$\mathbf{w}^* = \sum_{i,y} \alpha_{i,y}(\bar{\theta}^{T+1}) \Phi_{i,y} = C \left( \sum_{i,r \in R(x_i,y_i)} \phi_{i,r} - \sum_{i,r \in R(x_i)} \mu_{i,r} \phi_{i,r} \right)$$

where $\mu_{i,r} = \sum_y \alpha_{i,y}(\bar{\theta}^{T+1}) I(x_i, y, r)$.

**Fig. 5.** A "primal form" algorithm which is equivalent to the algorithm in Figure 3

## C A primal form algorithm

The main storage requirements of the algorithm in Figure 3 concern the vector $\bar{\theta}$. This has as many components as there are parts in the training set, which can be extremely large. For example, in the WCFG case, if we have sentences of length $l$, a grammar with $g$ rules, and $n$ examples, then there are $O(ngl^3)$ parts. For example, the Wall Street Journal treebank has roughly $n = 40,000$, $g = 12,000$, and $l = 25$ (on average), which leads to approximately $7.5 \times 10^{12}$ distinct parts.

In Figure 5 we give a modified algorithm with $O(d+G)$ space requirements, where $G$ is the space required to compute the $\mu_{i,r}$ variables of a single example $i$ and $d$ is the dimensionality of the feature vector representation. For example, $G = O(gl^3)$ in the WCFG case, so the overall space requirements are $O(d + gl^3)$, a significant saving on $O(ngl^3)$ for the algorithm in Figure 3.

The algorithm maintains a vector $\mathbf{z}$ at each iteration whose dimension is the same as that of the primal space. In cases where $d + G < |\bar{\theta}|$ this can lead to a considerable saving in terms of storage: it is clear than in cases where kernels are not used (i.e., where $d$ is relatively small), it is likely that this algorithm will be considerably more efficient in terms of space requirements. Note that the algorithms are equivalent because the equivalence $\theta^t_{i,r} = \eta C(t-1) l_{i,r} + \eta C^2 \langle \mathbf{z}^t, \phi_{i,r} \rangle$ holds for $t = 1 \ldots (T+1)$. This can again be proved by induction. The base case is to assume that the equivalence holds for $t = 1$. The inductive case is then relatively straightforward, following from the definitions of $\theta^t_{i,r}$ in terms of $\mathbf{z}^t$, and $\mathbf{z}^{t+1}$ in terms of $\mathbf{z}^t$.

## D An Alternative Proof of Convergence

Definitions:

- $\bar{\alpha}^t$ is the parameter vector at iteration $t$ of the EG algorithm. Recall that we assume that $\bar{\alpha}^t \in \mathbb{R}^m$ is formed by the concatenation of $n$ vectors $\bar{\alpha}_i^t \in \mathbb{R}^{m_i}$ for $i = 1 \dots n$, where $\sum_i m_i = m$. Moreover, we assume that $\bar{\alpha}^t$ is a member of $\Delta$, where $\Delta$ is defined as:

$$\Delta = \{\bar{\alpha} : \bar{\alpha} \in \mathbb{R}^m; \text{for } i = 1 \dots n, \sum_{j=1}^{m_i} \alpha_{i,j} = 1; \text{for all } i, j,\, \alpha_{i,j} \geq 0\} \qquad (14)$$

- $\bar{\theta}^t \in \mathbb{R}^m$ is the "natural" parameter vector corresponding to $\bar{\alpha}^t$. More precisely, we have the relationship

$$\bar{\alpha}_{i,j}^t = \frac{\exp\{\theta_{i,j}^t\}}{\sum_j \exp\{\theta_{i,j}^t\}}$$

