1. Multi-Sample Updates

In this section, we derive fast approximate optimization algorithms that meet the properties of Theorem 2.1.

1.1. Notation and Review

Recall that $\max_\alpha D_t(\alpha) = \min_w F_T(w)$, where $F_T(w)$ is the structured SVM training error and is defined as

$$F_T(w) = \sum_{t=1}^T f(w; Z_t)$$

(1)

$$f(w; Z_t) = \frac{\lambda}{2}\|w\|^2 + \ell(w, Z_t)$$

(2)

$$\ell(w, Z_t) = \max_Y (\langle w, \Psi(X_t, Y) \rangle + \Delta(Y, Y_t)) - \langle w, \Psi(X_t, Y_t) \rangle$$

(3)

and $D_t(\alpha)$ is the equivalent dual problem and is defined as

$$D_T(\alpha) = -\frac{\lambda T}{2}\|w\|^2 + \sum_{i,Y} \alpha_i^Y \Delta(Y, Y_i)$$

(4)

$$w^T = -\frac{1}{\lambda T} \sum_{i=1}^T u_i$$

(5)

$$u_i = \sum_Y \alpha_i^Y v(X_i, Y)$$

(6)

$$v(X_i, Y) = \Psi(X_i, Y) - \Psi(X_i, Y_i)$$

(7)

Our goal is to derive exact or approximate solvers to the problem

$$\arg \max_{\alpha_t} D_t(\alpha_1...\alpha_{t-1}, \alpha_t)$$

s.t., \( \forall j, \alpha_t^{Y_j} \geq 0, \sum_j \alpha_t^{Y_j} \leq 1 \) \hspace{1cm} (8)

which is equivalent to the following QP problem:

$$\min_{\alpha_t} \frac{1}{2} \alpha_t^T Q \alpha_t + c^T \alpha_t$$

s.t., \( \forall j, \alpha_t^{Y_j} \geq 0, \sum_j \alpha_t^{Y_j} \leq 1 \) \hspace{1cm} (9)

where $Q$ is a $K \times K$ matrix and $c$ is a $K$-vector with elements

$$Q_{ij} = \langle v(X_t, \bar{Y}_i^j), v(X_t, \bar{Y}_t^j) \rangle$$

(10)

$$c_j = -\lambda t \left( \langle w^{t-1}, \bar{V}_t^{Y_j} \rangle + \Delta(\bar{Y}_t^j, Y_t) \right)$$

(11)
1.2. A Fast, Approximate Multi-Sample Update Algorithm

In this section, derive an approximate update step that occurs over a set of samples \( \bar{Y}_t = Y_1^t, Y_2^t, ..., Y_K^t \). For convergence guarantees (see Theorem 2.1), we assume that this set includes the subgradient as the first sample: \( v(X_t, Y_i^t) = \nabla \ell(w^{i-1}; Z_t) \). The update provides an approximate solution to the optimization problem:

\[
\arg\max_{\alpha_t} D_t(\alpha_1 ... \alpha_{t-1}, \alpha_t) \\
\text{s.t., } \forall_j, \alpha_t Y_i^j \geq 0, \sum_j \alpha_t Y_i^j \leq 1
\]

(12)

Pseudo-code for the algorithm is shown in Algorithm 2. In this algorithm, each sample \( \bar{Y}_i^j \) is iterated over once. We define an iterative procedure, such that each \( \alpha_t Y_i^j \) is updated in order \( j = 1...K \). In each iteration, we solve (in closed form) for the optimal value to set \( \alpha_t Y_i^j \) which maximizes \( \Delta D_{t,j}^{expand} \):

\[
\Delta D_{t,j}^{expand} = D_t(\alpha_1 ... \alpha_{t-1}, \alpha_t Y_i^j) - D_t(\alpha_1 ... \alpha_{t-1}, 0)
\]

\[
= -\frac{\lambda t}{2} \left\| w^t - \frac{\alpha_t Y_i^j}{\lambda t} v(X_t, \bar{Y}_t^j) \right\|^2 + \frac{\lambda t}{2} \left\| w^t \right\|^2 + \alpha_t Y_i^j \Delta(\bar{Y}_i^j, Y_t)
\]

\[
= \alpha_t Y_i^j \left( (w^t, j-1, v(X_t, \bar{Y}_i^j)) + \Delta(\bar{Y}_i^j, Y_t) \right) - \frac{(\alpha_t Y_i^j)^2}{2 \lambda t} \left\| v(X_t, \bar{Y}_i^j) \right\|^2
\]

