A Family of Large Margin Linear Classifiers and Its Application in Dynamic Environments

Jianqiang Shen, Thomas G. Dietterich
1148 Kelley Engineering Center, School of EECS
Oregon State University, Corvallis, OR 97331, U.S.A.
{shenj,tgd}@eecs.oregonstate.edu

Abstract

Real-time problems, in which the learning must be fast and the importance of the features might be changing, pose a challenge to machine learning algorithms. To learn robust classifiers in such nonstationary environments, it is essential not to assign too much weight to any single feature. We solve the problems by combining regularization mechanisms with online large margin learning algorithms. We prove bounds on their error and show that removing features with small weights has little influence on the accuracy, suggesting that these methods exhibit feature selection ability. We show that such regularized learning algorithms automatically decrease the influence of the old training instances and focus on the more recent ones. This makes them especially attractive in the dynamic environments. We present experimental results on real datasets to show the merit of our algorithms.

Keywords: Classification, online learning, feature selection, real-time problems, adaptive algorithms.

1 Introduction

In many real-time problems, learning algorithms receive one instance in each iteration. They make a prediction and then receive a feedback regarding the prediction. The efficiency is critical and the computation must be finished within a limited time. Designed for such scenario, online learning algorithms update their hypothesis based on the received feedback. They have limited requirements for CPU time and memory. Their efficiency has made them popular for large-scale learning problems, such as natural language processing. When constructing classifiers over such high-dimensional datasets, we usually face the potential problem of over-fitting. A common strategy for addressing this issue is to first run a feature selection step. However, online learning algorithms often must handle nonstationary data, so standard feature selection methods are not appropriate. Small changes in the data can result in dramatic changes in the optimal feature set, and failure to detect this can severely hurt prediction accuracy.

In this paper, we design efficient large margin learning algorithms by combining regularization mechanisms with online updates. Regularization has been shown to be effective for the batch learning algorithms in learning from the data with many irrelevant features [18, 10]. The regularization penalty shrinks the feature weights towards zero and has the effect of controlling the variance of the learned model. Appropriate regularization can generally reduce over-fitting by trading off a small increase in bias for a large reduction in variance. Compared with feature selection, regularization is a continuous process that shrinks the influence of some features and is more stable, hence is more suitable for non-stationary data. Unlike other weight-shrinking online learning algorithms [11, 12], our algorithms penalize the model complexity without compromising the margin of training instances. This paper investigates both L1 and L2 regularization for online updates. We thoroughly analyze the characteristics of the regularization mechanism in online learning settings.

The regularization penalty drives the weights of many features towards zero. The theoretical analysis shows that ignoring features with small weights has little influence on the prediction accuracy. This feature selection effect can also explain why regularized online learning is usually more accurate, as also shown in the experiments. For real-world online learning problems, the distribution generating the data is usually changing as the time passes. A very discriminative feature can rapidly become less useful or even useless. By avoiding over-weighting a feature, our regularized methods can converge to the right model more quickly when the data changes. We also show that the L2 regularized learning method has another property fitting the dynamic environment – it automatically shrinks the influence of older training instances and pays more attention to the more recent ones.

We begin with an introduction to online learning and a discussion of our motivation. We then derive our regularized large margin algorithms and present their theoretical analysis. We show experimental results on real datasets to illus-

trate the performance of our algorithms. We conclude the paper with a discussion of future work.

2 Online Algorithms and Dynamic Environments

Online learning algorithms typically work on instances in sequence. In iteration t, the algorithm receives instance $\mathbf{x}_t \in \mathbb{R}^n$ and makes a prediction with function f_t. Then it receives y_t, the correct label of $\mathbf{x}_t$, and computes the update condition $\mathcal{C}$. If $\mathcal{C}$ is true, it updates f_{t+1} so that a requirement set $\mathcal{R}$ is satisfied. Our goal here is to minimize the prediction error of a single pass over all instances [1]. Some online learning algorithms have been proposed based on the different design of $\mathcal{C}$ and $\mathcal{R}$ [16, 13, 8, 14, 12, 4, 3]. We focus on the binary class problem where $y_t \in \{+1, -1\}$ and the linear classifier $f_t = \mathbf{w}_t \cdot \mathbf{x}_t$. The results can be easily generalized to multiple-class problems.

The term $y_t(\mathbf{w}_t \cdot \mathbf{x}_t)$ is generally referred to as the *margin*. Enforcing a large margin can often improve the prediction accuracy. In this paper, we focus on the case that we do a *Passive-Aggressive* (*PA*) update [3] when the classifier makes an error. The PA update modifies the learned model subject to two constraints: the correct label should have an appropriate score by a given margin, and the change to the weights should be minimal in order to reduce fluctuations. The PA update sets the new weight vector $\mathbf{w}_{t+1}$ to be the solution to the following constrained optimization problem, $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{2}\left\|\mathbf{w} - \mathbf{w}_t\right\|_2^2$ s.t. $y_t(\mathbf{w} \cdot \mathbf{x}_t) \geq 1$.

Typical online learning algorithms have no limitation on the feature weights. Some features can get quite large weights. This makes the classifier less reliable, especially in dynamic environments where certain features may become uninformative later. For example, let's consider the sentiment problem which predicts whether a product review is positive. When the product *iLearn* first appears in the market, it gets many positive reviews because of its novelty. Hence, the word *iLearn* is a good indicator of positive instances and gets a large weight. As customers get used to the new functions and other competitive products appear, the number of *iLearn*'s negative reviews becomes close to the number of its positive reviews. Consequently, the word *iLearn* is no longer predictive for sentiment. However, since it received a large weight during the early phases, learning algorithms will keep predicting reviews with the word *iLearn* as positive, and they will have difficulty in recovering from such mistakes. To avoid this, it is essential not to assign too much weight to any single feature.

