Implicit Online Learning with Kernels

S.V.N. Vishwanathan
Department of Statistics
Purdue University
250 N University Street
West Lafayette, IN, 47907-2066, USA

Li Cheng
College of Engineering and Computer Science
Australian National University –and–
NICTA, Locked Bag 8001
Canberra ACT 2601, Australia

VISHY@STAT.PURDUE.EDU
LI.CHENG@NICTA.COM.AU

Editor: U. N. Known

Abstract

We present a new algorithm for online learning in reproducing kernel Hilbert spaces (RKHS). Our algorithm, ILK (implicit online learning with kernels), employs a new, implicit update technique that can be applied to a wide variety of convex loss functions. We prove mistake bounds and analyze the convergence rates of our algorithm. We also point out connections to a number of existing algorithms.

Keywords: Online Learning, Bregman Divergences, Implicit Updates, Reproducing Kernel Hilbert Spaces, Hinge Loss, Structured Learning, Loss Functions, Subgradients.

1. Introduction

Online learning refers to a paradigm where at time $t$ an instance $x_t \in \mathcal{X}$ is presented to a learner, which uses its parameter vector $f_t$ to predict a label. This predicted label is then compared to the true label $y_t \in \mathcal{Y}$ via a non-negative, piecewise differentiable, and convex loss function $L(x_t, y_t, f_t)$. The learner then updates its parameter vector to minimize a convex risk function, and the process repeats.

Kivinen and Warmuth (1997) proposed a generic framework for online learning where the risk function, $J_t(f)$, to be minimized consists of two terms: a Bregman divergence between parameters $\Delta_G(f, f_t) := G(f) - G(f_t) - \langle f - f_t, \partial_f G(f_t) \rangle$ defined via a twice differentiable strictly convex function $G$, and the instantaneous loss $L(x_t, y_t, f)$. The parameter updates are then derived via the principle

$$f_{t+1} = \arg\min_f J_t(f) := \arg\min_f \{\Delta_G(f, f_t) + \eta_t L(x_t, y_t, f)\},$$

where $\eta_t$, the learning rate, is a tunable parameter. Since $J_t(f)$ is convex, (1) is solved by setting the gradient (or, if necessary, a subgradient) to 0. Using the fact that $\partial_f \Delta_G(f, f_t) =$
\[ \partial f G(f) - \partial f G(f_t), \text{ we obtain} \]
\[ \partial f G(f_{t+1}) = \partial f G(f_t) - \eta_t \partial f L(x_t, y_t, f_{t+1}). \quad (2) \]

Note the dependency on \( f_{t+1} \) on both the left as well as the right hand sides of the above equation. Therefore, it is difficult to determine \( \partial f L(x_t, y_t, f_{t+1}) \) in closed form. An explicit update, as opposed to the above implicit update, uses the approximation \( \partial f L(x_t, y_t, f_{t+1}) \approx \partial f L(x_t, y_t, f_t) \) to arrive at the more easily computable expression \( \text{(Kivinen et al., 2006)} \)

\[ \partial f G(f_{t+1}) = \partial f G(f_t) - \eta_t \partial f L(x_t, y_t, f_t). \quad (3) \]

For instance, if we set \( G(f) = \frac{1}{2} \| f \|^2 \), then \( \Delta G(f, f_t) = \frac{1}{2} \| f - f_t \|^2 \) and \( \partial f G(f) = f \), which results in the familiar stochastic gradient descent update

\[ f_{t+1} = f_t - \eta_t \partial f L(x_t, y_t, f_t). \quad (4) \]

In this paper, we are interested in online learning in a reproducing kernel Hilbert space (RKHS) \( \text{(Kivinen et al., 2004)} \). To lift the above framework into an RKHS, \( \mathcal{H} \), we restrict our attention to \( f \in \mathcal{H} \) and use

\[ G(f) = \frac{1}{2} \| f \|^2_{\mathcal{H}}. \quad (5) \]

Recall that if \( \mathcal{H} \) is an RKHS of functions \( f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \), then its defining kernel \( k : (\mathcal{X} \times \mathcal{Y})^2 \rightarrow \mathbb{R} \) satisfies the reproducing property; namely that \( \langle f, k((x, y), \cdot) \rangle_{\mathcal{H}} = f(x, y) \) for all \( f \in \mathcal{H} \). Therefore, by making the standard assumption that \( L \) only depends on \( f \) via its evaluations at \( f(x, y) \), one reaches the conclusion that \( \partial f L(x, y, f) \in \mathcal{H} \). In particular

\[ \partial f L(x, y, f) = \sum_{y \in \mathcal{Y}} \beta_y k((x, y), \cdot), \quad (6) \]

for some coefficients \( \beta_y \in \mathbb{R} \). Choose \( f_0 = 0 \). Plugging in the fact that \( \partial f G(f) = f \) into either (2) or (3) shows that there must exist coefficients \( \alpha_{i,y} \) fully specifying \( f_{t+1} \) via

\[ f_{t+1} = \sum_{i=0}^{t} \sum_{y \in \mathcal{Y}} \alpha_{i,y} k((x_i, y), \cdot). \quad (7) \]

In this paper we present a general recipe for performing implicit updates, (2), in RKHS, or equivalently, we present an algorithm that computes the coefficients \( \alpha_{i,y} \) for an implicit update.

### 1.1 Paper Contributions

We propose an algorithm template ILK (implicit online learning with kernels) that provides a general framework for computing implicit updates in an RKHS, and apply it many popular loss functions namely, quadratic, hinge, ordinal regression, and logistic losses, as well as their extensions to structured domains (see e.g. Taskar et al. (2004); Tsochantaridis et al. (2005); Cai and Hofmann (2004); Vishwanathan et al. (2006)). Along the
way, we point out connections to a number of existing algorithms including the Perceptron (Rosenblatt, 1958), Kernel-Adatron (Frieß et al., 1998), NORMA (Kivinen et al., 2004), online passive-aggressive algorithms (Crammer et al., 2006), PRanking (Crammer and Singer, 2005), additive online algorithms for category ranking (Crammer and Singer, 2003b), and support vector ordinal regression (Chu and Keerthi, 2005). When viewed in our framework, these algorithms either reduce to explicit updates, or implicit updates, albeit, motivated differently. This makes explicit the connections between these seemingly disparate algorithms and explains why the mistake bounds of many of them are very similar. We prove mistake bounds for our algorithms by lower bounding the dual improvement per iteration, a technique pioneered by Shalev-Shwartz and Singer (2006) and Tsochantaridis et al. (2005).

2. Implicit Updates in an RKHS

Recall from Section 1 that the goal of goal of online learning is to minimize the risk functional $J_t(f) = \Delta_G(f, f_t) + \eta t L(x_t, y_t, f)$. By plugging $\Delta_G(f, f_t) = \frac{1}{2} \|f - f_t\|^2$ into (1) we obtain

$$J_t(f) = \frac{1}{2} \|f - f_t\|^2 + \eta t L(x_t, y_t, f) \text{ with } f_{t+1} = \arg\min_f J_t(f). \quad (8)$$

Since $L$ is assumed convex with respect to $f$, setting $\partial_f J(f) = 0$ yields the implicit update

$$f_{t+1} = f_t - \eta t \partial_f L(x_t, y_t, f_{t+1}). \quad (9)$$

From (7) it follows that $f_{t+1}$ can also be written as

$$f_{t+1} = \sum_{i=0}^{t-1} \sum_{y \in Y} \alpha_{i,y} k((x_i, y), \cdot) + \sum_{y \in Y} \alpha_{t,y} k((x_t, y), \cdot) \quad (10)$$

$$= f_t + \sum_{y \in Y} \alpha_{t,y} k((x_t, y), \cdot) \quad (11)$$

for some coefficients $\alpha_{i,y} \in \mathbb{R}$. Comparing (9) and (11) and using

$$\partial_f L(x_t, y_t, f_{t+1}) = \sum_{y \in Y} \beta_{t,y} k((x_t, y), \cdot), \quad (12)$$

it is easy to see that

$$\alpha_{i,y} = \alpha_{i,y} \quad \text{ for } i = 0, \ldots, t-1, \text{ and all } y \in \mathcal{Y},$$

$$\alpha_{t,y} = -\eta t \beta_{t,y} \text{ for all } y \in \mathcal{Y}. \quad (13)$$

For ease of exposition we assume a fixed step size (learning rate) $\eta_t = \eta$. Consequently

$$\alpha_{i,y} = \alpha_{i,y} \quad \text{ for } i = 0, \ldots, t-1, \text{ and } y \in \mathcal{Y},$$

$$\alpha_{t,y} = -\eta \beta_{t,y} \text{ for all } y \in \mathcal{Y}. \quad (14)$$

We note in the passing that even though we assume a constant step size, sophisticated step size adaptation algorithms (e.g. Vishwanathan et al., 2006) can be modified in a straightforward manner to work in this setting. It is also customary to decay previous coefficients
by setting \( \alpha_{i,y} \leftarrow (1 - \tau)\alpha_{i,y} \) for \( i = 0, \ldots, t - 1 \) and some decay factor \( \tau < 1 \). This is particularly helpful in dealing with moving targets, that is, when the underlying true hypothesis changes over time (Herbster and Warmuth, 2001; Kivinen et al., 2004). Although our analysis can be extended to handle this, we do not consider this situation in the sequel.

Returning to the implicit updates, the main difficulty in performing such an update arises from the fact that \( \beta_{t,y} \) depends on \( f_{t+1} \) (e.g. in (16) of the square loss) which in turn depends on \( \beta_{t,y} \) via \( \alpha_{t,y} \). The general recipe to overcome this problem is to first use (9) to write \( \beta_{t,y} \) as a function of \( \alpha_{t,y} \). Plugging this back into (14) yields an equation in \( \alpha_{t,y} \) alone, which sometimes can be solved efficiently. We now elucidate details for some well-known loss functions. Detailed derivations for a number of other losses is relegated to Appendix A.

2.1 Square Loss

In this case, \( k((x_t, y), \cdot) = k(x_t, \cdot) \), that is, the kernel does not depend on the value of \( y \). Furthermore, we assume that \( \mathcal{Y} = \mathbb{R} \) and define

\[
L(x_t, y_t, f) := \frac{1}{2} (f(x_t) - y_t)^2 = \frac{1}{2} (\langle f(\cdot), k(x_t, \cdot) \rangle_{\mathcal{H}} - y_t)^2,
\]

(15)

which yields

\[
\partial_f L(x_t, y_t, f) = (f(x_t) - y_t) k(x_t, \cdot).
\]

(16)

Substituting into (14) and using (9) we have

\[
\alpha_t = -\eta (f_{t+1}(x_t) - y_t) = -\eta (f_t(x_t) + \alpha_t k(x_t, x_t) - y_t),
\]

(17)

which can be rearranged to

\[
\alpha_t = \frac{y_t - f_t(x_t)}{1 + \eta k(x_t, x_t)}.
\]

(18)

Contrast this with the explicit update:

\[
\alpha_t = \eta (y_t - f_t(x_t)).
\]

(19)

To show connections to existing algorithms we set \( \mathcal{H} = \mathbb{R}^n \) for some \( n \) in (18) to obtain

\[
\alpha_t = \frac{\eta (y_t - f_t(x_t))}{1 + \eta \|x_t\|^2},
\]

(20)

where we used the fact that in \( \mathbb{R}^n \) the feature map \( k(x, \cdot) = x \) and hence \( k(x, x) = \|x\|_2^2 \). Plugging this into (9) leads to

\[
f_{t+1} = f_t - \frac{\eta (f_t(x_t) - y_t)}{1 + \eta \|x_t\|^2} x_t,
\]

(21)

which is identical to Eq (4.8) of Kivinen and Warmuth (1997) and Eq (13) of Kivinen et al. (2006).
2.2 Binary Hinge Loss

As before, assume \( k((x_t, y_t), \cdot) = k(x_t, \cdot) \), and set \( \mathcal{Y} = \{ \pm 1 \} \). The hinge loss for binary classification is defined as

\[
L(x_t, y_t, f) := (\rho - y_t f(x_t))^+ = (\rho - y_t (f(x_t), \cdot))_{+},
\]

where \( \rho \geq 0 \) is the margin parameter and \( (\cdot)_{+} := \max(0, \cdot) \) is the hinge function. The binary hinge loss is convex but not differentiable at the hinge point.

Even though a convex function, \( G \), might not be differentiable everywhere, a subgradient always exists (Rockafellar, 1970; Hiriart-Urruty and Lemaréchal, 1993). Let \( f \) be a point where \( G \) is finite. Then a subgradient is the normal vector of any tangential supporting hyperplane of \( G \) at \( f \). Formally, \( g \) is called a subgradient of \( G \) at \( f \) if and only if

\[
G(f') \geq G(f) + \langle g, f' - f \rangle \quad \forall f'.
\]

The set of all subgradients at a point is called the subdifferential, and is denoted \( \partial G(f) \). If this set is not empty then \( G \) is said to be subdifferentiable at \( f \). If it contains exactly one element, \( i.e., \partial_f G(f) = \{ \nabla_f G(f) \} \), then \( G \) is said to be differentiable at \( f \).