- We define $\sigma : \mathbb{R}^n \rightarrow \mathbb{R}^n$ for any $n$ to be a mapping from the natural parameters to the mean parameters, i.e., we have $\bar{\alpha}_i^t = \sigma(\bar{\theta}_i^t)$ for all $t$, $i$. In addition we define $\bar{\sigma} : \mathbb{R}^m \rightarrow \mathbb{R}^m$ to be the mapping such that $\bar{\alpha}^t = \bar{\sigma}(\theta^t)$.
- We assume that the quadratic function that is being minimized is $Q(\bar{\alpha}) = \langle \mathbf{b}, \bar{\alpha} \rangle + \frac{1}{2}\bar{\alpha}'\mathbf{A}\bar{\alpha}$. We also define

$$L(\bar{\alpha}) = Q(\bar{\alpha}) - Q(\mathbf{u})$$

  where $\mathbf{u} = \arg\min_{\bar{\alpha} \in \Delta} Q(\bar{\alpha})$.
- We define $\nabla_{i,j}^t$ to be $\frac{\partial Q(\bar{\alpha}^t)}{\partial \bar{\alpha}_{i,j}}$. Note that we then have

$$\bar{\theta}^{t+1} = \bar{\theta}^t - \eta \nabla^t$$

  where $\eta$ is the learning rate.
- We define

$$D(\bar{\alpha}_i^t, \bar{\alpha}_i^{t+1}) = \sum_j \alpha_{i,j}^t \log \frac{\alpha_{i,j}^t}{\alpha_{i,j}^{t+1}}$$

  and

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) = \sum_i D(\bar{\alpha}_i^t, \bar{\alpha}_i^{t+1}) = \sum_{i,j} \alpha_{i,j}^t \log \frac{\alpha_{i,j}^t}{\alpha_{i,j}^{t+1}}$$

- We define $D_Q(\bar{\alpha}, \bar{\beta})$ to be the Bregman distance associated with $Q$,

$$D_Q(\bar{\alpha}, \bar{\beta}) = Q(\bar{\alpha}) - Q(\bar{\beta}) - \langle \nabla Q(\bar{\beta}), \bar{\alpha} - \bar{\beta} \rangle$$

The first lemma, from [5], is again central to our proof:

**Lemma 6.** *For any $\bar{u} \in \Delta$,*

$$\eta Q(\bar{\alpha}^t) - \eta Q(\bar{u}) \leq \bar{D}(\bar{u}, \bar{\alpha}^t) - \bar{D}(\bar{u}, \bar{\alpha}^{t+1}) + \bar{D}(\bar{\alpha}^t, \bar{\alpha}^{t+1}).$$

The main result, which will lead to the proof of convergence, is the following:

**Lemma 7.** *For all t, under the EG-QP algorithm for the loss function* $Q(\bar{\alpha}) = \langle \mathbf{b}, \bar{\alpha} \rangle + \frac{1}{2}\bar{\alpha}'\mathbf{A}\bar{\alpha}$ *we have*

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \eta^2(\lambda_{MAX} + 0.5B)L(\bar{\alpha}^t)$$

*where* $\lambda_{MAX}$ *is the maximum eigenvalue of* $\mathbf{A}$*, and*

$$B = \max_i \left( \max_j (\nabla Q(\mathbf{u}))_{i,j} - \min_j (\nabla Q(\mathbf{u}))_{i,j} \right) \leq 2 \left( n \max_{ij} |A_{ij}| + \max_i |b_i| \right).$$

The proof of Lemma 7 is the main step in the proof of convergence, and we will turn to this shortly. First, however, we note that Lemmas 6 and 7 together lead directly to the following convergence theorem:

**Theorem 4.** *If we define* $\mathbf{u} = \arg\min_{\bar{\alpha} \in \Delta} Q(\bar{\alpha})$*, then for any* $\eta < 1/(\lambda_{MAX} + 0.5B)$*,*

$$\frac{1}{T}\sum_{t=1}^{T} Q(\bar{\alpha}^t) \leq Q(\mathbf{u}) + \frac{\bar{D}(\mathbf{u}, \bar{\alpha}^1)}{T\eta(1 - \eta(\lambda_{MAX} + 0.5B))}$$