When constructing classifiers over high-dimensional datasets, we usually face the potential problem of overfitting. A common strategy for addressing this issue is to first run a feature selection step. Standard feature selection methods [19] adopt the batch approach and thus are unappropriate for the nonstationary problems faced by online learning. Some feature selection methods have been designed for online scenario [9, 6]. There are two issues with those approaches. First, they assume an adversarial environment and take a worse-case approach. Thus the performance is usually suboptimal in the normal case. Second, they usually have to solve some optimization problems and thus are not feasible for time-critical online problems. In this paper, we solve the over-fitting problem by applying regularization. We show our algorithms have the feature selection ability and can improve over the non-regularized algorithms.

We usually say that an instance is *active* if it triggers an update. Online learning algorithms typically set the initial weight vector to be the zero vector and do updates of the form $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$ where τ_t is the learning rate determined by the learning algorithm. Thus, $\mathbf{w}_t$ is a linear combination of the active instances, and the newer active instances play the same role as the older active instances. We show that for certain kinds of regularized online learning, the updates have the form $\mathbf{w}_{t+1} = \frac{1}{Z_t}\mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, where $Z_t \geq 1$. Thus, the coefficients of those active instances appearing in the earlier phase shrink and have less influence as additional instances are received.

3 Regularized Online Learning of Linear Classifiers

We investigate two kinds of regularization. The first one has an L2 norm penalty in the objective function, and the second one has an explicit norm requirement in the constraint.

3.1 Online Learning with Regularized Objective

Let α be a constant controlling the shrinkage rate, we can shrink the norm of the weight vector towards zero by adding a penalty in the objective function:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{2}\left\|\mathbf{w} - \mathbf{w}_t\right\|_2^2 + \frac{\alpha}{2}\left\|\mathbf{w}\right\|_2^2 \quad \text{s.t. } y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1, \tag{3.1}$$

We denote the hinge loss [10] at iteration t as ℓ_t. This gives us a simple closed-form update:

LEMMA 3.1. *Problem 3.1 has the closed-form solution* $\mathbf{w}_{t+1} = \frac{1}{1+\alpha}(\mathbf{w}_t + \tau_t y_t \mathbf{x}_t)$, *where* $\tau_t = \frac{\ell_t + \alpha}{\|\mathbf{x}_t\|_2^2}$.

Proof. The Lagrangian of the optimization problem in Problem 3.1 is

$$L(\mathbf{w}, \tau) = \frac{1}{2}\left\|\mathbf{w} - \mathbf{w}_t\right\|_2^2 + \frac{\alpha}{2}\left\|\mathbf{w}\right\|_2^2 + \tau(1 - y_t(\mathbf{w}\cdot\mathbf{x}_t)), \tag{3.2}$$

where $\tau \geq 0$ is the Lagrange multiplier. Differentiating this Lagrangian with respect to the elements of $\mathbf{w}$ and setting the partial derivatives to zero gives

$$\mathbf{w} = \frac{1}{1+\alpha}\mathbf{w}_t + \frac{\tau}{1+\alpha} y_t \mathbf{x}_t. \tag{3.3}$$

Replacing $\mathbf{w}$ in Eq 3.2 with Eq 3.3, the Lagrangian becomes

$$L(\tau) = \frac{1}{2}\left\|\frac{\tau}{1+\alpha}y_t\mathbf{x}_t - \frac{\alpha}{1+\alpha}\mathbf{w}_t\right\|_2^2 + \frac{\alpha}{2}\left\|\frac{\tau}{1+\alpha}y_t\mathbf{x}_t + \frac{1}{1+\alpha}\mathbf{w}_t\right\|_2^2 + \tau(1 - \frac{y_t(\mathbf{w}\cdot\mathbf{x}_t)}{1+\alpha} - \frac{\tau\left\|\mathbf{x}_t\right\|_2^2}{1+\alpha}).$$

By setting the derivative of this with respect to τ to zero, we obtain

$$1 - \frac{\tau}{1+\alpha}\left\|\mathbf{x}_t\right\|_2^2 - \frac{y_t(\mathbf{w}\cdot\mathbf{x}_t)}{1+\alpha} = 0$$
$$\Rightarrow \quad \tau = \frac{1 - y_t(\mathbf{w}\cdot\mathbf{x}_t) + \alpha}{\left\|\mathbf{x}_t\right\|_2^2}. \quad \blacksquare$$

Our algorithms try to minimize the penalized weight fluctuations while ensuring a large margin. There have been a few online learning algorithms trying to shrink weights towards zero [11, 12]. The existing work attempts to find a tradeoff between the fitting to the data and the simplicity of the learned model. Their updates do not directly enforce a large margin. Their shrinking mechanisms usually have to sacrifice the fitting to the training data. Experimental results show that our algorithms give much higher accuracy.

Let I_t be the *active set* at iteration t. An instance $\mathbf{x}_i$ is in I_t iff $i < t$ and it triggers an update. Based on Lemma 3.1, the learned decision function at time t can be rewritten as

$$f_t(\mathbf{x}) = \text{sign}\left(\sum_{i\in I_t}\frac{\tau_i y_i}{(1+\alpha)^{|I_t - I_i|}}\mathbf{x}_i\cdot\mathbf{x}\right) \tag{3.4}$$

We can employ the kernel trick by replacing the standard scalar product with a function $K(\cdot,\cdot)$ which satisfies the Mercer conditions. This form is interestingly similar to the Forgetron algorithm [5], an online kernel-based learning algorithm. It does the standard percetron update but controls the number of support vectors by removing the oldest support vector when the size of the support vector set is too large. Since removing a support vector may significantly change the hypothesis, it tries to "shrink" the weight of old support vectors. It multiplies the weights by $\phi_t \in (0,1]$ in each iteration. Our objective-regularized online learning algorithm automatically shrinks the weights of those support vectors and, hence, "forgets" the old support vectors.