Therefore, the subgradient \( \partial_f L(x_t, y_t, f) \) of the binary hinge loss exists everywhere and can be written as \( \beta_t k(x_t, \cdot) \), where

\[
\beta_t \in \begin{cases} 
0 & \text{if } y_t f(x_t) > \rho \\
[0, -y_t] & \text{if } y_t f(x_t) = \rho \\
-y_t & \text{if } y_t f(x_t) < \rho.
\end{cases}
\]

We need to balance between two conflicting requirements while computing \( \alpha_t \). On one hand we want the loss to be zero, which can be achieved by setting \( \rho - y_t f_{t+1}(x_t) = 0 \). On the other hand, the gradient of the loss at the new point \( \partial_f L(x_t, y_t, f_{t+1}) \) must satisfy (24). We satisfy both requirements by appropriately clipping the optimal estimate of \( \alpha_t \).

Let \( \hat{\alpha}_t \) denote the optimal estimate of \( \alpha_t \) which leads to \( \rho - y_t f_{t+1}(x_t) = 0 \). Using (9) we have \( \rho - y_t (f_t(x_t) + \hat{\alpha}_t k(x_t, x_t)) = 0 \), which can be rearranged to

\[
\hat{\alpha}_t = \frac{\rho - y_t f_t(x_t)}{y_t k(x_t, x_t)} = \frac{y_t (\rho - y_t f_t(x_t))}{k(x_t, x_t)}.
\]

But, using (24) and (14) we have that \( \alpha_t \in [0, \eta y_t] \) or equivalently \( y_t \alpha_t \in [0, \eta] \), by noting that \( y_t \cdot y_t = 1 \). By combining (25) and the need to clip \( \alpha_t \) we arrive at the final implicit update:

\[
\alpha_t = \begin{cases} 
\hat{\alpha}_t & \text{if } y_t \hat{\alpha}_t \in [0, \eta] \\
0 & \text{if } y_t \hat{\alpha}_t < 0 \\
\eta y_t & \text{if } y_t \hat{\alpha}_t > \eta.
\end{cases}
\]
Setting $\rho = 0$ and $\eta = 1$ into (25) and observing that $y_t \cdot y_t = 1$ yields $\hat{\alpha}_t = \frac{-f_t(x_t)}{k(x_t, x_t)}$. Clipping this estimate to satisfy $y_t \alpha_t \in [0, 1]$ recovers the binary Margin Infused Relaxed Algorithm (binary MIRA) of Crammer and Singer (see Figure 4 of Crammer and Singer (2003a)):

$$
\alpha_t = \begin{cases} 
\hat{\alpha}_t & \text{if } y_t \hat{\alpha}_t \in [0, 1] \\
0 & \text{if } y_t \hat{\alpha}_t < 0 \\
y_t & \text{if } y_t \hat{\alpha}_t > 1.
\end{cases}
$$

(28)

On the other hand, setting $\rho = 0$ and $\eta = 1$ into the explicit update recovers the Perceptron algorithm (Rosenblatt, 1958; Minsky and Papert, 1969).

2.3 Multiclass Hinge Loss

A variety of multiclass hinge losses have been proposed in literature that generalize the binary hinge loss and enforce a uniform margin of separation between the true label $y_t$ and every other label $y \neq y_t$. Here we present two such examples and their corresponding implicit and explicit update formulae.

2.3.1 Additive Multiclass Hinge Loss

A natural generalization of the binary hinge loss is to penalize all violating labels (Weston and Watkins, 1998):

$$
L(x_t, y_t, f) := \sum_{y \neq y_t} (\rho - \Delta f(y_t, y))_+, \text{ where } \Delta f(y_t, y) := f(x_t, y_t) - f(x_t, y).
$$

(29)

The binary hinge loss is recovered by setting $f(x, y) = \frac{y}{2} f(x)$ for $y \in \{\pm 1\}$ which yields

$$
\sum_{y \neq y_t} (\rho - \Delta f(y_t, y))_+ = (\rho - f(x_t, y_t) + f(x_t, -y_t))_+ = (\rho - y_t f(x_t))_+.
$$

(30)

A label $y$ is said to be margin violating if $\rho - \Delta f(y_t, y) \geq 0$, and its magnitude of violation is defined as $e(y) := \rho - \Delta f(y_t, y)$. By defining $\mathcal{E}_t := \{y \neq y_t \text{ s.t. } \rho - \Delta f(y_t, y) \geq 0\}$, it can be verified that the subgradient of the loss can be written as $\sum_y \beta_{t,y} k((x_t, y), \cdot)$, where $\sum_y \beta_{t,y} = 0$ and

$$
\beta_{t,y} \in [0, 1] \text{ for } y \in \mathcal{E}_t, \beta_{t,y} = -\sum_{y \notin \mathcal{E}_t} \beta_{t,y}, \text{ and all other } \beta_{t,y} = 0.
$$

(31)

An implicit update sets the loss after the update to be zero, but at the same time ensures that the gradient evaluated at $f_{t+1}$ satisfies (31). For the additive multiclass hinge loss, (29), both these objectives can be satisfied by setting $\rho - \Delta f_{t+1}(y_t, y) = 0$ for all $y \neq y_t$. Let $\hat{\alpha}_{t,y}$ be the optimal coefficients that lead to zero loss, and let $k_t(y, y') := k((x_t, y), (x_t, y'))^1$. Using (9), for all $y \neq y_t$ we have

$$
\rho - \Delta f_t(y_t, y) - \sum_{y'} \hat{\alpha}_{t,y'} k_t(y', y_t) + \sum_{y'} \hat{\alpha}_{t,y'} k_t(y', y) = 0,
$$

(32)

1. $k_t$ is a valid positive semi-definite kernel on $\mathcal{Y} \times \mathcal{Y}$, associated with the feature map $y \rightarrow k((x_t, y), (x_t, \cdot))$.
which can be rearranged to
\[ \sum_{y'} (k_t(y', y_t) - k_t(y', y)) \alpha_{t,y'} = \rho - \Delta f_t(y_t, y). \]

Furthermore, since \( \sum_y \beta_{t,y} = 0 \) it follows that \( \sum_y \hat{\alpha}_{t,y} = 0 \). All in all we get \(|\mathcal{Y}| \) linear equations in the \(|\mathcal{Y}| \) unknown variables \( \hat{\alpha}_{t,y} \):

\[ M\alpha = b. \]

Here \( M \) is a \( |\mathcal{Y}| \times |\mathcal{Y}| \) matrix with \( M[y_t, y'] = 1 \) and \( M[y, y'] = k_t(y', y_t) - k_t(y', y) \) for \( y \neq y_t \), while \( \alpha \) is the \( |\mathcal{Y}| \) length vectors of unknowns \( \alpha[y'] = \hat{\alpha}_{t,y'} \), and \( b \) is a \( |\mathcal{Y}| \) length vector with \( b[y_t] = 1 \) and \( b[y'] = \rho - \Delta f_t(y_t, y) \) for \( y \neq y_t \). The values of \( \hat{\alpha}_{t,y} \) found by solving the above linear system must then be clipped appropriately to satisfy (31). Note that if (32) is satisfied then the \( \mathcal{E}_t \) at \( f_{t+1} \) contains all \( y \neq y_t \).

As a special case consider the decomposing kernel \( k_t(y, y') = k(x_t, x_t) \cdot I_{y=y'} \), with \( k(x_t, x_t) \) a kernel on \( \mathcal{X} \times \mathcal{X} \), and \( I_{y=y'} = 1 \) if \( y = y' \) and 0 otherwise. Now (32) reduces to

\[ \rho - \Delta f_t(y_t, y) - (\hat{\alpha}_{t,y_t} - \hat{\alpha}_{t,Y}) k(x_t, x_t) = 0 \text{ for all } y \neq y_t. \]

Enforcing the constraint \( \sum_y \hat{\alpha}_{t,y} = 0 \) and some tedious but straightforward algebra (not shown here) yields

\[ \hat{\alpha}_{t,y} = \begin{cases} \nu_y + \frac{\rho}{k(x_t, x_t)} & \text{if } y = y_t \\ \nu_y & \text{otherwise}, \end{cases} \quad \text{where } \nu_y = -\frac{\rho + \sum_{y'} \Delta f_t(y', y)}{|\mathcal{Y}| \cdot k(x_t, x_t)}. \]

The clipped updates \( \hat{\alpha}_{t,y} \) can be found by finding an appropriate \( \rho^* \), which can be done iteratively by solving a fixed point iteration (Crammer and Singer, 2000) or by a binary search (Crammer and Singer, 2001).

In the special case of the decomposing kernel we can also perform an implicit update by only modifying the coefficients of the error set \( y \in \mathcal{E}_t \). In other words we set

\[ \rho - \Delta f_t(y_t, y) - (\hat{\alpha}_{t,y_t} - \hat{\alpha}_{t,Y}) k(x_t, x_t) = 0 \text{ for } y \in \mathcal{E}_t, \text{ and all other } \hat{\alpha}_{t,y} = 0. \]

Implicit updates are obtained by clipping \( \hat{\alpha}_{t,y} \) in order to satisfy (31). To show that this is a valid strategy we need to show that the loss after the update is zero. Clearly for all \( y \in \mathcal{E}_t \) we have \( \rho - \Delta f_{t+1}(y_t, y) = 0 \) and hence their contribution to \( L(x_t, y_t, f_{t+1}) \) is zero. What remains to show is that \( \rho - \Delta f_{t+1}(y_t, \bar{y}) \leq 0 \) for all \( \bar{y} \in \mathcal{E}_t := \{y \neq y_t, y \notin \mathcal{E}_t\} \). Since \( \hat{\alpha}_{t,y} = 0 \) for all \( \bar{y} \in \mathcal{E}_t \) this is equivalent to showing that \( \rho - \Delta f_t(y_t, \bar{y}) - \hat{\alpha}_{t,y_t} k(x_t, x_t) \leq 0 \).

By definition \( \rho - \Delta f_t(y_t, \bar{y}) \leq 0 \) for all \( \bar{y} \in \mathcal{E}_t \). Furthermore, \( k(x_t, x_t) \geq 0 \). Therefore, it is suffices to show that \( \hat{\alpha}_{t,y_t} \geq 0 \). Towards this end we rearrange (37) to obtain

\[ (\hat{\alpha}_{t,y_t} - \hat{\alpha}_{t,Y}) = \frac{\rho - \Delta f_t(y_t, y)}{k(x_t, x_t)} \text{ for all } y \in \mathcal{E}_t. \]

By definition \( \rho - \Delta f_t(y_t, y) \geq 0 \) for all \( y \in \mathcal{E}_t \), while \( k(x_t, x_t) \geq 0 \). It follows that \( \hat{\alpha}_{t,y_t} \geq \hat{\alpha}_{t,y} \) for all \( y \in \mathcal{E}_t \). Summing the inequality obtains \( |\mathcal{E}_t| \cdot \hat{\alpha}_{t,y_t} \geq \sum_{y \in \mathcal{E}_t} \hat{\alpha}_{t,y} \). Since \( \hat{\alpha}_{t,y} = 0 \) for all \( y \in \mathcal{E}_t \) one can rewrite the inequality as \((|\mathcal{E}_t| + 1) \cdot \hat{\alpha}_{t,y_t} \geq \sum_{y} \hat{\alpha}_{t,y} = 0 \), and hence \( \hat{\alpha}_{t,y_t} \geq 0 \).

---

2. We use \(| \cdot |\) to denote the cardinality of a set.
In order to perform an explicit update we set \( \alpha_{t,y} = -\eta \beta_{t,y} \) for all \( y \in \mathcal{Y} \), and coefficients \( \beta_{t,y} \) as follows (see also section 4 and Figure 2 of Crammer and Singer (2003a) or Section 6 of Crammer and Singer (2003b)):

**Uniform Update:** \( \beta_{t,y} = \frac{1}{|\mathcal{E}_t|} \) for \( y \in \mathcal{E}_t, \beta_{t,y_t} = -1 \), and all other \( \beta_{t,y} = 0 \).

**Margin-proportional Update:** \( \beta_{t,y} = \frac{\epsilon(y)}{L(x_{t,y}, f_t)} \) for \( y \neq y_t \), and \( \beta_{t,y_t} = -1 \).

**Randomized Update:** \( \beta_{t,y} = 1 \) for some \( y \in \mathcal{E}_t, \beta_{t,y_t} = -1 \), and all other \( \beta_{t,y} = 0 \).