This is maximized (by setting the derivative equal to 0) by choosing

\[
\alpha_t Y_i^j = \frac{\lambda t \left( (w^t, j-1, v(X_t, \bar{Y}_i^j)) + \Delta(\bar{Y}_i^j, Y_t) \right)}{\left\| v(X_t, \bar{Y}_i^j) \right\|^2}
\]

Unfortunately, due to the constraint \( \sum_{j=1}^K \alpha_t Y_i^j \leq 1 \) in Eq. 12, this update becomes invalid once \( \sum_{j=1}^K \alpha_t Y_i^j = 1 \). We therefore consider an alternate “swap” move, which works by setting \( \alpha_t Y_i^j \) while simultaneously scaling all parameters \( \alpha_t Y_i^j ... \alpha_t Y_i^{j-1} \) by \( s = 1 - \alpha_t Y_i^j \). This swap move preserves the constraint \( \sum_{j=1}^K \alpha_t Y_i^j = 1 \). The subsequent change in the dual objective is:

\[
\Delta D_{t,j}^{swap} = D_t(\alpha_1 ... \alpha_{t-1}, s \alpha_t Y_i^j, ..., s \alpha_t Y_i^{j-1}, \alpha_t Y_i^j) - D_t(\alpha_1 ... \alpha_{t-1}, s \alpha_t Y_i^j, ..., s \alpha_t Y_i^{j-1}, 0)
\]

\[
= -\frac{\lambda t}{2} \left\| w^t - \frac{(s-1)u_i^{j-1} + \alpha_t Y_i^j v(X_t, \bar{Y}_i^j)}{\lambda t} \right\|^2 + \frac{\lambda t}{2} \left\| w^t \right\|^2 + (s-1)D_i^{j-1} + \alpha_t Y_i^j \Delta(\bar{Y}_i^j, Y_t)
\]

\[
= -\frac{\lambda t}{2} \left\| w^t - \frac{-\alpha_t Y_i^j u_i^{j-1} + \alpha_t Y_i^j v(X_t, \bar{Y}_i^j)}{\lambda t} \right\|^2 + \frac{\lambda t}{2} \left\| w^t \right\|^2 - \alpha_t Y_i^j D_i^{j-1} + \alpha_t Y_i^j \Delta(\bar{Y}_i^j, Y_t)
\]

\[
= \alpha_t Y_i^j \left[ (w^t, v(X_t, \bar{Y}_i^j)) + \Delta(\bar{Y}_i^j, Y_t) \right] - \left( (w^t, u_i^{j-1}) + D_i^{j-1} \right) + \frac{(\alpha_t Y_i^j)^2}{2 \lambda t} \left\| u_i^{j-1} + v(X_t, \bar{Y}_i^j) \right\|^2
\]

where \( D_i^{j-1} \) is shorthand for

\[
D_i^{j-1} = \sum_{k=1}^{j-1} \alpha_t Y_i^k \Delta(\bar{Y}_i^k, Y_t)
\]

(13)

and can be interpreted as a (weighted) average loss over all samples. The value of \( \alpha_t Y_i^j \) which maximizes \( \Delta D_{t,j}^{swap} \) is:

\[
\alpha_t Y_i^j = \frac{\lambda t \left[ (w^t, v(X_t, \bar{Y}_i^j)) + \Delta(\bar{Y}_i^j, Y_t) \right] - \left( (w^t, u_i^{j-1}) + D_i^{j-1} \right)}{\left\| u_i^{j-1} + v(X_t, \bar{Y}_i^j) \right\|^2}
\]

(14)
Algorithm 1 MultiSampleUpdate

Input: New example $X_t$, $Y_t$, current weights $w^{t-1}$
Output: New weights $w^{t,K}$