Since we are handling nonstationary problems, the best hypothesis at each iteration might be changing. An *optimal algorithm* does the minimal number of updates and does not update its hypothesis unless our algorithm updates the hypothesis. Given a vector sequence $\mathbf{u}_0, .., \mathbf{u}_T \in \mathbb{R}^n$ where $\mathbf{u}_t$ is the hypothesis of the optimal algorithm at iteration t, we let ℓ_t^* denote the loss of $\mathbf{u}_t$ at iteration t. Our regularized algorithm is competitive with the optimal algorithm, as long as the change between two contiguous optimal hypotheses is not extremely dramatic. To see this, we assume that the norm is bounded for each instance $\mathbf{x}_t$, i.e., $\left\|\mathbf{x}_t\right\|_2 \le R$. The following theorem provides an error bound of our algorithm.

THEOREM 3.1. *Assume that there exists an optimal sequence of vectors $\mathbf{u}_0, ..., \mathbf{u}_T \in \mathbb{R}^n$ such that $\left\|\mathbf{u}_t\right\|_2 = D$, $\ell_t^* = 0$ for all t, $\left\|\mathbf{u}_t - \mathbf{u}_{t+1}\right\|_2 \le \mu$ and μ satisfies $g = \frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha} > 0$. Given $\max_t\left\|\mathbf{w}_t\right\|_2 = \beta$, the number of prediction mistakes made by the objective-regularized algorithm is bounded by $m \le \frac{D^2}{g}$.*

Proof. Let $\Delta_t = \left\|\mathbf{w}_t - \mathbf{u}_t\right\|_2^2 - \left\|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\right\|_2^2$. We can prove the bound by lower and upper bounding $\sum_t \Delta_t$. Since $\mathbf{w}_0$ is a zero vector and the norm is non-negative, $\sum_t \Delta_t = \left\|\mathbf{w}_0 - \mathbf{u}_0\right\|_2^2 - \left\|\mathbf{w}_T - \mathbf{u}_T\right\|_2^2 \le \left\|\mathbf{w}_0 - \mathbf{u}_0\right\|_2^2 = D^2$.

Obviously, only if $t \in I_T$, $\Delta_t \neq 0$. We will only consider this case here. Let $\mathbf{w}'_t = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, $\mathbf{w}_{t+1} = \frac{1}{1+\alpha}\mathbf{w}'_t$. Δ_t can be rewritten as

$$\begin{aligned}&(\left\|\mathbf{w}_t - \mathbf{u}_t\right\|_2^2 - \left\|\mathbf{w}'_t - \mathbf{u}_t\right\|_2^2)\\ &+(\left\|\mathbf{w}'_t - \mathbf{u}_t\right\|_2^2 - \left\|\mathbf{w}'_t - \mathbf{u}_{t+1}\right\|_2^2)\\ &+(\left\|\mathbf{w}'_t - \mathbf{u}_{t+1}\right\|_2^2 - \left\|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\right\|_2^2) = \delta_t + \psi_t + \epsilon_t.\end{aligned}$$

We will lower bound δ_t, ψ_t and ϵ_t.

For δ_t, we have

$$\begin{aligned}\delta_t =& -2\tau_t y_t\mathbf{x}_t\cdot(\mathbf{w}_t - \mathbf{u}_t) - \left\|\tau_t y_t\mathbf{x}_t\right\|_2^2\\ \ge& 2\tau_t\ell_t - \tau_t^2\left\|\mathbf{x}_t\right\|_2^2\end{aligned}$$

Plugging the definition of τ_t and considering $\ell_t \ge 1$ get

$$\delta_t \ge \frac{2\ell_t^2 + 2\ell_t\alpha}{\left\|\mathbf{x}_t\right\|_2^2} - \frac{\ell_t^2 + 2\ell_t\alpha + \alpha^2}{\left\|\mathbf{x}_t\right\|_2^2} \ge \frac{1-\alpha^2}{R^2} \tag{3.5}$$

For ψ_t, we have

$$\begin{aligned}\psi_t =& -2\mathbf{w}'_t\cdot(\mathbf{u}_t - \mathbf{u}_{t+1})\\ \ge& -2\left\|\mathbf{w}'_t\right\|_2\left\|\mathbf{u}_t - \mathbf{u}_{t+1}\right\|_2 \ge -2(1+\alpha)\mu\beta\end{aligned} \tag{3.6}$$

For ϵ_t, we have

$$\epsilon_t = (1 - \frac{1}{(1+\alpha)^2})\left\|\mathbf{w}'_t\right\|_2^2 - 2(1 - \frac{1}{1+\alpha})\mathbf{w}'_t\cdot\mathbf{u}_{t+1}$$

Using the fact that $\left\|\mathbf{u} - \mathbf{v}\right\|_2^2 \ge 0$ which equals to $\left\|\mathbf{u}\right\|_2^2 - 2\mathbf{u}\cdot\mathbf{v} \ge -\left\|\mathbf{v}\right\|_2^2$, we get

$$\begin{aligned}&(1 - \frac{1}{(1+\alpha)^2})\left\|\mathbf{w}'_t\right\|_2^2 - 2(1 - \frac{1}{1+\alpha})\mathbf{w}'_t\cdot\mathbf{u}_{t+1}\\ \ge& -\frac{1 - \frac{1}{1+\alpha}}{1 + \frac{1}{1+\alpha}}\left\|\mathbf{u}_{t+1}\right\|_2^2 = -\frac{\alpha D^2}{2+\alpha}\end{aligned} \tag{3.7}$$

Using Eq 3.5, 3.6 and 3.7, we get

$$\sum_{t=1}^{T}\Delta_t \ge m\left(\frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha}\right)$$

Applying $\sum_t \Delta_t \leq D^2$ gives

$$m\left(\frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha}\right) \leq D^2 \tag{3.8}$$

Since $g = \frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha} > 0$, we get the result in the theorem. ∎

Note the error bound increases as μ increases. This suggests that a learning problem with dramatic changing concepts is difficult, even for a regularized online learning algorithm.