### 2.3.2 Maximum Multiclass Hinge Loss

We now consider a variant of (29) that penalizes only the maximally violating label:

\[
L(x_t, y_t, f) := \left( \max_{y \neq y_t} \rho - \Delta f(y_t, y) \right)_+, \quad \text{where } \Delta f(y_t, y) := f(x_t, y_t) - f(x_t, y).
\]  

Denote \( f_t^* := \max_{y \neq y_t} \rho - \Delta f(y_t, y) \), and \( \mathcal{Y}_t^* := \arg\max_{y \neq y_t} \rho - \Delta f(y_t, y) = \arg\max_{y \neq y_t} f(x_t, y) \), then the subgradient of the loss can be written as \( \sum_y \beta_{t,y} k((x_t, y), \cdot) \), where \( \sum_y \beta_{t,y} = 0 \) and

- if \( f_t^* < 0 \) then \( \beta_{t,y} = 0 \) for all \( y \)
- if \( f_t^* = 0 \) then \( \beta_{t,y_t} \in [0, 1] \) for \( y_t^* \in \mathcal{Y}_t^* \), \( \beta_{t,y_t} \in [-1, 0] \), and all other \( \beta_{t,y} = 0 \)  
- if \( f_t^* > 0 \) then \( \beta_{t,y_t} \in [0, 1] \) for \( y_t^* \in \mathcal{Y}_t^* \), \( \beta_{t,y_t} = -1 \), and all other \( \beta_{t,y} = 0 \).

Note that the constraint \( \sum_y \beta_{t,y} = 0 \) along with the fact that \( \beta_{t,y_t} \in [-1, 0] \) effectively ensure that \( \sum_{y_t^*} \beta_{t,y_t^*} \) is constrained to be in the interval \([0, 1]\). Contrast this with the additive multiclass loss where \( \beta_{t,y_t} \) need not necessarily lie in the interval \([-1, 0]\).

As in the additive case the implicit update can set the loss after the update to be zero by ensuring that

\[
\rho - \Delta f_t(y_t, y) - \sum_{y'} \hat{\alpha}_{t,y_i} k_t(y', y_t) + \sum_{y'} \hat{\alpha}_{t,y_i} k_t(y', y) = 0,
\]  

but now the final update is obtained by clipping \( \hat{\alpha}_{t,y} \) in order to satisfy (40). Again consider the decomposing kernel \( k_t(y', y) = k(x_t, x_t) \cdot I_{y=y'} \). Now (41) reduces to

\[
\rho - \Delta f_t(y_t, y) - (\hat{\alpha}_{t,y_t} - \hat{\alpha}_{t,y_t}) k(x_t, x_t) = 0 \quad \text{for all } y \neq y_t.
\]  

Solving this yields (36), which can be clipped to satisfy (40). In the case of the decomposing kernel one can also follow Crammer et al. (2006) and modify just two coefficients: \( \alpha_{t,y_t} \) and an arbitrary \( \alpha_{t,y_t^*} \) for some \( y_t^* \in \mathcal{Y}_t^* \). Specifically, we set \( \alpha_t = \alpha_{t,y_t} = -\alpha_{t,y_t^*} \) and all other \( \alpha_{t,y} = 0 \) and ensure that

\[
\rho - \Delta f_t(y_t, y^*_t) - (\hat{\alpha}_{t,y_t} - \hat{\alpha}_{t,y_t^*}) k(x_t, x_t) = 0.
\]  

Solving for \( \hat{\alpha}_t \) the optimal estimate of \( \alpha_t \) yields

\[
\hat{\alpha}_t = \frac{\rho - \Delta f_t(y_t, y^*_t)}{2k(x_t, x_t)}.
\]
As before, the final update is obtained by appropriately clipping $\alpha_t$ to be in $[0, \eta]$. In order to show that this is a valid strategy one needs to show that the loss after the update is zero. This is easy to see because $\alpha_t \geq 0$ and hence the update ensures that $f_{t+1}(x_t, y_t^*) \geq f_{t+1}(x_t, y)$ for every $y \neq y_t$. This in turn means that $\mathcal{Y}_{t+1}^* = y_t^*$, but we already ensured that $\rho - \Delta f_{t+1}(y_t, y_t^*) = 0$ via (43).

For an explicit update one can use any member of the set of subgradients of the loss. For instance, whenever $f_t^* > 0$ we can set $\alpha_{t,y} = -\eta \beta_{t,y}$ for all $y \in \mathcal{Y}$, where

1. $\beta_{t,y_t^*} = \frac{1}{|\mathcal{Y}_t^*|}$ for $y_t^* \in \mathcal{Y}_t^*$, $\beta_{t,y_t} = -1$, and all other $\beta_{t,y} = 0$.
2. $\beta_{t,y_t^*} = 1$, for an arbitrary $y_t^* \in \mathcal{Y}_t^*$, $\beta_{t,y_t} = -1$, and all other $\beta_{t,y} = 0$.

2.4 Logistic Regression Loss

The logistic regression loss and its gradient can be written as

$$ L(x_t, y_t, f) := \log (1 + \exp(-y_t f(x_t))) $$

and

$$ \partial_f L(x_t, y_t, f) = \frac{-y_t k(x_t, \cdot)}{1 + \exp(y_t f(x_t))} $$

respectively. Using (9) and (14), we obtain

$$ \alpha_t = \frac{\eta y_t}{1 + \exp(y_t f(x_t) + \alpha_t y_t k(x_t, x_t))}. $$

(46)

Although this equation does not give a closed-form solution, the value of $\alpha_t$ can still be obtained by using a numerical root-finding routine (e.g. Press et al., 1992). The explicit updates in this case are given by

$$ \alpha_t = \frac{\eta y_t}{1 + \exp(y_t f(x_t))}. $$

(47)

3. Motivating Implicit Updates from the Dual

So far we have motivated implicit updates via minimizing the risk functional $J_t(f)$ defined as the sum of a Bregman divergence and a instantaneous risk (see e.g. (8)). There exists an equally elegant motivation via the Lagrangian dual, which we now elucidate. Our object of study in this section will be the batch version of the risk functional, (8), which at time instant $t$ can be written as

$$ P_t(f) := \frac{1}{2} \| f \|_H^2 + \eta \sum_{i=0}^t L(x_i, y_i, f). $$

(48)

We will derive its Lagrangian dual for various loss functions and show that online learning algorithms change exactly one dual coefficient per iteration (see also Shalev-Shwartz and Singer (2006)). In particular, implicit updates maximize the change in the dual, but by optimizing exactly one dual variable per iteration. In order to keep the presentation succinct we will only deal with some representative losses in the main body of the paper and delegate detailed derivations for the rest to Appendix B.
3.1 Binary Hinge Loss

First, we derive the Lagrange dual for the binary hinge loss (22). Towards this end we first rewrite the problem of minimizing the batch objective function, (48), as a constrained optimization problem:

\[
\min_{f, \xi_i} \frac{1}{2} \|f\|^2 + \eta \sum_{i=0}^{t} \xi_i \\
\text{s. t. } y_i \langle f, k(x_i, \cdot) \rangle \geq \rho - \xi_i, \xi_i \geq 0.
\]  

(49)

The corresponding Lagrangian is

\[
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} \|f\|^2 + \eta \sum_{i=0}^{t} \xi_i - \sum_{i=0}^{t} \gamma_i (y_i \langle f, k(x_i, \cdot) \rangle - \rho + \xi_i) - \sum_{i=0}^{t} \tau_i \xi_i,
\]

(50)

where \(\Xi_t := \{\xi_0, \ldots, \xi_t\}, \Gamma_t := \{\gamma_0, \ldots, \gamma_t\}, T_t := \{\tau_0, \ldots, \tau_t\}\) with \(\tau_i \geq 0\) and \(\gamma_i \geq 0\) for all \(i\). Taking gradients with respect to the primal variables and setting them to zero yields

\[
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^{t} \gamma_i y_i k(x_i, \cdot) \quad \text{and} \\
\frac{\partial L_t}{\partial \xi_i} = 0 \implies \eta - \gamma_i - \tau_i = 0.
\]

(51)

(52)

Since both \(\tau_i\) and \(\gamma_i\) are \(\geq 0\), it follows from (52) that \(\gamma_i \in [0, \eta]\). Plugging in (51) and (52) into (50) yields the dual:

\[
D_t(\Gamma_t) = \rho \sum_{i=0}^{t} \gamma_i - \frac{1}{2} \sum_{i=0}^{t} \sum_{j=0}^{t} \gamma_i y_i \gamma_j y_j k(x_i, x_j) \quad \text{s. t. } \gamma_i \in [0, \eta] \quad \text{for all } i.
\]

(53)

A greedy yet inefficient online learning algorithm minimizes \(P_t(f)\) at every time instant \(t\). Since \(P_t\) is convex strong duality applies, that is, if \(f^* = \text{argmin}_f P_t(f)\) and \(\Gamma^* = \text{argmax}_\Gamma D_t(\Gamma)\) then \(P_t(f^*) = D_t(\Gamma^*)\). Furthermore, for any \(f\) and dual feasible \(\Gamma\) we have \(P_t(f) \geq D_t(\Gamma)\). Therefore, a valid strategy to minimize \(P_t\) is to maximize the dual \(D_t(\Gamma_t)\). To understand the connection between this algorithm and our setting compare (51) and (7). In particular, (51) shows that minimizing the batch risk functional (48) yields \(f_t = \sum_{i=0}^{t} \gamma_i y_i k(x_i, \cdot)\) at time instant \(t\). On the other hand, (7) shows that online updates also produce a hypothesis \(f_t = \sum_{i=0}^{t} \alpha_i k(x_i, \cdot)\) at time \(t\). The key difference is that in the greedy algorithm we are allowed to modify all the dual coefficients \(\gamma_i\) for \(i = 0, \ldots, t\), while in our setting we are restricted to modifying only one coefficient namely \(\alpha_t\) (see (14)). Furthermore, any dual feasible solution \(\Gamma_t\) produces a valid predictor \(f = \sum_{i=0}^{t} \gamma_i y_i k(x_i, \cdot)\) via the dual connection (51). Our algorithm produces a dual feasible solution after every update. To see this set \(\gamma_i = y_i \alpha_i\) and recall that the implicit updates described in Section 2.2 ensure that \(y_i \alpha_i \in [0, \eta]\) for all \(i\).

Let \(D_{t-1}(\Gamma_{t-1})\) denote the dual at time instance \(t - 1\) and \(D_t(\Gamma_t)\) denote the dual at time instance \(t\). Since our algorithm modifies only \(\gamma_t\) at time instance \(t\) it is easy to write
the corresponding change in the dual:

\[ D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \rho \gamma_t - \frac{1}{2} \gamma_t^2 k(x_t, x_t) - \gamma_t y_t \sum_{i=1}^{t-1} \gamma_i y_i k(x_i, x_t), \]  

which can be further simplified using the dual connection, (51), to yield

\[ D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \gamma_t (\rho - y_t f_t(x_t)) - \frac{1}{2} \gamma_t^2 k(x_t, x_t). \]  

This is a quadratic form in \( \gamma_t \) with the constraint that \( \gamma_t \in [0, \eta] \). Suppose we are interested in maximizing the increase in the dual objective (see justification below), then we can define \( \hat{\gamma}_t = \frac{\rho - y_t f_t(x_t)}{k(x_t, x_t)} \), and observe that (55) is maximized by setting

\[ \gamma_t = \begin{cases} 
\hat{\gamma}_t & \text{if } \hat{\gamma}_t \in [0, \eta] \\
0 & \text{if } \hat{\gamma}_t < 0 \\
\eta & \text{if } \hat{\gamma}_t > \eta. 
\end{cases} \]  

Recalling that \( \gamma_t = y_t \alpha_t \) and \( y_t \cdot y_t = 1 \) recovers the implicit update (26).

The motivation for maximizing dual progress comes from duality theory. Given \( T \) data points, a batch learning algorithm minimizes \( P_T(f) \). For any \( t \leq T \) we have \( P_T(f) \geq P_t(f) \). As a consequence of weak duality, for any valid \( \Gamma_t \) with \( \gamma_t \in [0, \eta] \), we have \( P_t(f) \geq D_t(\Gamma_t) \). Putting things together this implies that \( P_T(f) \geq P_t(f) \geq D_t(\Gamma_t) \). Therefore, a valid strategy to minimize \( P_T(f) \) is to maximize \( D_t(\Gamma) \). Algorithms which perform an implicit update maximize dual progress at every trial subject to the constraint that they are allowed to modify exactly one dual variable, namely \( \gamma_t \), at the \( t \)-th iteration. For more details on the dual view of online learning we refer the interested reader to Shalev-Shwartz and Singer (2006) who pioneered this concept. Also see Tsochantaridis et al. (2005) who used a similar dual improvement argument to prove convergence rates of their cutting plane algorithm.