1: $\bar{Y}_t^1,...,\bar{Y}_t^K \leftarrow$ IMPORTANCESAMPLE($X_t$, $Y_t$, $w^{t-1}$)
2: Initialize $w^{t,0} \leftarrow \frac{1}{m-1}w^{t-1}$, $u^0_t \leftarrow 0$, $D^0_t \leftarrow 0$, $\alpha_t \leftarrow 0$
3: for $j = 1$ to $K$ do
4:     if $\alpha_t < 1$ then
5:         $\alpha_t^{Y_j} \leftarrow \min \left(1 - \alpha_t, \max \left(0, \frac{\lambda t(l_{j})}{\|v(X_t,Y_t')\|^2}\right)\right)$
6:         $s \leftarrow 1$
7:         $\alpha_t \leftarrow \alpha_t + \alpha_t^{Y_j}$
8:     else
9:         $\alpha_t^{Y_j} \leftarrow \min \left(1, \max \left(0, \frac{\lambda t((w^t.v(X_t,\bar{Y}_t^j)) + \Delta(Y_t^j, Y_t)) - ((w^t.v(u^{t-1}) + D^{t-1}))}{\|u^{t-1} + v(X_t,Y_t')\|^2}\right)\right)$
10:        $s \leftarrow 1 - \alpha_t^{Y_j}$
11:    end if
12: $u^t_j \leftarrow s u^{t-1} + \alpha_t^{Y_j} v(X_t, \bar{Y}_t^j)$
13: $w^{t,j} \leftarrow w^{t,j-1} - \frac{(s-1)u^{t-1} + \alpha_t^{Y_j} v(X_t, \bar{Y}_t^j)}{\lambda t}$
14: $D^t_j \leftarrow sD^{t-1} + \alpha_t^{Y_j} \Delta(\bar{Y}_t^j, Y_t)$
15: end for
16: Optionally repeat steps 3-16 multiple times

We can compute $\alpha_t^{Y_j}$ in $O(d)$ time, where $d$ is the dimensionality of the feature space $\Psi$, if we maintain updated values for $w^t_j$, $u^t_j$, and $D^t_j$. The appropriate updates if we were to scale $\alpha_t$ by $s$ and then set $\alpha_t^{Y_j}$ are:

$$u^t_j \leftarrow s u^{t-1} + \alpha_t^{Y_j} v(X_t, \bar{Y}_t^j)$$

$$w^{t,j} \leftarrow w^{t,j-1} - \frac{(s-1)u^{t-1} + \alpha_t^{Y_j} v(X_t, \bar{Y}_t^j)}{\lambda t}$$

$$D^t_j \leftarrow sD^{t-1} + \alpha_t^{Y_j} \Delta(\bar{Y}_t^j, Y_t)$$

1.3. A Fast, Exact Multi-Sample Update Algorithm For Multiclass Problems

Recall that a cost-sensitive multiclass SVM can be represented by a structured SVM (Eq 1) with a feature space that concatenates features for each class:

$$\Psi(X,Y) = [\psi_1(X,Y) ... \psi_C(X,Y)]$$

$$\psi_c(X,Y) = \begin{cases} \phi(X) & \text{if } Y = c \\ 0 & \text{otherwise} \end{cases}$$

This feature space has the property that

$$(\Psi(X,c_1), \Psi(X,c_2)) = \begin{cases} \phi^2(X) & \text{if } c_1 = c_2 \\ 0 & \text{otherwise} \end{cases}$$

It follows that the matrix $Q$ has entries (Eq 10)

$$Q_{ij} = \begin{cases} 2|\phi(X)|^2 & \text{if } i = j \\ |\phi(X)|^2 & \text{otherwise} \end{cases}$$

It follows that $Q = |\phi(X)|^2 (I_{K \times K} + I_{K \times K})$, where $I_{K \times K}$ is the identity matrix and $1_{K \times K}$ is a matrix of ones. Since Eq 9 is a quadratic program, if we ignore the constraints, a Newton-update

$$\alpha_t \leftarrow -H^{-1} \nabla$$
Algorithm 2 $\textsc{MulticlassMultiSampleUpdate}$