3.2 Online Learning with Norm Constraint The objective-regularized algorithm keeps shrinking the weights even when the weights have become quite small. This could hurt prediction accuracy. We can instead only shrink the weights when they get too large by enforcing a norm constraint:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|_2^2 \quad \text{s.t. } y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1 \text{ and } \|\mathbf{w}\|_2 \leq \beta \tag{3.9}$$

This leads to the following simple closed-form update:

LEMMA 3.2. *Problem 3.9 has the closed-form solution* $\mathbf{w}_{t+1} = \frac{1}{Z_t}(\mathbf{w}_t + \tau_t y_t \mathbf{x}_t)$, *where* $Z_t = \max\left\{1, \sqrt{\frac{\|\mathbf{w}_t\|_2^2\|\mathbf{x}_t\|_2^2 - (\mathbf{w}_t\cdot\mathbf{x}_t)^2}{\beta^2\|\mathbf{x}_t\|_2^2 - 1}}\right\}$, $\tau_t = \frac{\ell_t + Z_t - 1}{\|\mathbf{x}_t\|_2^2}$.

This algorithm performs the normal passive-aggressive update until the norm of $\mathbf{w}_t$ becomes large enough. It then shrinks the weights a little bit. The experiments show that this approach is slightly more accurate than the objective-regularized algorithm.

It is easy to show that both the objective-regularized and the L2 norm constrained algorithms are rotationally invariant. Let $\mathcal{M} = \{M \in \mathbb{R}^{n\times n} | MM' = M'M = I, |M| = 1\}$ be the class of rotational matrices, where I is the identity matrix. Given a learning algorithm L, we say it is *rotational invariant* [15] if for any training set S, rotational matrix $M \in \mathcal{M}$, and test example x, we have $L[S, x] = L[MS, Mx]$, where $L[S, x]$ is the predicted label of x resulting from using L to train on S.

LEMMA 3.3. *The learning algorithm solving Problem 3.1 and the algorithm solving Problem 3.9 are rotationally invariant.*

A detailed proof is presented in Appendix. Ng [15] shows that rotationally invariant algorithms can require a large number of training instances to learn a simple model when there are many irrelevant features. In such situations, learning algorithms with L1 regularization usually converge to the right model much faster. Thus, it is worth exploring the following L1 norm constrained learning algorithm:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|_2^2 \quad \text{s.t. } y_t(\mathbf{w}\cdot\mathbf{x}_t) \geq 1 \text{ and } \|\mathbf{w}\|_1 \leq \beta \tag{3.10}$$

This can be transformed into a quadratic programming problem and solved with an off-the-shelf quadratic programming package. Since the L1 norm constraint is still convex, it is guaranteed to find the global optimum. There is a discontinuity in the gradient of L1 with respect to w_i at $w_i = 0$. This tends to force a subset of weights to be exactly zero [18], so that the learned weight vector is sparse.

The following theorem provides an error bound for the online learning algorithms with L2 or L1 norm constraints:

THEOREM 3.2. *Assume that there exists an optimal sequence of vectors* $\mathbf{u}_0, ..., \mathbf{u}_T \in \mathbb{R}^n$ *such that* $\|\mathbf{u}_t\|_2 = D \leq \beta$, $\ell_t^* = 0$ *for all t,* $\|\mathbf{u}_t - \mathbf{u}_{t+1}\|_2 \leq \mu$ *and* μ *satisfies* $\frac{1}{R^2} - 2\mu\beta > 0$, *then the number of errors made by the algorithm with the L2 norm constraint is bounded by* $m \leq \frac{R^2D^2}{1-2\mu\beta R^2}$. *Similarly, if there exists an optimal vector sequence such that* $\|\mathbf{u}_t\|_1 = D \leq \beta$ *and* $\|\mathbf{u}_t - \mathbf{u}_{t+1}\|_\infty \leq \mu$, *then the number of errors made by the algorithm with the L1 norm constraint is bounded by* $m \leq \frac{R^2D^2}{1-2\mu\beta R^2}$.

Proof. We concentrate on the L2 norm case. The proof can be applied to the L1 case similarly. Let $\Delta_t = \|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2$. We can prove the bound by lower and upper bounding $\sum_t \Delta_t$. We know $\sum_t \Delta_t \leq D^2$.

We now lower bound Δ_t. We let $\Delta_t = (\|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_t\|_2^2) + (\|\mathbf{w}_{t+1} - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2) = \gamma_t + \chi_t$.

For χ_t, we have $\chi_t = -2\mathbf{w}_{t+1}\cdot(\mathbf{u}_t - \mathbf{u}_{t+1}) \geq -2\mu\beta$.

It is obvious that $\|\mathbf{u}\|_1 = \sum_i |u_i| \geq |\sum_i u_i|$. Thus

$$\|(\mathbf{w}_{t+1} - \mathbf{w}_t)\mathbf{x}_t\|_1 \geq |y_t(\mathbf{w}_{t+1} - \mathbf{w}_t)\cdot\mathbf{x}_t| \geq 1 \tag{3.11}$$

Applying Holder's inequality, we have

$$\|(\mathbf{w}_{t+1} - \mathbf{w}_t)\mathbf{x}_t\|_1 \leq \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2 \|\mathbf{x}_t\|_2 \tag{3.12}$$

$$\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2 \geq \frac{1}{\|\mathbf{x}_t\|_2} \geq \frac{1}{R} \tag{3.13}$$

Let $\mathcal{P}_S(\mathbf{w})$ denote the projection of $\mathbf{w}$ onto the convex set S, we know for any $\mathbf{u} \in S$, we have $\|\mathbf{w} - \mathbf{u}\|_2^2 - \|\mathcal{P}_S(\mathbf{w}) - \mathbf{u}\|_2^2 \geq \|\mathbf{w} - \mathcal{P}_S(\mathbf{w})\|_2^2$ [2].