### 3.2 Square Loss

In order to derive the Lagrange dual for regularized risk minimization with the square loss, (15), we rewrite the batch objective function, (48), as

\[ \frac{1}{2} \|f\|_{\mathcal{H}}^2 + \eta \sum_{i=0}^{t} \xi_i^+ + \eta \sum_{i=0}^{t} \xi_i^- \]  

s. t. \( \langle f, k(x_i, \cdot) \rangle \geq y_i - \xi_i^+ \) with \( \xi_i^+ \geq 0 \),

\( \langle f, k(x_i, \cdot) \rangle \geq y_i + \xi_i^- \) with \( \xi_i^- \geq 0 \).
Defining $\Xi^+_t := \{\xi_0^+, \ldots, \xi_t^+\}$, $\Xi^-_t := \{\xi_0^-, \ldots, \xi_t^-\}$, $\Gamma^+_t := \{\gamma_0^+, \ldots, \gamma_t^+\}$, $\Gamma^-_t := \{\gamma_0^-, \ldots, \gamma_t^-\}$, $T^+_t := \{\tau_0^+, \ldots, \tau_t^+\}$, and $T^-_t := \{\tau_0^-, \ldots, \tau_t^-\}$ the corresponding Lagrangian is

$$L_{\text{primal}}(f, \Xi^+_t, \Xi^-_t, \Gamma^+_t, \Gamma^-_t, T^+_t, T^-_t) = \frac{1}{2} ||f||_H^2 + \eta \sum_{i=0}^t \xi_i^+ \gamma_i^+ + \eta \sum_{i=0}^t \xi_i^- \gamma_i^-$$

Taking derivatives with respect to the primal variables and setting them to zero yields

$$\frac{\partial L}{\partial f} = 0 \implies f = \sum_{i=0}^t (\gamma_i^+ + \gamma_i^-)k(x_i, \cdot) \quad (59)$$

$$\frac{\partial L}{\partial \xi_i^+} = 0 \implies 2\eta \xi_i^+ = \gamma_i^+ + \tau_i^+ \quad (60)$$

$$\frac{\partial L}{\partial \xi_i^-} = 0 \implies 2\eta \xi_i^- = \gamma_i^- + \tau_i^- \quad (61)$$

Plugging (59), (60), (61) into (58) in order to eliminate the primal variables yields

$$D(\Gamma^+_t, \Gamma^-_t, T^+_t, T^-_t) = - \frac{1}{2} \sum_{i,j} (\gamma_i^+ + \gamma_i^-)(\gamma_j^+ + \gamma_j^-)k(x_i, x_j)$$

$$- \sum_{i=0}^t \frac{(\gamma_i^+ + \tau_i^+)^2}{\eta} - \sum_{i=0}^t \frac{(\gamma_i^- + \tau_i^-)^2}{\eta} + \sum_{i=0}^t (\gamma_i^+ + \gamma_i^-)y_i. \quad (62)$$

Since both $\tau_i^+$ and $\tau_i^-$ are $\geq 0$, it is easy to see that the dual is maximized by setting both $\tau_i^+$ and $\tau_i^-$ to 0. This yields the dual problem

$$D(\Gamma^+_t, \Gamma^-_t) = - \frac{1}{2} \sum_{i,j} (\gamma_i^+ + \gamma_i^-)(\gamma_j^+ + \gamma_j^-)k(x_i, x_j)$$

$$- \sum_{i=0}^t \frac{\gamma_i^2}{\eta} - \sum_{i=0}^t \frac{\gamma_i^-}{\eta} + \sum_{i=0}^t (\gamma_i^+ + \gamma_i^-)y_i. \quad (63)$$
Comparing (59) and (7) shows that \( \alpha_i = \gamma_i^+ + \gamma_i^- \). The change in the dual can now be written as

\[
D_t(\Gamma^+_t, \Gamma^-_t) - D_{t-1}(\Gamma^+_{t-1}, \Gamma^-_{t-1}) = - \frac{1}{2}(\gamma^+_t + \gamma^-_t)^2 k(x_t, x_t) - (\gamma^+_t + \gamma^-_t)(f(x_t) - y_t) \\
- \frac{\gamma^+_t}{\eta} - \frac{\gamma^-_t}{\eta}. 
\]

(64)

Taking gradients with respect to \( \gamma^+_t \) and \( \gamma^-_t \) and setting them to 0 yields

\[
-(\gamma^+_t + \gamma^-_t)k(x_t, x_t) - (f(x_t) - y_t) - \frac{2\gamma^+_t}{\eta} = 0 \quad (65)
\]

\[
-(\gamma^+_t + \gamma^-_t)k(x_t, x_t) - (f(x_t) - y_t) - \frac{2\gamma^-_t}{\eta} = 0. \quad (66)
\]

Adding the above equations

\[
-2(\gamma^+_t + \gamma^-_t)k(x_t, x_t) - 2(f(x_t) - y_t) - \frac{2(\gamma^+_t + \gamma^-_t)}{\eta} = 0, \quad (67)
\]

and rearranging

\[
(\gamma^+_t + \gamma^-_t) = \alpha_t = \frac{y_t - f(x_t)}{\frac{1}{\eta} + k(x_t, x_t)} \quad (68)
\]

recovered the implicit update (18).

### 3.3 Maximum Multiclass Hinge Loss

We will only concentrate on the maximum multiclass hinge loss (39). Derivations for the additive multiclass hinge loss, (29), are very similar and can be found in Appendix B.3. The batch optimization problem (48) for the maximum multiclass hinge loss, (39), can be rewritten as

\[
\min_{f, \xi} \frac{1}{2} ||f||^2_H + \eta \sum_{i=0}^t \xi_i \\
\text{s. t. } \Delta f(y_i, y) \geq \rho - \xi_i, \xi_i \geq 0 \text{ for all } i \text{ and } y \neq y_i. \quad (69)
\]

Note that in contrast to (143) we only have one constraint, \( \xi_i \), per data point. The corresponding Lagrangian is

\[
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} ||f||^2_H + \eta \sum_{i=0}^t \xi_i - \sum_{i=0}^t \tau_i \xi_i \\
- \sum_{i=0}^t \sum_{y \neq y_i} \gamma_{i,y}(\Delta f(y_i, y) - \rho + \xi_i), \quad (70)
\]
where \( \Xi_t := \{\xi_0, \ldots, \xi_t\} \), \( \Gamma_t := \{\gamma_{0,y}, \ldots, \gamma_{t,y}\} \), \( T_t := \{\tau_0, \ldots, \tau_t\} \) with \( \tau_i \geq 0 \) and \( \gamma_{i,y} \geq 0 \) for all \( i \) and all \( y_i \neq y_i \). The KKT conditions (Boyd and Vandenberghe, 2004) assert that \( \gamma_{i,y} > 0 \) if, and only if, \( \Delta f(y_i, y) = \rho - \xi_i \). In other words, \( \gamma_{i,y} \) is non-zero if, and only if, \( y \in \text{argmax}_y, y \neq y_i [\rho - \Delta f(y_i, y')] \) and \( L(x_i, y_i, f) \geq 0 \).

Let \( \Delta k_i(y_i, y) := k((x_i, y_i), \cdot) - k((x_i, y_i), \cdot) \). Taking gradients with respect to the primal variables and setting them to zero yields

\[
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^t \sum_{y \neq y_i} \gamma_{i,y} \Delta k_i(y_i, y) \quad \text{(71)}
\]

\[
\frac{\partial L_t}{\partial \xi_i} = 0 \implies \eta - \sum_{y \neq y_i} \gamma_{i,y} - \tau_i = 0. \quad \text{(72)}
\]

Since both \( \tau_i \) and \( \gamma_{i,y} \) are \( \geq 0 \), it follows from (72) that \( \sum_{y \neq y_i} \gamma_{i,y} \in [0, \eta] \). Plugging in (71) and (72) into (70) yields the dual:

\[
D_t(\Gamma_t) = \rho \sum_{i=0}^t \sum_{y \neq y_i} \gamma_{i,y} - \frac{1}{2} ||\theta_{t+1}||^2
\]

s.t. \( \gamma_{i,y} \geq 0 \) and \( \sum_{y \neq y_i} \gamma_{i,y} \in [0, \eta] \) for all \( i \),

with

\[
\theta_{t+1} = f_{t+1} = \sum_{i=0}^t \sum_{y \neq y_i} \gamma_{i,y} \Delta k_i(y_i, y_i). \quad \text{(74)}
\]

Observe that (73) is exactly the same as the dual of the additive multiclass loss, (147), with one key difference. Here we enforce the constraint \( \sum_{y \neq y_i} \gamma_{i,y} \in [0, \eta] \) while in the additive case we enforce the constraint \( \gamma_{i,y} \in [0, \eta] \) for each \( y \neq y_i \) individually.

By direct comparison with (7) it follows that the coefficients \( \alpha_{i,y} \) of our algorithm, and the Lagrange multipliers \( \gamma_{i,y} \) introduced above are related by

\[
\alpha_{i,y} = \begin{cases} 
-\gamma_{i,y} & \text{if } y \neq y_i \\
\sum_{y' \neq y_i} \gamma_{i,y'} & \text{otherwise.}
\end{cases} \quad \text{(75)}
\]

Let \( k_t(y, y') := k((x_t, y), (x_t, y')) \). After some tedious algebra (not shown here) the change in the dual, \( D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) \) can be written as

\[
\sum_{y \neq y_i} \gamma_{t,y} \left[ (\rho - \Delta f(y_i, y)) - \frac{1}{2} \sum_{y' \neq y_i} \gamma_{t,y'} (k_t(y_i, y) - k_t(y_i, y_i) - k_t(y_i, y_i) + k(y, y')) \right]. \quad \text{(76)}
\]

In order to maximize dual progress we take gradients with respect to \( \gamma_{t,y} \) and set it to 0. This yields for all \( y \neq y_i \)

\[
(\rho - \Delta f(y_i, y)) - \frac{1}{2} \sum_{y' \neq y_i} \gamma_{t,y'} (k_t(y_i, y_i) - k_t(y_i, y_i) - k_t(y_i, y_i) + k(y, y')) = 0. \quad \text{(77)}
\]

1013
Using (75) the above equation can be rewritten as

\[
\rho - \Delta f(y_t, y) - \sum_{y'} \alpha_{t,y'} k_t(y', y_t) + \sum_{y'} \alpha_{t,y'} k_t(y, y') = 0.
\]

(78)

This recovers (41), again showing that implicit updates maximize the progress in the dual at every trial.

In order to show connections to the multiclass MIRA algorithm of Crammer and Singer (2003a) we start with the constraints of the optimization problem (73) and Equation (75). Since \(-\alpha_{t,y} = \gamma_{t,y} \geq 0\), it follows that \(\alpha_{t,y} \leq 0\) for all \(y \neq y_t\). Furthermore, \(\alpha_{t,y_t} = \sum_{y' \neq y_t} \gamma_{t,y'} \in [0, \eta]\). Next we set \(\rho = 0\) and plug in the decomposing kernel \(k_t(y, y') = k(x_t, x_t) \cdot I_{y=y'}\) into (76) to obtain

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \sum_{y \neq y_t} \gamma_{t,y} \left[ -\Delta f(y_t, y) - \frac{1}{2} \left( \sum_{y' \neq y_t} \gamma_{t,y'} + \gamma_{t,y} \right) k(x_t, x_t) \right].
\]

(79)

Plugging in (75) and the definition of \(\Delta f(y_t, y)\) into the above equation and rearranging

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = -\sum_y \alpha_{t,y} f(x_t, y) - \frac{1}{2} \sum_y \alpha_{t,y}^2 k(x_t, x_t).
\]

(80)

In this special case, maximizing dual progress entails minimizing \(\sum_y \alpha_{t,y} \left( f(x_t, y) + \frac{1}{2} \alpha_{t,y} k(x_t, x_t) \right)\) subject to the constraints \(\alpha_{t,y} \leq 0\) for all \(y \neq y_t\), \(\alpha_{t,y_t} \in [0, \eta]\), and \(\sum_y \alpha_{t,y} = 0\). All in all, to maximize dual progress we

\[
\text{minimize } \sum_y \alpha_{t,y} f(x_t, y) + \frac{1}{2} \sum_y \alpha_{t,y}^2 k(x_t, x_t)
\]

s.t. \(\sum_y \alpha_{t,y} = 0\), \(\alpha_{t,y_t} \in [0, \eta]\) and \(\alpha_{t,y} \leq 0\) for all \(y \neq y_t\).

(81)

This essentially recovers the optimization problem solved by the multiclass MIRA algorithm of Crammer and Singer (2003a) (see e.g. Equation (12) of Crammer and Singer (2003a)), thus showing that MIRA is a special case of our updates.