**Input:** New example $X_t, Y_t$, current weights $w^{t-1}$

**Output:** New weights $w^t$

1. For $i = 1 \ldots K$, $c_i \leftarrow -\lambda t \left( \langle w^{t-1}, \nabla(X_t, Y_t^i) \rangle + \Delta(Y_t^i, Y_t) \right)$
2. Let $c_{j(1)}, c_{j(2)}, \ldots, c_{j(K)}$ be the entries of $c$ in ascending order
3. $s \leftarrow 0$, $n = 1$
4. while $n \leq K$ and $c_{j(n)} \leq \min \left( \frac{s}{n+1}, \frac{1+s}{n} \right)$ do
5. $s \leftarrow s + c_{j(n)}$
6. $n \leftarrow n + 1$
7. end while
8. for $i = 1$ to $n$ do
9. $\alpha_{j(i)} \leftarrow - \left( c_{j(i)} - \min \left( \frac{s}{n+1}, \frac{1+s}{n} \right) \right)$
10. $w^t \leftarrow w^t - \frac{\alpha_{j(i)} \nabla(X_t, y_{j(i)})}{\lambda t}$
11. end for

Algorithm 3

1. for $t = 1$ to $T$ do
2. Receive an example $Z_t \leftarrow (X_t, Y_t)$ where $i = t \mod n$
3. Suffer loss $f(w^{t-1}; Z_t) = \frac{1}{2} \| w^{t-1} \|^2 + \ell(w^{t-1}; Z_t)$
4. Update $w^t$ using Algorithm 2
5. If $\| w^t \| > \frac{1}{\sqrt{\lambda}}$, $w^t \leftarrow \frac{1}{\| w^t \|} w^t$
6. end for

with entries of the Hessian matrix $H$ being $H_{ij} = Q_{ij}$, and entries of the gradient vector $\nabla$ being

$$
H = |\phi(X)|^2 \left( I_{K \times K} + I_{K \times K} \right)
$$

$$
\nabla_i = c_i = -\lambda t \left( \langle w, \nabla(X_t, Y_t^i) \rangle + \Delta(Y_t^i, Y_t) \right)
$$

Since the inverse of the sum of two matrices satisfies, $(A + B)^{-1} = A^{-1} - A^{-1} B A^{-1} \frac{1}{1 + \text{trace}(BA^{-1})}$, it follows that the inverse Hessian $H^{-1}$ is:

$$
H^{-1} = \frac{1}{|\phi(X)|^2} \left( I_{K \times K} - \frac{1}{K+1} I_{K \times K} \right)
$$

The constraint $\alpha_{y^{i,j}} \leq 0$ can be handled by sorting $c_i$ in increasing and considering a smaller matrix $Q$ that excludes entries where $\alpha_{y^{i,j}}$ would like to go below zero. The constraints $\sum_j \alpha_{y^{i,j}} \leq 1$ can be handled by reducing the step size.

2. Bounds For Structured Learning

The results of [?] can be applied to structured SVMs and can be used to provide generalization guarantees to customizable loss functions $\Delta(g(X_t; w), Y_t)$, and to bound the convergence rate of online or sequential optimization algorithms:

**Theorem 2.1** Let $f(w; Z)$ be the structured SVM objective defined in Eqn 2. Let $L = \sqrt{\lambda} + 2R$, where $\| \Psi(X, Y) \| \leq R$ is a bound on the image of $\Psi$. Then using Algorithm 1:

1. **Online Regret:** The average loss accumulated by Algorithm 2 over $T$ iterations can be bounded in relation to the minimum achievable training error:

$$
\frac{1}{T} \sum_{t=1}^{T} \Delta(g(X_t; w^{t-1}), Y_t) \leq \frac{1}{T} \min_{w: \| \Psi(X, Y) \| \leq 2R} F_T(w) + \frac{L^2 (\log(T) + 1)}{2\lambda T}
$$
2. **Generalization Error**: Let \( Z_1 \ldots Z_{n+1} \) be examples selected independently at random from some distribution \( \mathbb{P}(Z) \). With \( T = n \), the expected loss when training on \( Z_1 \ldots Z_n \) and testing on \( Z_{n+1} \) can be bounded in relation to the Bayes optimal solution:

\[
\mathbb{E}_{Z_1 \ldots Z_n} [ \mathbb{E}_Z [ \Delta (g(X; w^n), Y) ]] \leq \min_w \mathbb{E}_Z [ f(w; Z) ] + \frac{L^2 (\log(n) + 1)}{2 \lambda n} \tag{26}
\]