*In addition, choosing* $\eta = 0.5/(\lambda_{MAX} + 0.5B)$ *ensures that*

$$\frac{1}{T}\sum_{t=1}^{T} Q(\bar{\alpha}^t) \leq Q(\mathbf{u}) + \frac{4}{T}(\lambda_{MAX} + 0.5B)D(\mathbf{u}, \bar{\alpha}^1)$$

*Proof.* Combining Lemmas 6 and 7 gives

$$\eta L(\bar{\alpha}^t) \leq \bar{D}(\mathbf{u}, \bar{\alpha}^t) - \bar{D}(\mathbf{u}, \bar{\alpha}^{t+1}) + \eta^2(\lambda_{MAX} + 0.5B)L(\bar{\alpha}^t)$$

Rearranging terms gives

$$L(\bar{\alpha}^t) \leq \frac{\bar{D}(\mathbf{u}, \bar{\alpha}^t) - D(\mathbf{u}, \bar{\alpha}^{t+1})}{\eta(1 - \eta(\lambda_{MAX} + 0.5B))}$$

for any $\eta < 1/(\lambda_{MAX} + 0.5B)$. Summing over $t$ gives

$$\frac{1}{T}\sum_{t=1}^{T} L(\bar{\alpha}^t) \leq \frac{D(\mathbf{u}, \bar{\alpha}^1) - D(\mathbf{u}, \bar{\alpha}^{T+1})}{\eta(1 - \eta(\lambda_{MAX} + 0.5B))}$$

Finally, noting that $L(\bar{\alpha}^t) = Q(\bar{\alpha}^t) - Q(\mathbf{u})$ and $D(\mathbf{u}, \bar{\alpha}^{T+1}) \geq 0$ completes the proof.

We now turn to the proof of Lemma 7. For this we introduce lemmas 8, 9, 10, and 11:

**Lemma 8.** *For any convex loss function* $Q(\bar{\alpha})$*, if* $\mathbf{u} = \arg\min_{\bar{\alpha} \in \Delta} Q(\bar{\alpha})$*, then*

$$L(\bar{\alpha}) = Q(\bar{\alpha}) - Q(\mathbf{u}) = D_Q(\bar{\alpha}, \mathbf{u}) + \langle \bar{\beta}^*, \bar{\alpha} \rangle$$

*where* $D_Q$ *is the Bregman divergence defined by* $Q$*, and* $\bar{\beta}^*$ *has the following properties:*

1. *There are constants* $\lambda_i$ *for* $i = 1 \dots n$ *such that for all* $i, j$, $\bar{\beta}^*_{i,j} = (\nabla Q(\mathbf{u}))_{i,j} - \lambda_i$.
2. *For all* $i, j$, $\bar{\beta}^*_{i,j} \geq 0$.
3. *For all* $i$, *there is some* $j$ *such that* $\bar{\beta}^*_{i,j} = 0$.
4. *From the previous three properties, it follows that for all* $i, j$,

$$\bar{\beta}^*_{i,j} \leq \max_i \left( \max_j (\nabla Q(\mathbf{u}))_{i,j} - \min_j (\nabla Q(\mathbf{u}))_{i,j} \right)$$

**Lemma 9.** *Consider two convex loss functions* $L_1(\bar{\alpha})$ *and* $L_2(\bar{\alpha})$. *The EG updates define a sequence of parameter vectors for either loss function as* $\bar{\alpha}^t = \bar{\sigma}(\bar{\theta}^t)$ *where* $\bar{\theta}^{t+1} = \bar{\theta}^t - \eta\nabla L_1(\sigma(\bar{\theta}^t))$ *or* $\bar{\theta}^{t+1} = \bar{\theta}^t - \eta\nabla L_2(\sigma(\bar{\theta}^t))$ *depending on the loss function in question. Assume the loss functions are such that*

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \gamma_1 \eta^2 L_1(\bar{\alpha}^t) \quad D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \gamma_2 \eta^2 L_2(\bar{\alpha}^t)$$

*for* $L_1$ *and* $L_2$ *respectively. Now consider the loss function* $L(\bar{\alpha}) = L_1(\bar{\alpha}) + L_2(\bar{\alpha})$. *For this loss function, the EG updates are such that*