### 4. Mistake Bounds

We only prove mistake bounds for the binary and multiclass hinge loss. Derivations for other losses is very similar and hence omitted. We begin with a technical lemma (Smola et al., 2007):

**Lemma 1** The maximum of \(cx - \frac{1}{2}dx^2\) with \(c, d > 0\) and \(x \in [0, \eta]\) is at least \(\frac{c}{2d} \min(\eta, c/d)\).

**Proof** The unconstrained maximum of the problem is \(x^* = c/d\) with maximum value \(c^2/2d\). If \(c/d \leq \eta\), this is also the constrained minimum. For \(c/d > \eta\) we take \(x^* = \eta\), which yields \(c\eta c - d\eta^2/2\). The latter is at least \(c\eta/2\). Taking the minimum over both maxima proves the claim.

We are now ready to state our mistake bounds for the binary hinge loss.
**Theorem 2** Let \( \{(x_0, y_0), \ldots, (x_T, y_T)\} \) be an arbitrary sequence of observations such that \( x \in \mathcal{X}, y_i \in \{\pm 1\} \) and \( k(x, x) \leq R^2 \) for all \( x \). The number of mistakes made by ILK using the binary hinge loss is at most

\[
\frac{2}{\rho} \max \left( \frac{1}{\eta}, \frac{R^2}{\rho} \right) \inf_f \left( \frac{1}{2} \|f\|_H^2 + \eta \sum_{i=0}^T L(x_i, y_i, f) \right) \tag{82}
\]

**Proof** Let \( P_T(f) \) denote the batch risk functional after \( T \) trials, \( T \) the trials on which the algorithm made a mistake, and \( |T| \) the total number of mistakes. As a consequence of weak duality, and the fact that \( D_1(\gamma) = 0 \), a simple telescoping argument yields

\[
\inf_f P_T(f) \geq \sup_{\Gamma} D_T(\Gamma) \geq D_T(\Gamma_T) \geq \sum_{t \in T} D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}). \tag{83}
\]

By virtue of Lemma 1, the dual progress obtained by maximizing (55) for every \( t \in T \) is

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) \geq \frac{\rho - y_t f_t(x_t)}{2} \min \left( \eta, \frac{\rho - y_t f_t(x_t)}{k(x_t, x_t)} \right). \tag{84}
\]

Furthermore, whenever our algorithm makes a mistake \( \rho - y_t f_t(x_t) \geq \rho \). This, and our assumption \( k(x_t, x_t) \leq R^2 \) allow us to write

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_t) \geq \frac{\rho}{2} \min \left( \eta, \frac{\rho}{2 R^2} \right). \tag{85}
\]

Plugging the above equation and the definition of \( P_T(f) \) into (83) and rearranging terms yields the desired result.

To prove our mistake bound for the multiclass hinge loss we need the following auxiliary lemma. Even though the result implied by this lemma is rather weak (see also Lemma 10 and Corollary 13 of Tsochantaridis et al. (2005)) it is sufficient for our analysis.

**Lemma 3** Let \( D \succeq 0 \) and \( \Gamma \in \mathbb{R}^n \) be a vector with the \( i \)-th component \( \Gamma_i \geq 0 \). The maximum of \( \Gamma^T x - \frac{1}{\rho} x^T D x \) with each \( x_i \in [0, \eta] \) is at least \( \frac{\eta}{2} \min(\eta, \Gamma_i/D_{ii}) \), where \( D_{ii} \) denotes the \( i \)-th diagonal entry of \( D \).

**Proof** Let \( D^* \) denote the maximum of \( \Gamma^T x - \frac{1}{\rho} x^T D x \) with each \( x_i \in [0, \eta] \). Clearly \( D^* \) is larger than the maximum of \( \Gamma_i x_i - \frac{1}{\rho} x_i^2 D_{ii} \) with \( x_i \in [0, \eta] \). \( D_{ii} \geq 0 \) since \( D \succeq 0 \). The result follows from Lemma 3.

We are now ready to state and prove our mistake bounds for the multiclass hinge loss.

**Theorem 4** Let \( \{(x_0, y_0), \ldots, (x_T, y_T)\} \) be an arbitrary sequence of observations such that \( k((x, y), (x, y)) \leq R^2 \) for all \( x \in \mathcal{X} \) and \( y_i \in \mathcal{Y} \). The number of mistakes made by ILK using the multiclass hinge loss is at most

\[
\frac{2}{\rho} \max \left( \frac{1}{\eta}, \frac{R^2}{\rho} \right) \inf_f \left( \frac{1}{2} \|f\|_H^2 + \eta \sum_{i=0}^T L(x_i, y_i, f) \right) \tag{86}
\]
Proof (sketch) Analogous to the proof for the binary hinge loss we bound the progress in the dual. Whenever the algorithm makes a mistake \( \rho - \Delta f(y_t, \bar{y}) \geq \rho \) for some \( \bar{y} \). Recall that at every iteration we maximize (150) subject to the constraint that each \( \gamma_{t,y} \in [0, \eta] \). By virtue of Lemma 3, the dual progress is at least

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) \geq \frac{\rho - \Delta f(y_t, \bar{y})}{2} \min \left( \eta, \frac{\rho - \Delta f(y_t, \bar{y})}{k(x_t, x_t)} \right).
\] (87)

Furthermore, whenever our algorithm makes a mistake we have \( \rho - \Delta f(y_t, \bar{y}) \geq \rho \). This, and our assumption \( k(x_t, x_t) \leq R^2 \) allow us to write

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) \geq \frac{\rho}{2} \min \left( \eta, \frac{\rho}{R^2} \right).
\] (88)

Plugging this and the definition of \( P_T(f) \) into (83) and rearranging terms yields the desired result.

5. Related Work

Even though the idea of an implicit update dates back to Schapire and Warmuth (1996); Kivinen and Warmuth (1997) (see e.g. the discussion following Eq (4.2) of Kivinen and Warmuth (1997) as well as the update rule Eq (4.8)), the term was coined explicitly by Kivinen et al. (2006). In the signal processing community, the algorithm which uses the implicit update for linear regression is widely known as the normalized LMS (least means squared) algorithm (Hassibi et al., 1996). In the machine learning community, identical updates were derived by Kivinen and Warmuth (1997), while the connection to the normalized LMS updates was established by Kivinen et al. (2006). Schapire and Warmuth (1996); Kivinen and Warmuth (1997); Kivinen et al. (2006) and related literature has concentrated on extending the online learning framework to arbitrary Bregman divergences and proving worst case loss bounds. In contrast, our focus is on online learning algorithms in an RKHS, which could potentially be infinite dimensional; consequently we use a simple square norm regularizer, that is, we set \( G(f) = \frac{1}{2} \| f \|_{\mathcal{H}}^2 \), and work directly in the space of coefficients \( \alpha_i, \bar{y} \) (see Eq (7)).

Recently, there has been a lot of interest in online learning with kernels (Kivinen et al., 2004; Crammer et al., 2006; Dekel et al., 2006; Shalev-Shwartz and Singer, 2006). Kivinen et al. (2004) proposed a straightforward extension of gradient descent to RKHS. Their algorithm, NORMA, performs an explicit update, that is, it approximates \( \partial f L(x_t, y_t, f_{t+1}) \approx \partial f L(x_t, y_t, f_t) \) to obtain the following counterpart of (9):

\[
f_{t+1} = f_t - \eta \partial f L(x_t, y_t, f_t).
\] (89)

NORMA works with different loss functions and is able to track a moving target. Our algorithm ILK can be thought of as a logical extension of NORMA, which uses an implicit instead of an explicit update.

In related work, Crammer et al. (2006) propose the passive aggressive update strategy. In a nutshell, their algorithm solves an optimization algorithm in order to perform an
Implicit Online Learning with Kernels

update. This can be viewed as solving a line search for performing implicit updates, as suggested by Kivinen et al. (2006). It therefore comes as no surprise that for the binary hinge loss, when \( \eta_t = 1 \) and \( \lambda = 0 \) our proposed update formula for \( \alpha_t \) (26) reduces to the PA-I algorithm of Crammer et al. (2006). While the loss functions they handle are generally linear (hinge loss and its various generalizations), our updates can handle other non-linear losses such as quadratic or logistic loss. Another closely related work is that based on dual coordinate descent recently proposed by Hsieh et al. (2008) and Keerthi et al. (2008). They motivate their updates by performing optimal coordinate descent in the dual. As discussed in Section 3 our SILK updates can also be motivated analogously from the dual.

Another approach to online learning in an RKHS is due to Dodd et al. (2005). They use the explicit stochastic gradient update, (4), but tune the step size \( \eta_t \) to ensure that \( f_{t+1}(x_t) = y_t \). This leads to \( f_{t+1} = f_t - \frac{1}{||x_t||^2} \partial_{f} L(x_t, y_t, f_t) \), which is identical to the corrective update of Kivinen and Warmuth (1997) (see e.g. Eq (3.2) and the discussion about the GDV algorithm in Kivinen and Warmuth (1997)). While Kivinen and Warmuth (1997) prove worst case loss bounds, Dodd et al. (2005) only demonstrate convergence in the limit.

Various techniques have been proposed for analyzing mistake bounds of online algorithms. We will describe two which are closely related to our work. Kivinen et al. (2004) lower bound the progress towards the optimal hypothesis in order to derive their mistake bounds. Their bounds are applicable even when the underlying target hypothesis is changing with time, that is, in the case of moving targets. On the other hand, Shalev-Shwartz and Singer (2006) prove mistake bounds for online algorithms by lower bounding the progress in the dual. In particular, they derive improved bounds for the passive aggressive update strategy of Crammer et al. (2006). Although not stated explicitly, essentially the same technique of lower bounding the dual improvement was used by Tsochantaridis et al. (2005) to show polynomial time convergence of their algorithm. The main difference however, is that Tsochantaridis et al. (2005) only work with a quadratic objective function, while the framework proposed by Crammer et al. (2006) can handle arbitrary convex functions.

6. Outlook and Discussion

The aim of this research is to unify and put into perspective many different online learning algorithms. Towards this end we presented a general recipe for performing implicit online updates in an RKHS. Specifically, we showed that for many popular loss functions these updates can be computed efficiently. This provides a coherent framework for studying online algorithms, both old and new. The richness of our framework is amply demonstrated by the fact that many new algorithms can be derived, while many existing algorithms fall out as special cases. At the same time, we can prove mistake bounds for our entire family of algorithms by essentially using the same techniques.

The gradient of the convex function \( G \) used for defining the parameter divergence in (1) is called the link function (Kivinen and Warmuth, 1997). There are essentially two means of introducing non-linearity in machine learning algorithms; either by using a kernel function or by using a non-linear link function. Both approaches have their merits and demerits. In the case of kernels one works with a rich parameter space given by an RKHS, but the downside is that one cannot access the individual elements of the parameter vector.
What this means is that we are essentially restricted to using the square norm regularizer, $G(f) = \frac{1}{2} ||f||^2_H$. We chose this route because it leads to closed form solutions for many popular loss functions (essentially the hinge loss and its generalization). The corresponding algorithms using a non-linear link function typically need to perform a line search which is numerically more expensive, and analytically more difficult to analyze.

Acknowledgements

Part of this work was done when SVNV was with NICTA and ANU. A short version of this paper was presented at the 2006 NIPS conference (Cheng et al., 2006). We thank Jin Yu, Xinhua Zhang, and Choon-Hui Teo for carefully proof reading the paper and providing feedback. NICTA is funded by the Australian Government’s Backing Australia’s Ability and the Centre of Excellence programs. This work is also supported by the IST Program of the European Community, under the FP7 Network of Excellence, ICT-216886-NOE.