3. **Empirical error**: Let \( \bar{w} = \frac{1}{T} \sum_{t=1}^{T} w^{t-1} \). If Algorithm 1 is run for \( T = mn \) iterations, passing over each example \( m \geq 1 \) times, the average training error can be bounded in relation to the minimal achievable training error:

\[
\frac{1}{n} F_n(\bar{w}) \leq \left( \frac{1}{n} \min_w F_n(w) \right) + \frac{L^2 (\log(mn) + 1)}{2 \lambda mn} \tag{27}
\]

A similar bound holds for arbitrary strongly convex loss functions, and our proof closely follows the one presented in \cite{1}. One implication is that if we take only a single pass through the training set (setting \( m = 1 \)), where train time equals test time, the empirical error bound (e.g., error because we haven’t spent enough computation time) converges at the same asymptotic rate as the generalization error (e.g., error because we don’t have enough training examples). Thus when computation time is a bottleneck, it is better to use as many training examples as possible (use a large value for \( n \) and a small value for \( m \)).

A second implication is that the convergence rates for structured SVMs are the same as for linear SVMs, with the same constants involved \cite{2}, and therefore we don’t sacrifice theoretical guarantees by using an application specific loss function \( \Delta (g(X; w), Y) \).

**Proof** The main result follows because the dual objective in each step of Algorithm 1 increases by a predictable amount (see Lemma 2.2). Summing over each iteration and using weak duality, Lemma 2.4 shows that

\[
\frac{1}{T} \sum_{i=1}^{T} f(w^{t-1}; Z_i) \leq \frac{1}{T} \min_w F_T(w) + \frac{G^2 (\log(T) + 1)}{2\lambda T}
\]

where \( G \) is a bound on the gradient \( \nabla f(w^{t-1}; Z_i) \). Lemma 2.5 establishes a numerical bound on the gradient \( G = L = \sqrt{\lambda} + 2R \) for structured SVMs. Finally, Lemma 2.6 establishes that \( f(w; Z) \) is an upper bound on \( \Delta (g(X; w), Y) \). It therefore follows that

\[
\frac{1}{T} \sum_{i=1}^{T} \Delta (g(X_i; w^{t-1}), Y_i) \leq \frac{1}{T} \min_w F_T(w) + \frac{L^2 (\log(T) + 1)}{2\lambda T}
\]

thus proving Thm 2.1.1. Thm 2.1.2 is directly applicable from this regret bound and theorem 2 in \cite{3}. Lastly, by Jensen’s inequality \( \sum_{i=1}^{T} f(\bar{w}; Z_i) \leq \sum_{i=1}^{T} f(w^{t-1}; Z_i) \). Since Algorithm 2 iterates over each example \( m \) times, \( F_n(\bar{w}; Z_i) = \sum_{i=1}^{T} f(\bar{w}; Z_i) \). Thm 2.1.3 follows from Thm 2.1.1.

**Lemma 2.2** The change in the dual objective in each step in Algorithm 2 is at least

\[
D_t(\alpha_1 \ldots \alpha_{t-1}, \alpha_t) - D_{t-1}(\alpha_1 \ldots \alpha_{t-1}) \geq f(w^{t-1}; Z_t) - \frac{1}{2\lambda t} \| \nabla f(w^{t-1}; Z_t) \|^2
\]