We note that $\mathbf{w}_{t+1}$ is a projection of $\mathbf{w}_t$ onto the convex set $S = \{\mathbf{w} \in \mathbb{R}^n | loss(\mathbf{w}, (y_t, \mathbf{x}_t)) = 0 \text{ and } \|\mathbf{w}\|_2 \leq \beta\}$ for the L2 norm constrained algorithm. Since $\mathbf{u}_t \in S$,

$$\gamma_t \geq \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_2^2 \geq \frac{1}{R^2} \tag{3.14}$$

Thus, we have

$$\sum_{i \in I_T} \left(\frac{1}{R^2} - 2\mu\beta \right) \leq D^2 \tag{3.15}$$

$$m \leq \frac{R^2 D^2}{1 - 2\mu\beta R^2} \tag{3.16}$$

We get the result in the theorem. ∎

Since $\|\mathbf{u}_t - \mathbf{u}_{t+1}\|_\infty \leq \|\mathbf{u}_t - \mathbf{u}_{t+1}\|_2$, the error bounds suggest that the L1 algorithm could tolerate more fluctuations of the best hypotheses than the L2 algorithm.

3.3 Feature Selection Effect of Regularized Online Learning Regularized online learning methods force many feature weights to be small. We now show that we can remove these features with small weights without hurting the accuracy too much. Assume that features are sorted in ascending order according to the absolute values of their weights. Suppose we remove the first k features so that $\sum_{i=1}^{k} w_i^2 < \sigma$ and $\sum_{i=1}^{k+1} w_i^2 \geq \sigma$. As long as σ is small, the number of errors is still close to the original regularized method. For the objective-regularized algorithm, we have the following error bound:

THEOREM 3.3. *Assume that there exists an optimal sequence of vectors $\mathbf{u}_0, ..., \mathbf{u}_T \in \mathbb{R}^n$ such that $\|\mathbf{u}_t\|_2 = D$, $\ell_t^* = 0$ for all t, $\|\mathbf{u}_t - \mathbf{u}_{t+1}\|_2 \leq \mu$ and μ satisfies $h = \frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha} - 2\sqrt{\sigma}D > 0$. Given $\max_t \|\mathbf{w}_t\|_2 = \beta$, then the number of prediction mistakes made by the objective-regularized algorithm which removes small weights is bounded by $m \leq \frac{D^2}{h}$.*

Proof. Let $\Delta_t = \|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2$. We can prove the bound by lower and upper bounding $\sum_t \Delta_t$. As the above, we know $\sum_t \Delta_t \leq D^2$.

Let $\mathbf{w}'_t = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$, $\mathbf{w}''_t = \frac{1}{1+\alpha}\mathbf{w}'_t$. We define vector $\mathbf{v}_t \in \mathbb{R}^n$ such as

$$\mathbf{v}_{ti} = \begin{cases} \mathbf{w}''_{t\,i} & \text{if } \sum_{j=1}^{i} \mathbf{w}''^{\,2}_{t\,j} < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{3.17}$$

Then $\mathbf{w}_{t+1} = \mathbf{w}''_t - \mathbf{v}_t$. Δ_t can be rewritten as

$$\begin{aligned} &(\|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}'_t - \mathbf{u}_t\|_2^2) \\ &+(\|\mathbf{w}'_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}'_t - \mathbf{u}_{t+1}\|_2^2) \\ &+(\|\mathbf{w}'_t - \mathbf{u}_{t+1}\|_2^2 - \|\mathbf{w}''_t - \mathbf{u}_{t+1}\|_2^2) \\ &+(\|\mathbf{w}''_t - \mathbf{u}_{t+1}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2) = \delta_t + \psi_t + \epsilon_t + \rho_t. \end{aligned}$$

We have proved the lower bound of δ_t, ψ_t and ϵ_t. For ρ_t, we have

$$\rho_t = 2\mathbf{v}_t \cdot \mathbf{w}''_t - 2\mathbf{v}_t \cdot \mathbf{u}_{t+1} - \|\mathbf{v}_t\|_2^2$$

It is obvious that $\mathbf{v}_t \cdot \mathbf{w}''_t = \|\mathbf{v}_t\|_2^2$, thus we have

$$\rho_t = \|\mathbf{v}_t\|_2^2 - 2\mathbf{v}_t \cdot \mathbf{u}_{t+1} \geq -2\mathbf{v}_t \cdot \mathbf{u}_{t+1} \tag{3.18}$$

Applying Cauchy-Schwarz inequality, we get

$$\rho_t \geq -2\|\mathbf{v}_t\|_2 \|\mathbf{u}_{t+1}\|_2 \geq -2\sqrt{\sigma}D \tag{3.19}$$

Using Eq 3.5, 3.7 and 3.19, we get

$$\sum_{t=0}^{T} \Delta_t \geq m \left(\frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha} - 2\sqrt{\sigma}D \right)$$

Since $\sum_t \Delta_t \leq D^2$, we have

$$m \left(\frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha} - 2\sqrt{\sigma}D \right) \leq D^2$$

Since $h = \frac{1-\alpha^2}{R^2} - 2(1+\alpha)\mu\beta - \frac{\alpha D^2}{2+\alpha} - 2\sqrt{\sigma}D > 0$, we get the result in the theorem. ∎