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq (\gamma_1 + \gamma_2)\eta^2 L(\bar{\alpha}^t)$$

**Lemma 10.** *Consider a loss function* $L(\bar{\alpha}) = \langle \bar{\beta}, \bar{\alpha} \rangle$ *where* $0 \leq \bar{\beta}_{i,j} \leq R$ *for all* $i, j$. *Then under the EG upates for this loss function,*

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \frac{R}{2}\eta^2 L(\bar{\alpha}^t)$$

**Lemma 11.** *Consider a loss function* $L(\bar{\alpha}) = D_Q(\bar{\alpha}, \mathbf{u})$ *where* $D_Q$ *is the Bregman divergence under* $Q$; $Q(\bar{\alpha}) = \langle \mathbf{b}, \bar{\alpha} \rangle + \frac{1}{2}\bar{\alpha}'\mathbf{A}\bar{\alpha}$; *and* $\mathbf{u} = \arg\min_{\bar{\alpha} \in \Delta} Q(\bar{\alpha})$. *Then under the EG upates for this loss function,*

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \lambda_{MAX} \eta^2 L(\bar{\alpha}^t)$$

*where* $\lambda_{MAX}$ *is the largest eigenvalue of* $\mathbf{A}$.

We can now prove lemma 7:

*Proof.* *(of Lemma 7)* By Lemma 8 we can write our loss $L(\bar{\alpha}) = Q(\bar{\alpha}) - Q(\mathbf{u})$ as

$$L(\bar{\alpha}) = L_1(\bar{\alpha}) + L_2(\bar{\alpha})$$

where $L_1(\bar{\alpha}) = D_Q(\bar{\alpha}, \mathbf{u})$ and $L_2(\bar{\alpha}) = \langle \bar{\beta}^*, \bar{\alpha} \rangle$, and $\bar{\beta}^*$ is a vector that satisfies the four properties in Lemma 8. By lemma 11, the EG updates for $L_1$ lead to $D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \eta^2 \lambda_{MAX} L_1(\bar{\alpha}^t)$. By lemma 10, the EG updates for $L_2$ lead to $D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \eta^2 \frac{B}{2} L_2(\bar{\alpha}^t)$. By lemma 9, we can combine these results to show that the EG updates for $L(\bar{\alpha})$ lead to $D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) \leq \eta^2(\lambda_{MAX} + 0.5B)L_2(\bar{\alpha}^t)$, proving the lemma.

We now turn to the proofs of lemmas 8–11:

*Proof.* *(of Lemma 8)* Consider the following Lagrangian, which arises from the minimization of $Q(\bar{\alpha})$ over the set $\Delta$:

$$\mathcal{L}(\bar{\alpha}, \bar{\lambda}, \bar{\beta}) = Q(\bar{\alpha}) - \sum_i \lambda_i \left( \sum_j \bar{\alpha}_{i,j} - 1 \right) - \sum_{i,j} \bar{\beta}_{i,j} \bar{\alpha}_{i,j}$$

The lagrange multipliers $\lambda_i$ enforce the constraints $\sum_j \bar{\alpha}_{i,j} = 1$ for all $i$. The lagrange multipliers $\bar{\beta}_{i,j}$ correspond to the inequality constraints $\bar{\alpha}_{i,j} \geq 0$ for all $i, j$. We then differentiate this w.r.t. $\bar{\alpha}$, giving

$$\frac{\partial \mathcal{L}(\bar{\alpha}, \bar{\lambda}, \bar{\beta})}{\partial \bar{\alpha}} = \nabla Q(\bar{\alpha}) - v(\bar{\lambda}) - \bar{\beta}$$

where $v(\bar{\lambda})$ is a vector such that $v(\bar{\lambda})_{i,j} = \lambda_i$ for all $i, j$. If $\bar{\lambda}^*$ and $\bar{\beta}^*$ are the values for the lagrange multipliers at the saddle point of the lagrangian, then $\frac{\partial \mathcal{L}(\mathbf{u}, \bar{\lambda}^*, \bar{\beta}^*)}{\partial \bar{\alpha}} = 0$, giving