**Proof** Consider the simpler case where each step of Algorithm 1 simply adds a new dual variable in the direction of the
subgradient $\tilde{v}_t^i = \nabla \ell(w^{t-1}; Z_t)$ with weight $\alpha_t^i = 1$. The change in the dual objective is exactly

$$
\mathcal{D}_t(\alpha_1...\alpha_{t-1}, \alpha_t^i) - \mathcal{D}_{t-1}(\alpha_1...\alpha_{t-1})
$$

$$
= \left( -\frac{\lambda t}{2} \|w_t\|^2 + \sum_{i', Y_{i'}} \alpha_{t,i'} \Delta(Y, Y_{i'}) \right) - \left( -\frac{\lambda(t-1)}{2} \|w^{t-1}\|^2 + \sum_{i', Y_{i'}} \alpha_{t,i'}' \Delta(Y, Y_{i'}) \right)
$$

$$
= -\frac{\lambda t - 2\lambda}{2t} \|w^{t-1}\|^2 + \frac{t-1}{t} \langle w^{t-1}, \tilde{v}_t^i \rangle - \frac{\lambda(t-1)}{2t} \|w^{t-1}\|^2 + \Delta(Y, \tilde{Y}_t)
$$

$$
= \left( \frac{\lambda}{2} \|w^{t-1}\|^2 + \ell(w^{t-1}, Z_t) \right) - \frac{1}{2t} \|w^{t-1} + \tilde{v}_t^i\|^2
$$

$$
= f(w^{t-1}, Z_t) - \frac{1}{2t} \|\nabla f(w^{t-1}, Z_t)\|^2
$$

Since Algorithm 1, maximizes $\mathcal{D}_t(\alpha_1...\alpha_{t-1}, \alpha_t^i) - \mathcal{D}_{t-1}(\alpha_1...\alpha_{t-1})$ over a set of samples that includes $\nabla \ell(w^{t-1}; Z_t)$, the increase in the dual objective must be at least as much. This completes the proof as long as Line 5 of Algorithm 2 does not reduce the dual objective (which we prove in Lemma 2.3):

**Lemma 2.3** The projection step in Line 5 of Algorithm 2 cannot decrease the dual objective.

**Proof** Line 5 of Algorithm 2 projects $w^t$ onto the $L_2$ ball $\|w^t\|^2 \leq \frac{1}{\lambda}$. It checks if $\|w^t\| > \frac{1}{\sqrt{\lambda}}$, and if so scales $w^t$ by

$$
s \leftarrow \frac{1}{\sqrt{\lambda}} \frac{1}{\|w^t\|}
$$

(28)

where $0 < s < 1$. The corresponding change in the dual objective is

$$
\Delta \mathcal{D}_t^{proj} = \mathcal{D}_t(s \alpha) - \mathcal{D}_t(\alpha)
$$

(29)

$$
= \left[ -\frac{t\lambda}{2} \|sw^t\|^2 + \sum_{i,Y} \alpha_i^Y \Delta(Y, Y_t) \right] - \mathcal{D}_t(\alpha)
$$

(30)

$$
= \left[ -\frac{t\lambda}{2} \|sw^t\|^2 + s \left( \frac{t\lambda}{2} \|w^t\|^2 + \mathcal{D}_t(\alpha) \right) \right] - \mathcal{D}_t(\alpha)
$$

(31)

$$
= \left[ \frac{t\lambda}{2} s^2 \|w^t\|^2 + s \left( \frac{t\lambda}{2} \|w^t\|^2 + \mathcal{D}_t(\alpha) \right) \right] - \mathcal{D}_t(\alpha)
$$

(32)

$$
= -\frac{t}{2} + \frac{s}{2s} + (s-1)\mathcal{D}_t(\alpha)
$$

(33)

$$
\geq (1-s) \left( \frac{t}{2s} - \mathcal{D}_t(\alpha) \right)
$$

(34)

$$
\geq 0
$$

(35)
where the last line follows because $1 - s > 0$ and $D_t(\alpha) \leq \frac{\lambda}{2t}$:

$$D_t(\alpha) = -\frac{t\lambda}{2} \|w^t\|^2 + \sum i\gamma \Delta(Y_i)$$

\[= -\frac{t\lambda}{2} \frac{1}{\lambda s^2} + \sum i\gamma \Delta(Y_i) \tag{37}\]

\[= -\frac{t}{2s^2} + \sum i\gamma \Delta(Y_i) \tag{38}\]

\[\leq -\frac{t}{2s^2} + t \leq -\frac{t}{2} + t \leq \frac{t}{2s} \tag{39}\]

where we have assumed $\Delta(Y_i) \leq 1$.