Similarly, for the algorithms with norm constraints, we have the following error bound:

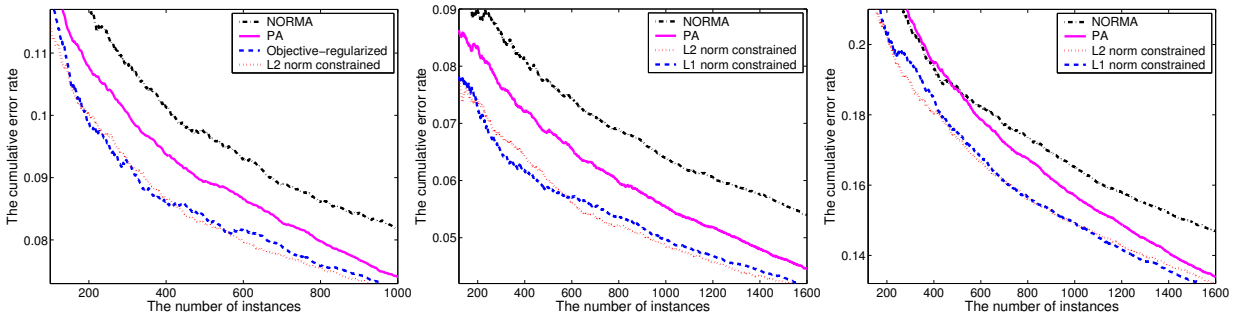
THEOREM 3.4. *Assume that there exists an optimal sequence of vectors $\mathbf{u}_0, ..., \mathbf{u}_T \in \mathbb{R}^n$ such that $\|\mathbf{u}_t\|_2 = D \leq \beta$, $\ell_t^* = 0$ for all t, $\|\mathbf{u}_t - \mathbf{u}_{t+1}\|_2 \leq \mu$ and μ satisfies $q = \frac{1}{R^2} - 2\mu\beta - 2\sqrt{\sigma}D > 0$, then the number of errors made by the algorithm with the L2 norm constraint is bounded by $m \leq \frac{D^2}{q}$. Similarly, if there exists an optimal vector sequence such that $\|\mathbf{u}_t\|_1 = D$ and $\|\mathbf{u}_t - \mathbf{u}_{t+1}\|_\infty \leq \mu$, the number of errors made by the algorithm with the L1 norm constraint is bounded by $m \leq \frac{D^2}{q}$.*

From the theorems, removing those small features slightly increases the error bounds by subtracting $2\sqrt{\sigma}\|\mathbf{u}\|_2$ from the denominator. Given a small σ, ignoring the small features has little influence on the prediction accuracy. Our experimental results show that the accuracy is not hurt even when we remove more than half of the features. This feature selection effect can also explain the superior performance of regularized online learning: our algorithms drive many features towards zero and ignore them in prediction. The algorithms tend to learn a model with a sparse parameter vector when the feature space is large.

4 Experimental Results

Datasets. We tested our algorithms on one image dataset (USPS) and two text datasets (20NewsGroup, SRAA [1]). We removed the header of each text document and converted the remainder into a 0/1 vector without stemming or a stoplist. To make the experiments more realistic and dynamic, we

[1] Available at http://www.cs.umass.edu/~mccallum/data/sraa.tar.gz



(a) USPS (b) 20NewsGroup (c) SRAA

Figure 1: Cumulative error rate of different online methods as a function of the number of instances.

define a non-stationary process for generating the data sequence. In each data set, we choose four classes (P1, P2, N1, N2) and we treat P1 and P2 as positive, N1 and N2 as negative. Then we define four phases for the data sequence. In Phase 1, the probability of class P1 decreases from 0.7 to 0.5 and the probability of N1 increases from 0.3 to 0.5. In Phase 2, the probability of P1 decreases from 0.5 to 0.3, P2 increases from 0 to 0.2, and N1 is fixed at 0.5. In Phase 3, P2 is fixed at 0.5, N1 decrease from 0.2 to 0, and N2 increases from 0.3 to 0.5. In Phase 4, P2 decreases from 0.5 to 0.3 and N2 increases from 0.5 to 0.7. Classes not mentioned have probability 0. For the USPS set, we sample 500 instances in each phase and define P1 to be the digit "3", P2 to be "7", N1 to be "8", and N2 to be "9". For the 20NewsGroups and SRAA set, we sample 800 instances in each phase. For 20NewsGroups, P1 is "Baseball", P2 is "Hockey", N1 is "Auto", and N2 is "Motorcycle". For SRAA, P1 is "RealAutos", P2 is "RealAviation", N1 is "SimulatedAuto", and N2 is "SimulatedAviation". There are 256 features in the sampled USPS data, around 8000 features in the 20NewsGroup data, and around 9000 features in the SRAA data.

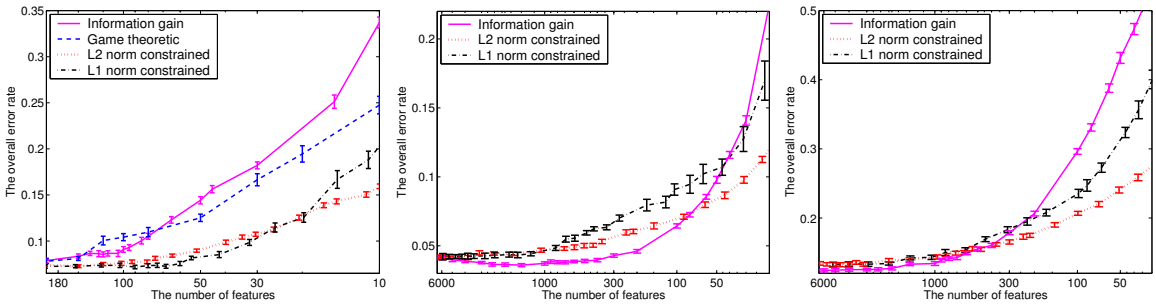
Approach. We applied our algorithms to the generated data sequence as follows. We used the first half of the data to select the regularization parameter and then train the model on the first half. We then measure online performance using the second half of the data. Since the second half corresponds to phases 3 and 4, this approach means that the initial model (learned in phases 1 and 2) performs very badly initially. The results are similar if the regularization parameters are tuned on the second half of the data. All results are averaged over 40 trials.