$$\nabla Q(\mathbf{u}) = v(\bar{\lambda}^*) + \bar{\beta}^*$$

This identity shows that $\bar{\beta}^*$ satisfies property 1 in the lemma. By the usual KKT conditions, $\bar{\beta}^*_{i,j} \geq 0$, satisfying property 2. By complementary slackness, $\langle \bar{\beta}^*, \mathbf{u} \rangle = 0$ and therefore $\bar{\beta}^*_{i,j} = 0$ if $\mathbf{u}_{i,j} > 0$. For all $i$, there must be at least one $j$ such that $\mathbf{u}_{i,j} > 0$, and hence for all $i$ there must be some $j$ such that $\bar{\beta}^*_{i,j} = 0$. Property 3 of the lemma therefore also holds for $\bar{\beta}^*$.

Note also that $\langle v(\bar{\lambda}^*), \bar{\alpha} - \mathbf{u} \rangle = 0$ for all $\bar{\alpha} \in \Delta$, as $\langle v(\bar{\lambda}^*), \bar{\alpha} \rangle = \langle v(\bar{\lambda}^*), \mathbf{u} \rangle = \sum_i \lambda_i^*$. Thus for all $\bar{\alpha} \in \Delta$,

$$\langle \nabla Q(\mathbf{u}), \bar{\alpha} - \mathbf{u} \rangle = \langle v(\bar{\lambda}^*) + \bar{\beta}^*, \bar{\alpha} - \mathbf{u} \rangle = \langle \bar{\beta}^*, \bar{\alpha} \rangle$$

Finally, from the identity $D_Q(\bar{\alpha}, \mathbf{u}) = Q(\bar{\alpha}) - Q(\mathbf{u}) - \langle \nabla Q(\mathbf{u}), \bar{\alpha} - \mathbf{u} \rangle$, we have

$$L(\bar{\alpha}) = Q(\bar{\alpha}) - Q(\mathbf{u}) = D_Q(\bar{\alpha}, \mathbf{u}) + \langle \nabla Q(\mathbf{u}), \bar{\alpha} - \mathbf{u} \rangle = D_Q(\bar{\alpha}, \mathbf{u}) + \langle \bar{\beta}^*, \bar{\alpha} \rangle$$

thus proving the lemma.

*Proof.* *(of Lemma 9)* The first step is to note that $D(\bar{\sigma}(\bar{\theta}^1), \bar{\sigma}(\bar{\theta}^2))$ is convex in $\bar{\theta}^2$. Under the EG updates for $L(\bar{\alpha})$, we have $\bar{\theta}^{t+1} = \bar{\theta}^t - \eta \nabla L(\bar{\alpha}^t) = \bar{\theta}^t - \eta \nabla L_1(\bar{\alpha}^t) - \eta \nabla L_2(\bar{\alpha}^t)$. Our goal is to find an upper bound for $D(\bar{\alpha}^t, \bar{\alpha}^{t+1})$. For convenience we define $\nabla_1 = \nabla L_1(\bar{\alpha}^t)$ and $\nabla_2 = \nabla L_2(\bar{\alpha}^t)$; it then follows that

$$
\begin{aligned}
D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) &= D(\bar{\sigma}(\bar{\theta}^t), \bar{\sigma}(\theta^{t+1})) \\
&= D(\bar{\sigma}(\bar{\theta}^t), \bar{\sigma}(\theta^t - \eta\nabla_1 - \eta\nabla_2)) \qquad (15) \\
&\leq \quad \frac{\gamma_1}{\gamma_1+\gamma_2} D(\bar{\sigma}(\bar{\theta}^t), \bar{\sigma}(\bar{\theta}^t - \eta\frac{\gamma_1+\gamma_2}{\gamma_1}\nabla_1)) \\
&\quad + \frac{\gamma_2}{\gamma_1+\gamma_2} D(\bar{\sigma}(\bar{\theta}^t), \bar{\sigma}(\bar{\theta}^t - \eta\frac{\gamma_1+\gamma_2}{\gamma_2}\nabla_2)) \qquad (16) \\
&\leq \quad \frac{\gamma_1}{\gamma_1+\gamma_2} \times \eta^2 \left(\frac{\gamma_1+\gamma_2}{\gamma_1}\right)^2 \gamma_1 L_1(\bar{\alpha}^t) \\
&\quad + \frac{\gamma_2}{\gamma_1+\gamma_2} \times \eta^2 \left(\frac{\gamma_1+\gamma_2}{\gamma_2}\right)^2 \gamma_2 L_2(\bar{\alpha}^t) \qquad (17) \\
&= \eta^2(\gamma_1+\gamma_2)L(\bar{\alpha}^t)
\end{aligned}
$$