**Lemma 2.4** Let $G$ be a bound on $\nabla f(w_i^{-1}; Z_i)$. The average loss accumulated by Algorithm 2 over $T$ iterations is bounded by

$$\frac{1}{T} \sum_{t=1}^{T} f(w_i^{t-1}; Z_i) \leq \frac{1}{T} \min_{w} F_T(w) + \frac{G^2 (\log(T) + 1)}{2\lambda T}$$

**Proof** By Lemma 2.2 and the definition of $G$,

$$D_t(\alpha_1...\alpha_{t-1}, \gamma_i) - D_{t-1}(\alpha_1...\alpha_{t-1}) \geq f(w_i^{t-1}; Z_i) - \frac{G^2}{2\lambda t}$$

Summing over $t = 1...T$:

$$D_T(\alpha_1...\alpha_T) = \sum_{t=1}^{T} \left( D_t(\alpha_1...\alpha_{t-1}, \gamma_i) - D_{t-1}(\alpha_1...\alpha_{t-1}) \right) \geq \sum_{t=1}^{T} f(w_i^{t-1}; Z_i) - \frac{G^2}{2\lambda} \sum_{t=1}^{T} \frac{1}{t}$$

Since $\sum_{t=1}^{T} \frac{1}{t} \leq \log(T) + 1$

$$D_T(\alpha_1...\alpha_T) \geq \sum_{t=1}^{T} f(w_i^{t-1}; Z_i) - \frac{G^2 (\log(T) + 1)}{2\lambda}$$

Applying weak duality $D_T(\alpha_1...\alpha_T) \leq \min_{w} F_T(w)$ and rearranging terms completes the proof.

**Lemma 2.5** The magnitude of the gradient of the structured SVM error $\nabla f(w_i^{t-1}; Z_i)$ in Algorithm 2 is bounded $\|\nabla f(w_i^{t-1}; Z_i)\| \leq \sqrt{X} + 2R$, where $\|\Psi(X, Y)\| \leq R$ is a bound on the image of $\Psi$.

**Proof** Since $\nabla f(w_i^{t-1}; Z_i) = \Psi(X_i, Y_i) - \Psi(X_i, Y_i)$ and $\|\Psi(X, Y)\| \leq R$ for all $X, Y$, it must be the case that $\|\nabla f(w_i^{t-1}; Z_i)\| \leq 2R$. Line 5 of Algorithm 2 ensures that $\|w_i^{t-1}\| \leq \frac{1}{\sqrt{X}}$. Since $\nabla f(w_i^{t-1}; Z_i) = \lambda w_i^{t-1} + \nabla l(w_i^{t-1}; Z_i)$, by the triangle inequality, $\|\nabla f(w_i^{t-1}; Z_i)\| \leq \sqrt{X} + 2R$.

**Lemma 2.6** Let $g(X; w) = \arg\max_{Y} \langle w, \Psi(X, Y) \rangle$ be the label predicted by model parameters $w$. The loss associated with this prediction is upper-bounded by the structured hinge loss: $\ell(w; Z) \geq \Delta(g(X; w), Y)$

**Proof**

$$\max_{Y'} \langle w, \Psi(X, Y') \rangle + \Delta(Y', Y) \geq \langle w, \Psi(X, g(X; w)) \rangle + \Delta(g(X; w), Y)$$

$$\max_{Y'} \langle w, \Psi(X, Y') \rangle + \Delta(Y', Y) \geq \langle w, \Psi(X, Y) \rangle + \Delta(g(X; w), Y)$$

$$\max_{Y'} \langle w, \Psi(X, Y') \rangle + \Delta(Y', Y) - \langle w, \Psi(X, Y) \rangle \geq \Delta(g(X; w), Y)$$

$$\ell(w; Z) \geq \Delta(g(X; w), Y)$$

**References**