References


Appendix A. Implicit Updates For Other Loss Functions

A.1 Square Hinge Loss

Perhaps the simplest generalization of the hinge loss, the square hinge loss for binary classification is defined as

\[ L(x_t, y_t, f) := \frac{1}{2}[(\rho - y_t f(x_t))_+]^2 = \frac{1}{2}[(\rho - y_t k(x_t, \cdot))\eta^\top]^2, \]  

(90)
where, as before, $\rho \geq 0$ is the margin parameter. Here we assume a kernel that does not depend on the labels: $k((x_t, y_t), \cdot) = k(x_t, \cdot)$, and set $Y = \{\pm 1\}$. Unlike the hinge loss, the square hinge loss is differentiable everywhere, and its gradient is given by

$$\partial_f L(x_t, y_t, f) = \beta_t k(x_t, \cdot), \text{ where } \beta_t = -y_t(\rho - y_t f(x_t))_+.$$  \hspace{1cm} (91)

To see this write $\partial_f L(x_t, y_t, f) = [(\rho - y_t f(x_t))_+ \partial_f (\rho - y_t f(x_t))_+]$, and observe that if $\rho - y_t f(x_t) \geq 0$, then $\partial_f L(x_t, y_t, f) = -y_t(\rho - y_t f(x_t))k(x_t, \cdot)$, and if $\rho - y_t f(x_t) < 0$, then $\partial_f L(x_t, y_t, f) = 0$. Substituting into (14) and using (9)

$$\alpha_t = \eta y_t (\rho - y_t f_t + 1(x_t))_+ = \eta y_t (\rho - y_t f_t(x_t) - \alpha_t y_t k(x_t, x_t))_+.$$  \hspace{1cm} (92)

By rearranging

$$\alpha_t = \frac{y_t(\rho - y_t f_t(x_t))}{\eta + k(x_t, x_t)}.$$  \hspace{1cm} (93)

if $\rho - y_t f_t(x_t) \geq 0$, and 0 otherwise. All in all

$$\alpha_t = \frac{y_t(\rho - y_t f_t(x_t))_+}{\eta + k(x_t, x_t)}.$$  \hspace{1cm} (94)

Contrast this with the explicit update

$$\alpha_t = \eta y_t (\rho - y_t f_t(x_t))_+.$$  \hspace{1cm} (95)

### A.2 Novelty Detection Loss

The hinge loss for novelty detection is defined as (Schölkopf et al., 2000)

$$L(x_t, f):= (\rho - f(x_t))_+ = (\rho - \langle f, k(x_t, \cdot) \rangle_H)_+ ,$$  \hspace{1cm} (96)

while its subgradient can be written as $\beta_t k(x_t, \cdot)$, where

$$\beta_t \in \begin{cases} 0 & \text{if } f(x_t) > \rho \\ [0, -1] & \text{if } f(x_t) = \rho \\ -1 & \text{if } f(x_t) < \rho. \end{cases}$$  \hspace{1cm} (97)

Let $\hat{\alpha}_t$ denote the optimal estimate of $\alpha_t$ which leads to $\rho - f_{t+1}(x_t) = 0$. Using (9)

$$\rho - (f_t(x_t) + \hat{\alpha}_t k(x_t, x_t)) = 0,$$  \hspace{1cm} (98)

which can be rearranged to

$$\hat{\alpha}_t = \frac{\rho - f_t(x_t)}{k(x_t, x_t)}.$$  \hspace{1cm} (99)
On the other hand, using (97) and (14) we have $\hat{\alpha}_t \in [0, \eta]$. This yields the final clipped update

$$
\alpha_t = \begin{cases} 
\hat{\alpha}_t & \text{if } \hat{\alpha}_t \in [0, \eta] \\
0 & \text{if } \hat{\alpha}_t < 0 \\
\eta & \text{if } \hat{\alpha}_t > \eta.
\end{cases} \quad (100)
$$

The explicit update for this case is simply given by

$$
\alpha_t = \begin{cases} 
0 & \text{if } f(x_t) > \rho \\
[0, \eta] & \text{if } f(x_t) = \rho \\
\eta & \text{if } f(x_t) < \rho.
\end{cases} \quad (101)
$$

### A.3 Category Ranking Hinge Loss

The following generalization of the hinge loss, suitable for category ranking, was introduced by Crammer and Singer (2003b). Here, each instance $x_t$ is associated with a set of labels $\mathcal{Y}_t$, and the loss is measured by

$$
L(x_t, \mathcal{Y}_t, f) := \left( \max_{y \not\in \mathcal{Y}_t, y_t \in \mathcal{Y}_t} \left[ \rho - \Delta f(y_t, y) \right] \right)_+, \quad (102)
$$

where we used our familiar notation $\Delta f(y_t, y) := f(x_t, y_t) - f(x_t, y)$.

Contrast this with the maximum multiclass hinge loss (39), where $\mathcal{Y}_t$ is assumed to contain only one element $y_t$, and hence the max $y_t \in \mathcal{Y}_t$ is redundant.

Define $f^*_t := \max_{y \not\in \mathcal{Y}_t, y_t \in \mathcal{Y}_t} \left[ \rho - \Delta f(y_t, y) \right]$ and $\mathcal{Y}^*_t := \{(y_t, y^*_t) : \argmax_{y \not\in \mathcal{Y}_t, y_t \in \mathcal{Y}_t} \left[ \rho - \Delta f(y_t, y) \right] \}$.

Then the subgradient of the loss can be written as $\sum_y \beta_{t,y} k((x_t, y), \cdot)$, where

$$
\sum_y \beta_{t,y} = 0 \quad \text{and} \\
\begin{align*}
\text{if } f^* < 0 & \text{ then } \beta_{t,y} = 0 \text{ for all } y \\
\text{if } f^* = 0 & \text{ then } \beta_{t,y_t} \in [0, 1], \beta_{t,y} \in [-1, 0], \text{ and } \sum_{y_t} \beta_{t,y_t} \geq -1 \text{ for } (y_t, y^*_t) \in \mathcal{Y}^* \\
\text{if } f^* > 0 & \text{ then } \beta_{t,y_t} \in [0, 1], \beta_{t,y} \in [-1, 0], \text{ and } \sum_{y_t} \beta_{t,y_t} = -1 \text{ for } (y_t, y^*_t) \in \mathcal{Y}^*.
\end{align*} \quad (103)
$$

Note that $\beta_{t,y}$ not mentioned in the above equations are by default set to 0.

The implicit update ensures sets the loss after the update to be zero. This can be achieved analogously to the additive multiclass hinge loss (see (32) and the discussion following it in Section 2.3.1) by setting

$$
\rho - \Delta f_t(y_t, y) - \sum_{y'} \hat{\alpha}_{t,y'} k_t(y', y_t) + \sum_{y'} \hat{\alpha}_{t,y'} k_t(y', y) = 0 \quad (104)
$$

for all $y \not\in \mathcal{Y}_t$ and $y_t \in \mathcal{Y}_t$. Note that we used the convention $k_t(y', y) := k((x_t, y'), (x_t, y))$ in the above equation. In the case of the decomposing kernel, $k_t(y, y') = k(x_t, x_t) \cdot I_{y=y'}$; (104) reduces to

$$
\rho - \Delta f_t(y_t, y) - (\hat{\alpha}_{t,y_t} - \hat{\alpha}_{t,y}) k(x_t, x_t) = 0 \quad (105)
$$
for all $y_t \in \mathcal{Y}_t$ and $y \notin \mathcal{Y}_t$. Enforcing the constraint $\sum_y \hat{\alpha}_{t,y} = 0$ and some tedious algebra (not shown here) yields:

$$\hat{\alpha}_{t,y} = \begin{cases} 
\frac{\nu_y + \rho}{k(x_t, x_t)} & \text{if } y \in \mathcal{Y}_t \\
\nu_y & \text{otherwise,}
\end{cases}$$

where $\nu_y = -\rho |\mathcal{Y}_t| + \sum_{y'} \Delta f_t(y', y) / |\mathcal{Y}| \cdot k(x_t, x_t)$. (106)

Contrast this with (36) for the additive multiclass hinge loss, where $\mathcal{Y}_t$ contains exactly one element namely $y_t$. The clipping schemes that we discussed for (36) are also applicable in this case.

### A.4 Ordinal Regression Loss

In ordinal regression each instance is associated with a rank which is an integer in $\{1, \ldots, k\}$. This can be encoded as a vector $y$ in $\{\pm 1\}^k$ as follows: If the rank of an instance is $r$, set the first $r$ components of $y$ to +1 and the rest of the components to −1. The prediction is a $k$-dimensional vector whose $r$-th component is $f(x_t, r)$. The ordinal regression loss we consider generalizes the binary hinge loss:

$$L(x_t, y_t, f) := \sum_{r=1}^k (\rho - y_{t,r} f(x_t, r))_+, \quad (107)$$

with $y_{t,r}$ denoting the $r$-th component of the encoded vector $y_t$. The subgradient of the loss can be written as $\partial_f L = \sum_r \beta_{t,r} k((x_t, r), \cdot)$, where

$$\beta_{t,r} = \begin{cases} 
0 & \text{if } y_{t,r} f(x_t, r) > \rho \\
[0, -y_{t,r}] & \text{if } y_{t,r} f(x_t, r) = \rho \\
y_{t,r} & \text{if } y_{t,r} f(x_t, r) < \rho.
\end{cases} \quad (108)$$

Let $\hat{\alpha}_{t,r}$ denote the optimal coefficients which lead to zero loss after the update, and let $k_t(r, \cdot) := k((x_t, r), \cdot)$. Using (9) and the definition of the loss

$$\rho - y_{t,r} \left( f_t(x_t, r) + \sum_{r'} \hat{\alpha}_{t,r'} k_t(r, r') \right) = 0 \quad \text{for all } r. \quad (109)$$

To write this out in vector notation define $k$ dimensional vectors $\hat{\alpha}_t$ and $l$ whose $r$-th components are $\hat{\alpha}_{t,r}$ and $\rho - y_{t,r} f_t(x_t, r)$ respectively. Furthermore, define $K_t \in \mathbb{R}^{k \times k}$ to be the kernel matrix with components $k_t(r, r')$. The optimal $\hat{\alpha}_t$ can now be written as

$$\hat{\alpha}_t = K_t^{-1} l, \quad (110)$$

with the actual updates $\alpha_{t,r}$ obtained by clipping each $\hat{\alpha}_{t,r}$ to lie in $[0, -\eta y_{t,r}]$. We now turn our attention to a decomposing kernel $k((x, r), (x', r')) := k(x, x') + I_{r=r'}$, which when combined with (7) shows that $f(x, r)$ decomposes as $f(x, r) = \sum_i \alpha_i r k(x, x) + \sum_i \alpha_i x$. Equivalently, by stacking up $f(x, r)$ into a vector we obtain $f = g(x) e + b$ for some scalar function $g(x)$ and vector $b \in \mathbb{R}^k$. This special case was studied by Crammer and Singer (2005) and Chu and Keerthi (2005). We will show in Section B.5 that the updates of
No-PRank, the algorithm proposed by Crammer and Singer (2005), are obtained by solving (109). For now observe that the decomposing kernel leads to $K_t = k(x_t, x_t) e e^T + I$. By applying the Sherman-Morrison-Woodbury formula (Bernstein, 2005)

$$\hat{\alpha}_t = \left( I - \frac{k(x_t, x_t)}{1 + k(x_t, x_t)} e e^T \right) l. \quad (111)$$

### A.5 Graph Structured Hinge Loss

The graph structured hinge loss exploits structure in the output space to enforce a different margin of separation between the true label $y_t$ and every other label $y \neq y_t$, and is written as (Taskar et al., 2004; Vishwanathan et al., 2006)

$$L(x_t, y_t, f) := \left( \max_{y \neq y_t} [\delta(y_t, y) + f(x_t, y)] - f(x_t, y_t) \right)_+, \quad (112)$$

where $\delta(y_t, y)$ is a function (possibly defined on a graph whose vertices form the output space), which measures the margin of separation to be enforced between labels $y$ and $y_t$. Typically $\delta(y, y) = 0$ and $\delta(y, y) > 0$ if $y \neq y$. To see that this generalizes the multiclass hinge loss simply set $\delta(y, y') = \rho$ if $y \neq y'$ and zero otherwise.

A closely related formulation is due to Tsochantaridis et al. (2005), who rescale the slack variable instead of rescaling the margin of separation. The resultant loss is written as

$$L(x_t, y_t, f) := \left( \max_{y \neq y_t} [\delta(y_t, y) + \delta(y_t, y)(f(x_t, y) - f(x_t, y_t))] \right)_+. \quad (113)$$

The two formulations can be unified via

$$L(x_t, y_t, f) := \left( \max_{y \neq y_t} [\delta(y_t, y) - \delta(y_t, y) \kappa \Delta f(y_t, y)] \right)_+. \quad (114)$$

Setting $\kappa = 0$ yields the Taskar et al. (2004) formulation, while $\kappa = 1$ yields the (Tsochantaridis et al., 2005) formulation. Define

$$k_t(y, y') := k((x_t, y), (x_t, y')), \quad (115)$$

$$f^*_t := \max_{y \neq y_t} [\delta(y_t, y) - \delta(y_t, y) \kappa \Delta f(y_t, y)], \quad (116)$$

$$Y^*_t := \arg\max_{y \neq y_t} [\delta(y_t, y) - \delta(y_t, y) \kappa \Delta f(y_t, y)], \quad (117)$$

and

$$\delta^* := \max_{y \in Y^*} \delta(y_t, y)^\kappa. \quad (118)$$
Now, the subgradient of the loss can be written as \( \sum_y \beta_{t,y} k((x_t, y), \cdot) \), where \( \sum_y \beta_{t,y} = 0 \) and

if \( f^*_t < 0 \) then \( \beta_{t,y} = 0 \) for all \( y \in \mathcal{Y} \)

if \( f^*_t = 0 \) then \( \beta_{t,y} \in [0, \delta(y_t, y^*_t)] \) for \( y^*_t \in \mathcal{Y}_t^* \), \( \beta_{t,y} \in [-\delta^*, 0] \), and other \( \beta_{t,y} = 0 \)

if \( f^*_t > 0 \) then \( \beta_{t,y} \in [0, \delta(y_t, y^*_t)] \) for \( y^*_t \in \mathcal{Y}_t^* \), \( \beta_{t,y} = -\delta^* \), and other \( \beta_{t,y} = 0 \).