Here Eq. 16 follows from Eq. 15 by the convexity of $D(\bar{\sigma}(\bar{\theta}^1), \bar{\sigma}(\bar{\theta}^2))$ in $\bar{\theta}^2$, and Eq. 17 follows from Eq. 16 by the assumptions in the lemma. This completes the proof.

*Proof. (of Lemma 10)* Under the generalized gradient update, we have $\nabla L(\bar{\alpha}) = \bar{\beta}$, and $\bar{\theta}^{t+1} = \bar{\theta}^t - \eta\bar{\beta}$. The EG updates are then as follows, for all $i, j$:

$$\bar{\alpha}_{i,j}^{t+1} = \frac{\bar{\alpha}_{i,j}^t \exp\{-\eta\beta_{i,j}\}}{\sum_j \alpha_{i,j}^t \exp\{-\eta\beta_{i,j}\}}$$

Now, consider a single segment $\bar{\alpha}_i^t$ of the parameter vector. For convenience, define $\mathbf{v} = \alpha_i^t$, $\mathbf{w} = \alpha_i^{t+1}$, and $\mathbf{x} = \bar{\beta}_i$. We will now bound $D(\bar{\alpha}_i^t, \bar{\alpha}_i^{t+1}) = D(\mathbf{v}, \mathbf{w})$ under the EG update. We have

$$
\begin{aligned}
D(\mathbf{v}, \mathbf{w}) &= \sum_j v_j \log\frac{v_j}{w_j} \\
&= \sum_j v_j \log\frac{\sum_k v_k \exp\{-\eta x_k\}}{\exp\{-\eta x_j\}} \\
&= \eta\mathbf{v}\cdot\mathbf{x} + \log\sum_k v_k \exp\{-\eta x_k\} \qquad (18)
\end{aligned}
$$

We will now derive an upper bound for the second term. Take $\gamma = e^{-\eta R}$ where $R$ is a constant such that $x_j \leq R$ for all $x_j$. By a convexity argument, we have

$$\alpha^x \leq 1 - x(1-\alpha)$$

for $\alpha \geq 0$ and $0 \leq x \leq 1$. ([**?**] use this bound in a very similar problem, the derivation of the *Hedge* online algorithm. This bound is also used in a proof on page 27 of [**?**].)

We then have

$$\begin{aligned}\log \sum_k v_k \exp\{-\eta x_k\} &= \log \sum_k v_k \gamma^{x_k/R} \\ &\leq \log \sum_k v_k (1 - \frac{x_k}{R}(1-\gamma)) \\ &= \log \left(1 - \frac{\mathbf{v}\cdot\mathbf{x}}{R}(1-\gamma)\right) \\ &\leq \frac{\mathbf{v}\cdot\mathbf{x}}{R}(\gamma - 1) \\ &= \frac{\mathbf{v}\cdot\mathbf{x}}{R}(\exp\{-\eta R\} - 1) \end{aligned} \tag{19}$$

Combining Eq. 18 and Eq. 19 gives

$$D(\mathbf{v},\mathbf{w}) \leq \mathbf{v}\cdot\mathbf{x}\left(\frac{\exp\{-\eta R\} - 1}{R} + \eta\right) \tag{20}$$