Comparison with Other Online Learning Methods. We compare our algorithms with the linear NORMA algorithm [12] which is another regularized online learning algorithm and the standard passive-aggressive algorithm [3] which is a state-of-the-art online learning algorithm. The results are shown in Figure 1. The regularized online algorithms outperform PA in all three problems. The difference in accuracy is particularly large at the beginning of the online testing. The difference then decreases as more instances come in. This suggests that the regularized algorithms can adapt well to the changing environment and can converge to a good model more quickly. The method with L2 norm constraint slightly outperforms the objective-regularized method. This may be because the objective-regularized method keeps shrinking the weight vector even when the weights are small, which could lead to over-shrinking. Thus, for the rotationally invariant algorithms we will focus on the method with the L2 norm constraint.

By ensuring a large margin, our algorithms give much higher accuracy than NORMA. Although NORMA can also quickly adjust its hypothesis, it is not as accurate as our algorithms, because NORMA updates the hypothesis based on a stochastic gradient approach and the update does not take into account how large the loss is. The accuracy difference between NORMA and our algorithm is small at the beginning and then increases as more training instances come in. This is especially clear for the SRAA dataset which is a relatively more difficult problem.

The method with L2 norm constraint shows results that are competitive with the method with L1 norm constraint. For the 20NewsGroups problem, which is relatively easy to predict, the L2 constraint even slightly outperforms the L1 norm constraint. A possible reason for the nearly identical performance is that the number of irrelevant features is within an order of magnitude of the number of informative features in these problems. As we will show later, more than half of the features are informative in the USPS problem. For the text problems, there are several times more the meaningless features than the informative features, but the number of informative features is still quite large. It is well known that L1 regularization works best when a small number of features play moderate-sized effects in classification, while L2 regularization is quite competitive when a large number of features play small effects in classification [18].

For the L1 constrained algorithm, we use the current weight vector as the initial value when doing the update by solving the quadratic programming problem. This makes the



(a) USPS (b) 20News (c) SRAA

Figure 2: Performance of feature selection methods as a function of the number of features, with 95% confidence intervals.

update quite efficient. For example, it usually takes less than 10 minutes to process all 3200 20NewsGroup instances on a Linux machine (AMD64 2.6Ghz CPU, 2G memory).

Feature Selection Effect. Next we investigate the feature selection power of our algorithms. We explicitly ignore features with small weights so that the sum of their squared weights roughly equals to the threshold σ. We then count the average number of features per iteration and the overall error rate over the testing sequence.

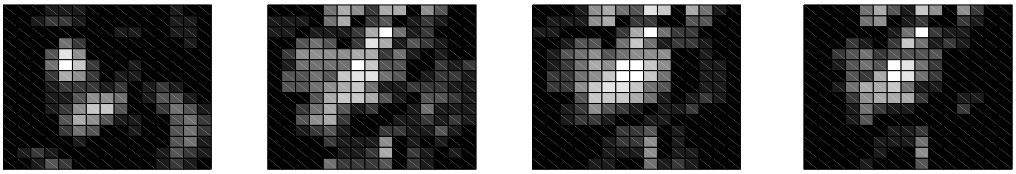
We compared the L1 and L2-norm constrained algorithms with two feature selection methods: information gain [19] and a game-theoretic method. When updating the model, we first perform feature selection and then carry out the model update. Online learning usually faces a dynamic, sometimes even adversarial environment [9, 6]. We can devise a game-theoretic feature selection algorithm by trying to maximize the accuracy in the worst case. We treat feature selection as a two-person game [7]. Let the learner be the row player and the environment be the column player. The game can be thought as this: given an observation $\mathbf{x}$, the row player chooses a feature x_i and probabilistically determines $y[i]$, the label of $\mathbf{x}$ based only on feature x_i. Simultaneously, the column player chooses a feature x_j and probabilistically determines the label $y[j]$. If they predict the same label, then the row player gets reward 1, otherwise 0. We want to design a reward matrix M so that $M(i,j)$ is proportional to the likelihood that $y[i] = y[j]$ when the row player selects feature x_i and the column player selects feature x_j. There could be many choices. One possibility is the information gain of the feature pair $\langle x_i, x_j \rangle$. Let $\mathbf{u}$ be the learner's selection strategy and $\mathbf{v}$ be the environment's strategy, our goal is then to find a strategy that can maximize the reward in the worst case:

$$\max_{\mathbf{u}} \min_{\mathbf{v}} (\mathbf{u} M \mathbf{v}) \quad \text{s.t. } \forall i \ u_i = 0/1, \quad \forall j \ v_j = 0/1, \quad \sum_i u_i = \sum_j v_j = k, \tag{4.20}$$

where k is the number of selected features. We then select those features with $u_i = 1$. In practice, we can relax the integer requirement and replace the constraint $\forall i \ u_i = 0/1, \ \forall j \ v_j = 0/1$ with constraint $\forall i \ 0 \le u_i \le 1, \ \forall j \ 0 \le v_j \le 1$. The problem can be converted into a linear programming problem and be solved efficiently. We then select k features with the largest u_i.