All the explicit updates discussed for the multiclass case also apply here. For the sake of brevity, we will not discuss them here. Analogous to (32), an implicit update can be obtained by solving

\[
\delta(y_t, y) - \delta(y_t, y^*_t)^{\kappa} \left( \Delta f_t(y_t, y) - \sum_{y'} \alpha_{t,y'} k_t(y', y_t) + \sum_{y'} \alpha_{t,y'} k_t(y', y) \right) = 0. \tag{122}
\]

But, solving the above set of linear equations is quite challenging, because \( |\mathcal{Y}| \) might be exponentially large. We will therefore focus on a very simple special case: \( \eta = 1 \), \( \delta(y, y') \geq 1 \) if \( y \neq y' \), and, as before, the decomposing kernel \( k_t(y, y') = k(x_t, x_t) \cdot I_{y=y'} \). Furthermore, we will set \( \alpha_t = \alpha_{t,y} = -\alpha_{t,y} \) for some \( y^*_t \in \mathcal{Y}_t^* \) which maximizes \( \delta(y_t, y^*_t) \). Using (9) and setting \( L(x_t, y_t, f_{t+1}) = 0 \) yields for \( y \neq y^*_t \)

\[
\delta(y_t, y) - \delta(y_t, y^*_t)^{\kappa} (\Delta f_t(y_t, y) + \hat{\alpha}_t k(x_t, x_t)) \leq 0, \tag{123}
\]

and for \( y^*_t \)

\[
\delta(y_t, y^*_t) - \delta(y_t, y^*_t)^{\kappa} (\Delta f_t(y_t, y^*_t) + 2\hat{\alpha}_t k(x_t, x_t)) \leq 0. \tag{124}
\]

Both the above constraints can be simplified and rearranged to

\[
\frac{L(x_t, y_t, f_t)}{\delta(y_t, y^*_t)^{\kappa} k(x_t, x_t)} \leq \hat{\alpha}_t \text{ and } \frac{L(x_t, y_t, f_t)}{2 \delta(y_t, y^*_t)^{\kappa} k(x_t, x_t)} \leq \hat{\alpha}_t. \tag{125}
\]

It is easy to see that setting \( \hat{\alpha}_t = \frac{L(x_t, y_t, f_t)}{k(x_t, x_t)} \) satisfies both the above constraints. Now, we simply set \( \alpha_t = \min(1, \hat{\alpha}_t) \).

Appendix B. Dual Updates For Other Loss Functions

B.1 Square Hinge Loss

We now derive the Lagrange dual for the square binary hinge loss (90). We rewrite the problem of minimizing the batch objective function, (48), as a constrained optimization problem:

\[
\min_{f, \xi} \frac{1}{2} ||f||^2_T + \frac{\eta}{2} \sum_i \xi_i^2 \tag{126}
\]

s. t. \( y_i \langle f, k(x_i, \cdot) \rangle \geq \rho - \xi_i, \xi_i \geq 0 \).
The corresponding Lagrangian is

\[
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} ||f||^2 + \frac{\eta}{2} \sum_{i=0}^{t} \xi_i^2 - \sum_{i=0}^{t} \gamma_i (y_i \langle f, k(x_i, \cdot) \rangle) - \rho + \xi_i - \sum_{i=0}^{t} \tau_i \xi_i, \tag{127}
\]

where \(\Xi_t := \{\xi_0, \ldots, \xi_t\}, \Gamma_t := \{\gamma_0, \ldots, \gamma_t\}, T_t := \{\tau_0, \ldots, \tau_t\}\) with \(\tau_i \geq 0\) and \(\gamma_i \geq 0\) for all \(i\). Taking gradients with respect to the primal variables and setting them to zero yields

\[
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^{t} \gamma_i y_i k(x_i, \cdot) \quad \text{and} \quad \tag{128}
\]

\[
\frac{\partial L_t}{\partial \xi_i} = 0 \implies \xi_i = \frac{\gamma_i + \tau_i}{\eta}. \tag{129}
\]

Plugging in (128) and (129) into (127) allows us to eliminate the primal variables \(f\) and \(\xi_i\):

\[
D_t(\Gamma_t, T_t) = \rho \sum_{i=0}^{t} \gamma_i - \frac{1}{2} \sum_{i=0}^{t} \sum_{j=0}^{t} \gamma_i y_i \gamma_j y_j k(x_i, x_j) - \frac{1}{2\eta} \sum_{i=0}^{t} (\gamma_i + \tau_i)^2. \tag{130}
\]

Since \(\tau_i \geq 0\) it is easy to see that the above expression is maximized by setting \(\tau_i = 0\). This yields the Lagrangian dual:

\[
D_t(\Gamma_t, T_t) = \rho \sum_{i=0}^{t} \gamma_i - \frac{1}{2} \sum_{i=0}^{t} \sum_{j=0}^{t} \gamma_i y_i \gamma_j y_j k(x_i, x_j) - \frac{1}{2\eta} \sum_{i=0}^{t} \gamma_i^2. \tag{131}
\]

As before, by direct comparison with (7) it follows that \(\alpha_i = \gamma_i y_i\) or or equivalently \(\gamma_i = y_i \alpha_i\). Now, it is easy to write the change in the dual due to the change in exactly one dual coefficient, \(\gamma_t\), as

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \rho \gamma_t - \frac{1}{2} \gamma_t^2 k(x_t, x_t) - \left(\sum_{i=1}^{t-1} \gamma_i y_i k(x_i, x_t) - \frac{\gamma_t^2}{2\eta}\right), \tag{132}
\]

which can be further simplified using (128) to yield

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \gamma_t (\rho - y_t f_t(x_t)) - \frac{1}{2} \gamma_t^2 \left(k(x_t, x_t) + \frac{1}{\eta}\right). \tag{133}
\]

a quadratic form in \(\gamma_t\). It is easy to verify that dual progress is maximized by setting

\[
\gamma_t = \frac{(\rho - y_t f_t(x_t))_+}{\frac{1}{\eta} + k(x_t, x_t)}. \tag{134}
\]

Recalling that \(\gamma_t = y_t \alpha_t\) recovers the implicit update (94).
B.2 Novelty Detection Loss

We now derive the Lagrange dual for the novelty detection loss (96). Towards this end we first rewrite the batch objective function, (48), as

$$
\frac{1}{2} \| f \|_{H, t}^2 + \eta \sum_{i=0}^{t} \xi_i \tag{135}
$$

s. t. \( \langle f, k(x_i, \cdot) \rangle \geq \rho - \xi_i, \xi_i \geq 0. \tag{136} \)

The corresponding Lagrangian is

$$
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} \| f \|_{H, t}^2 + \eta \sum_{i=0}^{t} \xi_i - \sum_{i=0}^{t} \gamma_i(\langle f, k(x_i, \cdot) \rangle - \rho + \xi_i) - \sum_{i=0}^{t} \tau_i \xi_i, \tag{137} \)

where \( \Xi := \{\xi_0, \ldots, \xi_t\}, \Gamma := \{\gamma_0, \ldots, \gamma_t\}, T := \{\tau_0, \ldots, \tau_t\} \) with \( \tau_i \geq 0 \) and \( \gamma_i \geq 0 \) for all \( i \).

Taking gradients with respect to the primal variables and setting them to zero yields

$$
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^{t} \gamma_i k(x_i, \cdot) \tag{138} \)

$$

$$
\frac{\partial L_t}{\partial \xi_i} = 0 \implies \eta - \gamma_i - \tau_i = 0. \tag{139} \)

Since both \( \tau_i \) and \( \gamma_i \) are \( \geq 0 \), it follows from (139) that \( \gamma_i \in [0, \eta] \). Furthermore, comparing (138) and (7) shows that \( \alpha_i = \gamma_i \). Plugging in (138) and (139) into (137) and defining \( \Gamma_t := \{\gamma_0, \ldots, \gamma_t\} \) yields the dual:

$$
D_t(\Gamma_t) = \rho \sum_{i=0}^{t} \gamma_i - \frac{1}{2} \sum_{i,j} \gamma_i \gamma_j k(x_i, x_j) \text{ s.t. } \gamma_i \in [0, \eta] \text{ for all } i. \tag{140} \)

The change in the dual can now be written as

$$
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \gamma_t(\rho - f_t(x_t)) - \frac{1}{2} \gamma_t^2 k(x_t, x_t), \tag{141} \)

subject to the constraint that \( \gamma_t \in [0, \eta] \). If we define \( \hat{\gamma}_t = \frac{\rho - f_t(x_t)}{k(x_t, x_t)} \) then dual progress is maximized by setting

$$
\gamma_t = \begin{cases} 
\hat{\gamma}_t & \text{if } \hat{\gamma}_t \in [0, \eta] \\
0 & \text{if } \hat{\gamma}_t < 0 \\
\eta & \text{if } \hat{\gamma}_t > \eta,
\end{cases} \tag{142} \)

which recovers the implicit update (100).
B.3 Multiclass Hinge Loss

As before we rewrite the problem of minimizing the batch objective function, \((48)\), with the additive multiclass hinge loss as a constrained optimization problem:

\[
\min_{f, \xi_t} \frac{1}{2} \| f \|_2^2 + \eta \sum_{i=0}^{t} \sum_{y \neq y_i} \xi_{i,y} \tag{143}
\]

s. t. \( \Delta f(y_i, y) \geq \rho - \xi_{i,y}, \ \xi_{i,y} \geq 0 \) for all \( i \) and \( y \neq y_i \).

The corresponding Lagrangian is

\[
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} \| f \|_2^2 + \eta \sum_{i=0}^{t} \sum_{y \neq y_i} \xi_{i,y} - \sum_{i=0}^{t} \sum_{y \neq y_i} \gamma_{i,y} (\Delta f(y_i, y) - \rho + \xi_{i,y}) \tag{144}
\]

\[\text{with } \Xi_t := \{ \xi_{0,y}, \ldots, \xi_{t,y} \}, \ \Gamma_t := \{ \gamma_{0,y}, \ldots, \gamma_{t,y} \}, \ T_t := \{ \tau_{0,y}, \ldots, \tau_{t,y} \} \text{ with } \tau_{i,y} \geq 0 \text{ and } \gamma_{i,y} \geq 0 \text{ for all } i \text{ and all } y \neq y_i. \]

Let \( \Delta k_i(y_i, y) := k((x_i, y_i), \cdot) - k((x_i, y), \cdot) \). Taking gradients with respect to the primal variables and setting them to zero yields

\[
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^{t} \sum_{y \neq y_i} \gamma_{i,y} \Delta k_i(y_i, y) \text{ and } \tag{145}
\]

\[
\frac{\partial L_t}{\partial \xi_{i,y}} = 0 \implies \eta - \gamma_{i,y} - \tau_{i,y} = 0. \tag{146}
\]

Since both \( \tau_{i,y} \) and \( \gamma_{i,y} \) are \( \geq 0 \), it follows from \((146)\) that \( \gamma_{i,y} \in [0, \eta] \). Plugging in \((145)\) and \((146)\) into \((144)\) yields the dual:

\[
D_t(\Gamma_t) = \rho \sum_{i=0}^{t} \sum_{y \neq y_i} \gamma_{i,y} - \frac{1}{2} \| \theta_{t+1} \|_2^2 \text{ s.t. } \gamma_{i,y} \in [0, \eta] \text{ for all } i, \text{ and } y \neq y_i, \tag{147}
\]

with

\[
\theta_{t+1} = f_{t+1} = \sum_{i=0}^{t} \sum_{y \neq y_i} \gamma_{i,y} \Delta k_i(y_i, y). \tag{148}
\]

By direct comparison with \((7)\) it follows that the coefficients \( \alpha_{i,y} \) of our algorithm, and the Lagrange multipliers \( \gamma_{i,y} \) introduced above are related by

\[
\alpha_{i,y} = \begin{cases} 
-\gamma_{i,y} & \text{if } y \neq y_i \\
\sum_{y' \neq y} \gamma_{i,y'} & \text{otherwise.}
\end{cases} \tag{149}
\]

After some tedious algebra (not shown here) the change in the dual can be written as

\[
D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1}) = \sum_{y \neq y_t} \gamma_{t,y} \left[ \rho - \Delta f(y_t, y) \right] - \frac{1}{2} \sum_{y' \neq y_t} \gamma_{t,y'} (k_t(y_t, y_t) - k_t(y, y_t) - k_t(y', y_t) + k_t(y, y')) \tag{150}
\]
In order to maximize dual progress we take gradients with respect to \( \gamma_{t,y} \) and set it to 0. This yields for all \( y \neq y_t \)
\[
(\rho - \Delta f(y_t, y)) - \sum_{y' \neq y_t} \gamma_{t,y'} (k_t(y_t, y_t) - k_t(y_t, y_t) - k_t(y_t, y_t) + k_t(y_t, y')) = 0. \tag{151}
\]

Using (149) the above equation can be rewritten as
\[
(\rho - \Delta f(y_t, y)) - \sum_{y'} \alpha_{t,y'} k_t(y_t, y_t) + \sum_{y'} \alpha_{t,y'} k_t(y_t, y') = 0. \tag{152}
\]

This recovers (32), again showing that implicit updates maximize the progress in the dual at every trial.