From the bound $e^{-x} \leq 1 - x + x^2/2$ for any $x \geq 0$, we then have $\frac{\exp\{-\eta R\}-1}{R} \leq -\eta + \frac{R\eta^2}{2}$. Applying this bound to Eq. 20 gives

$$D(\mathbf{v},\mathbf{w}) \leq \frac{R\eta^2}{2}\mathbf{v}\cdot\mathbf{x}$$

or equivalently,

$$D(\bar{\alpha}_i^t, \bar{\alpha}_i^{t+1}) \leq \frac{R\eta^2}{2}\langle \bar{\beta}_i^*, \bar{\alpha}_i^t\rangle$$

Summing over $i$ gives

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) = \sum_i D(\bar{\alpha}_i^t, \bar{\alpha}_i^{t+1}) \leq \sum_i \frac{R\eta^2}{2}\langle \bar{\beta}_i^*, \bar{\alpha}_i^t\rangle = \frac{R\eta^2}{2}\langle \bar{\beta}^*, \bar{\alpha}^t\rangle$$

which is the result in the lemma.

*Proof. (of Lemma 11)* The first property which will prove important is as follows: for any $\bar{\theta}$ and $\mathbf{s}$:

$$D(\bar{\sigma}(\bar{\theta}), \bar{\sigma}(\bar{\theta} - \bar{\mathbf{s}})) \leq \frac{1}{2}\|\mathbf{s}\|_2^2 \tag{21}$$

To show this, first consider a single segment $\bar{\theta}_i$, with its corresponding update $\mathbf{s}_i$. See [5] Examples 2 and 5, and proposition 2, for a proof that

$$D(\sigma(\bar{\theta}_i), \sigma(\bar{\theta}_i - \bar{\mathbf{s}}_i)) \leq \frac{1}{2}\|\mathbf{s}_i\|_\infty^2$$

By noting that $\|\mathbf{s}_i\|_\infty^2 \leq \|\mathbf{s}_i\|_2^2$, and by summing over $i$, we prove the result in Eq. 21.

We now apply this to the EG updates, giving the following result:

$$D(\bar{\alpha}^t, \bar{\alpha}^{t+1}) = D(\bar{\sigma}(\bar{\theta}^t), \bar{\sigma}(\bar{\theta}^t - \eta\nabla L(\bar{\alpha}^t)) \leq \frac{1}{2}\eta^2\|\nabla L(\bar{\alpha}^t)\|_2^2 \tag{22}$$

Next, by differentiating $L$ it can be verified that

$$\nabla L(\bar{\alpha}) = \mathbf{A}(\bar{\alpha} - \mathbf{u})$$

Note also that some simple algebra shows that $L(\bar{\alpha}) = \frac{1}{2}(\bar{\alpha} - \mathbf{u})'\mathbf{A}(\bar{\alpha} - \mathbf{u})$ We define $\mathbf{X}$ to be a matrix such that $\mathbf{A} = \mathbf{XX}$ ($\mathbf{X}$ is the "square-root" of $\mathbf{A}$, and must exist because $\mathbf{A}$ is positive semi-definite). We then have for all $\bar{\alpha}$

$$\begin{aligned}
\|\nabla L(\bar{\alpha})\|_2^2 &= (\bar{\alpha} - \mathbf{u})'\mathbf{A}'\mathbf{A}(\bar{\alpha} - \mathbf{u}) \\
&= (\bar{\alpha} - \mathbf{u})'\mathbf{XXXX}(\bar{\alpha} - \mathbf{u}) \\
&= (\bar{\alpha} - \mathbf{u})'\mathbf{XAX}(\bar{\alpha} - \mathbf{u}) \\
&\leq \lambda_{MAX}(\bar{\alpha} - \mathbf{u})'\mathbf{A}(\bar{\alpha} - \mathbf{u}) \\
&= 2\lambda_{MAX}L(\bar{\alpha})
\end{aligned}$$

where $\lambda_{MAX}$ is the largest eigenvalue of $\mathbf{A}$. Combining this result with Eq. 22 proves the lemma.