**B.4 Category Ranking Hinge Loss**

The derivation for the category ranking hinge loss, (102), is a generalization of the maximum multiclass hinge loss. In this case the batch optimization problem (48) can be rewritten as

\[
\min_{f,\xi_t} \frac{1}{2} ||f||_H^2 + \eta \sum_{i=0}^{t} \xi_i \tag{153}
\]

\[
\text{s. t. } \Delta f(y_i, \bar{y}_i) \geq \rho - \xi_i, \quad \xi_i \geq 0 \text{ for all } i \text{ and } y_i \in \mathcal{Y}_i \text{ and } \bar{y}_i \notin \mathcal{Y}_i.
\]

Note that in contrast to (143) we only have one constraint, \( \xi_i \), per data point. The corresponding Lagrangian is

\[
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} ||f||_H^2 + \eta \sum_{i=0}^{t} \xi_i - \sum_{i=0}^{t} \gamma_{i,y_i} \Delta f(y_i, \bar{y}_i) - \rho + \xi_i, \tag{154}
\]

where \( \Xi_t := \{x_0, \ldots, \xi_t\}, \Gamma_t := \{\gamma_{y_i, \bar{y}_i}\}, T_t := \{\tau_0, \ldots, \tau_t\} \) with \( \tau_i \geq 0 \) and \( \gamma_{y_i, \bar{y}_i} \geq 0 \) for all \( i \) and all \( y_i \in \mathcal{Y}_i \) and \( \bar{y}_i \notin \mathcal{Y}_i \). The KKT conditions (Boyd and Vandenberghe, 2004) assert that \( \gamma_{y_i, \bar{y}_i} > 0 \) if, and only if, \( \Delta f(y_i, \bar{y}_i) = \rho - \xi_i \). In other words, \( \gamma_{y_i, \bar{y}_i} \) is non-zero if, and only if, \( (y_i, \bar{y}_i) \in \arg\max_{y_i' \notin \mathcal{Y}_i} \gamma_{y_i' \bar{y}_i'} \) with \( \rho - \Delta f(y_i', \bar{y}_i') \) and \( L(x_t, \gamma_t, f) \geq 0 \).

Let \( \Delta k_t(x_i, y) := k((x_i, y), \cdot) - k((x_i, y), \cdot) \). Taking gradients with respect to the primal variables and setting them to zero yields

\[
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^{t} \sum_{y_i \in \mathcal{Y}_i, \bar{y}_i \notin \mathcal{Y}_i} \gamma_{y_i, \bar{y}_i} \Delta k_t(y_i, \bar{y}_i) \tag{155}
\]

\[
\frac{\partial L_t}{\partial \xi_i} = 0 \implies \eta - \sum_{y_i \in \mathcal{Y}_i, \bar{y}_i \notin \mathcal{Y}_i} \gamma_{y_i, \bar{y}_i} \tau_i = 0. \tag{156}
\]
Since both $\tau_i$ and $\gamma_{y_i,\bar{y}_i}$ are $\geq 0$, it follows from (156) that $\sum_{y_i \in \mathcal{Y}_i} \sum_{\bar{y}_i \not\in \mathcal{Y}_i} \gamma_{y_i,\bar{y}_i} \in [0, \eta]$. Plugging in (155) and (156) into (154) yields the dual:

$$D_t(\Gamma_t) = \rho \sum_{i=0}^{t} \sum_{y_i \in \mathcal{Y}_i} \sum_{\bar{y}_i \not\in \mathcal{Y}_i} \gamma_{y_i,\bar{y}_i} - \frac{1}{2} \left\| \theta_{t+1} \right\|^2$$

s.t. $\sum_{y_i \in \mathcal{Y}_i} \sum_{\bar{y}_i \not\in \mathcal{Y}_i} \gamma_{y_i,\bar{y}_i} \in [0, \eta]$ for all $i$, and $y \in \mathcal{Y}_i$ and $\bar{y}_i \not\in \mathcal{Y}_i$,

with

$$\theta_{t+1} = f_{t+1} = \sum_{i=0}^{t} \sum_{y_i \in \mathcal{Y}_i} \sum_{\bar{y}_i \not\in \mathcal{Y}_i} \gamma_{y_i,\bar{y}_i} \Delta k_i(y_i, \bar{y}_i).$$

By direct comparison with (7) it follows that the coefficients $\alpha_{i,y}$ of our algorithm, and the Lagrange multipliers $\gamma_{i,y}$ introduced above are related by

$$\alpha_{i,y} = \begin{cases} \sum_{y_i \not\in \mathcal{Y}_i} \gamma_{y_i,\bar{y}_i} & \text{if } y \in \mathcal{Y}_i, \\ -\sum_{y_i \not\in \mathcal{Y}_i} \gamma_{y_i,y} & \text{if } y \not\in \mathcal{Y}_i. \end{cases}$$

Let $k_t(y,y') := k((x_t,y),(x_t,y'))$. After some tedious algebra (not shown here) the change in the dual, $D_t(\Gamma_t) - D_{t-1}(\Gamma_{t-1})$ can be written as

$$\sum_{y_t \in \mathcal{Y}_t} \sum_{y \not\in \mathcal{Y}_t} \gamma_{y_t,y} \left[ (\rho - \Delta f(y_t,y)) - \frac{1}{2} \sum_{y_t' \not\in \mathcal{Y}_t} \sum_{y' \not\in \mathcal{Y}_t} \gamma_{y_t',y'}(k_t(y_t',y_t) - k_t(y_t,y_t) - k_t(y',y_t) + k_t(y,y')) \right].$$

In order to maximize dual progress we take gradients with respect to $\gamma_{y_t,y}$ and set it to 0. This yields for all $y_t \in \mathcal{Y}_t$ and $y \not\in \mathcal{Y}_t$

$$(\rho - \Delta f(y_t,y)) - \frac{1}{2} \sum_{y_t' \not\in \mathcal{Y}_t} \sum_{y' \not\in \mathcal{Y}_t} \gamma_{y_t',y'}(k_t(y_t',y_t) - k_t(y_t,y_t) - k_t(y',y_t) + k_t(y,y')) = 0.$$ (161)

Using (159) the above equation can be rewritten as

$$(\rho - \Delta f(y_t,y)) - \sum_{y'} \alpha_{t,y'} k_t(y',y_t) + \sum_{y'} \alpha_{t,y'} k_t(y,y') = 0.$$ (162)

This recovers (104), again showing that implicit updates maximize the progress in the dual at every trial.

**B.5 Ordinal Regression Loss**

We derive the Lagrange dual for the ordinal regression loss (107). Recall that $\langle f, k((x_i,r), \cdot) \rangle = f(x_i,r)$. Using this we rewrite the batch objective function, (48), as

$$\frac{1}{2} \| f \|_{\ell_t}^2 + \eta \sum_{i=0}^{t} \sum_{r=1}^{k} \xi_{i,r}$$

s. t. $y_{i,r} \langle f, k((x_i,r), \cdot) \rangle \geq \rho - \xi_{i,r}, \xi_{i,r} \geq 0.$ (164)
The corresponding Lagrangian is

\[
L_t(f, \Xi_t, \Gamma_t, T_t) = \frac{1}{2} \|f\|^2 + \eta \sum_{i=0}^{t} \sum_{r=1}^{k} \xi_{i,r} - \sum_{i=0}^{t} \sum_{r=1}^{k} \tau_{i,r} \xi_{i,r}
\]

\[
- \sum_{i=0}^{t} \sum_{r=1}^{k} \gamma_{i,r}(y_{i,r}(f, k((x_i, r), \cdot)) - \rho + \xi_{i,r}),
\]

where \(\Xi_t := \{\xi_{0,1}, \ldots, \xi_{0,k}, \ldots, \xi_{t,1}, \ldots, \xi_{t,k}\}, \Gamma_t := \{\gamma_{0,1}, \ldots, \gamma_{0,k}, \ldots, \gamma_{t,1}, \ldots, \gamma_{t,k}\}, T_t := \{\tau_{0,1}, \ldots, \tau_{0,k}, \ldots, \tau_{t,1}, \ldots, \tau_{t,k}\}\) with \(\tau_{i,r} \geq 0\) and \(\gamma_{i,r} \geq 0\) for all \(i\) and all \(r\). Taking gradients with respect to the primal variables and setting them to zero yields

\[
\frac{\partial L_t}{\partial f} = 0 \implies f = \sum_{i=0}^{t} \sum_{r=1}^{k} \gamma_{i,r} y_{i,r} k((x_i, r), \cdot)
\]

(166)

\[
\frac{\partial L_t}{\partial \xi_{i,r}} = 0 \implies \eta - \gamma_{i,r} - \tau_{i,r} = 0.
\]

(167)

Since both \(\tau_{i,r}\) and \(\gamma_{i,r}\) are \(\geq 0\), it follows from (167) that \(\gamma_{i,r} \in [0, \eta]\). Plugging in (166) and (167) into (165), and letting \(k_t(r, r') := k((x_t, r), (x_t, r'))\) yields the dual:

\[
D_t(\Gamma_t) = \rho \sum_{i=0}^{t} \sum_{r=1}^{k} \gamma_{i,r} - \frac{1}{2} \sum_{i=0}^{t} \sum_{j=0}^{t} \sum_{r=1}^{k} \sum_{r'=1}^{k} \gamma_{i,r} y_{i,r} \gamma_{j,r'} y_{j,r'} k_t(r, r')
\]

\[
\text{s.t. } \gamma_{i,r} \in [0, \eta] \text{ for all } i.
\]

(168)

By comparing (166) with (7) it follows that the coefficients \(\alpha_i\) of our algorithm, and the Lagrange multipliers \(\gamma_i\) introduced above are related via \(\alpha_{i,r} = \gamma_{i,r} y_{i,r}\) or equivalently \(\gamma_{i,r} = y_{i,r} \alpha_{i,r}\). After some tedious algebraic manipulations the change in the dual, \(D_t(\gamma^{t+1}) - D_t(\gamma^t)\), can be written as

\[
\rho \sum_{r=1}^{k} \gamma_{t,r} - \frac{1}{2} \sum_{r,r'} \gamma_{t,r} y_{t,r} \gamma_{t,r'} y_{t,r'} k_t(r, r') - \sum_{r=1}^{k} \gamma_{t,r} y_{t,r} f(x_t, r').
\]

(169)

In order to maximize dual progress we set the gradient with respect to \(\gamma_{t,r}\) to zero. This yields

\[
\rho - y_{t,r} \sum_{r=1}^{k} \gamma_{t,r'} y_{t,r'} k_t(r, r') - y_{t,r} f(x_t, r') = 0 \text{ for all } r.
\]

(170)

We need to satisfy (170) subject to the constraint that \(\gamma_{t,r} \in [0, \eta]\). Noting that \(\alpha_{t,r} = \gamma_{t,r} y_{t,r}\) recovers (109), the implicit update for ordinal regression.

To see the connection to the No-PRank algorithm of Crammer and Singer (2005) we plug in the decomposing kernel \(k_t(r, r') = k((x_t, r), (x_t, r')) := k(x_t, x_t) + I_{r=r'}\) into (169) to obtain:

\[
\sum_{r=1}^{k} \gamma_{t,r} \left(\rho - y_{t,r} f(x_t, r')\right) - \frac{1}{2} k(x_t, x_t) \left(\sum_{r} y_{t,r} \gamma_{t,r}\right)^2 - \frac{1}{2} \sum_{r=1}^{k} \gamma_{t,r}^2.
\]

(171)

Maximizing (171) subject to the constraint \(\gamma_{t,r} \in [0, \eta]\) essentially recovers the dual of the optimization problem solved by No-PRank (Equation (4.5) in Crammer and Singer (2005)).