The feature selection results are plotted in Figure 2. In most cases, the regularized methods significantly outperform other feature selection methods, especially for the USPS problem in which most features are informative. Information gain does not realize that the data is changing and still selects features aggressively based on overall mutual information. Its performance is extremely bad when we only select a small number of features. Information gain shows improvement only for the 20NewsGroup problem, which is relatively easy to predict. The game-theoretic method selects better feature sets than information gain. The weight shrinking of the regularized methods is continuous and thus produces smoother results. The L1 norm constrained method is more likely to put more weights on some informative features. Once these features are removed, its performance is severely hurt. Although it shows slightly better performance when we allow many features, it becomes unstable and its accuracy is much worse than the L2 constrained method when the number of selected features is limited. The accuracy of our regularized method is not hurt even when we remove most features. This suggests that our regularized methods have the ability to ignore some features and learn a sparse model.

Adapting to the New Environment. For online learning, its learned model is determined by its *active set*. The regularized methods naturally reduce the influence of the old active instances and put more attention on the more recent ones. We compare the final learned models for the USPS problem. We plot the learned models as follows: each feature corresponds to its position in the image, and the gray scale is proportional to its absolute weight. We apply the information gain criterion to the entire data sequence and get the weight for each feature. We also apply the PA algorithm and the regularized algorithms to the sequence. We compute the absolute weights of the features at the end of the testing. The results are plotted in Figure 3. At the end of Phase 4,



(a) Information gain (b) PA (c) L2 norm constraint (d) L1 norm constraint

Figure 3: The learned models. The lighter color means the larger absolute value of weight.

the system is simply predicting whether a digit is "7" or "9". Hence, the features from the regularized algorithms make much more sense. The L1 constrained method more aggressively puts large weights on a smaller number of features. The PA algorithm still puts large weights on those features that can discriminate "3" and "8".

5 Conclusions And Further Work

We thoroughly analyzed the characteristics of the regularization mechanism in online learning settings and presented three efficient large margin learning algorithms. We theoretically analyzed our algorithms to obtain error bounds. They have some interesting characteristics that make them especially attractive in the dynamic environments: they shrink the weights towards zero and make it easy to adjust the model when the environment is changing. Our regularized methods learn a sparse model by ignoring some features and exhibit good feature selection ability. They naturally shrink the influence of the old active instances and put more weights on the more recent ones. We have successfully applied a variation of our algorithms to the activity recognition [17].

The learned weight vector of our L2 regularized methods is a linear combination of the active instances. It would be interesting to employ the kernel trick here by replacing the standard scalar product with a function satisfying the Mercer conditions and compare our algorithms with the Forgetron algorithm [5]. The Forgetron is an online kernel-based learning algorithm that controls the number of support vectors by removing the oldest support vector when the number of support vectors exceeds a fixed quota. Since removing a support vector may significantly change the hypothesis, it aggressively "shrinks" the weight of old support vectors. Our objective-regularized online algorithm naturally shrinks the weights of those support vectors and we could also control the number of support vectors by removing the oldest one.

Appendix – Detailed Proof of the Theorems

Proof of Lemma 3.2

Proof. The Lagrangian of Problem 3.9 is

$$L(\mathbf{w},\tau) = \frac{1}{2}\left\|\mathbf{w}-\mathbf{w}_t\right\|_2^2 + \tau(1-y_t(\mathbf{w}\cdot\mathbf{x}_t)) + \lambda(\|\mathbf{w}\|_2^2-\beta^2),$$

where $\tau \geq 0$ and $\lambda \geq 0$ are the Lagrange multipliers. Differentiating this Lagrangian with respect to the elements of $\mathbf{w}$ and setting the partial derivative to zero gives

$$\mathbf{w} = \frac{1}{1+2\lambda}(\mathbf{w}_t + \tau y_t \mathbf{x}_t). \tag{5.21}$$

We let $Z = 1+2\lambda$. The KKT conditions require constraint $1-y_t(\mathbf{w}\cdot\mathbf{x}_t) \leq 0$ to be active, which leads to

$$\tau = \frac{Z - y_t(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}_t\|_2^2}. \tag{5.22}$$

The KKT conditions require $\lambda(\|\mathbf{w}\|_2^2-\beta^2)=0$. We discuss two cases here. First, $\lambda = 0$, then we get $\tau = \frac{1-y_t(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}_t\|_2^2}$ from Equation 5.22.

Second, $\|\mathbf{w}\|_2^2-\beta^2=0$. We replace τ with $\frac{Z-y_t(\mathbf{w}\cdot\mathbf{x}_t)}{\|\mathbf{x}_t\|_2^2}$ in it and get

$$\begin{aligned}
&\|\mathbf{w}_t\|_2^2 + \frac{2Zy_t(\mathbf{w}\cdot\mathbf{x}_t) - 2(\mathbf{w}\cdot\mathbf{x}_t)^2}{\|\mathbf{x}_t\|_2^2} \\
&+ \frac{Z^2 - 2Zy_t(\mathbf{w}\cdot\mathbf{x}_t) + (\mathbf{w}\cdot\mathbf{x}_t)^2}{\|\mathbf{x}_t\|_2^2} = \beta^2 Z^2 \\
&(\beta^2\|\mathbf{x}_t\|_2^2 - 1)Z^2 = \|\mathbf{w}_t\|_2^2\|\mathbf{x}_t\|_2^2 - (\mathbf{w}\cdot\mathbf{x}_t)^2.
\end{aligned}$$

This can be possible only if $\beta^2\|\mathbf{x}_t\|_2^2 - 1 \leq \|\mathbf{w}_t\|_2^2\|\mathbf{x}_t\|_2^2 - (\mathbf{w}\cdot\mathbf{x}_t)^2$ since $Z = 1+2\lambda \geq 1$. In this case, we get $Z = \sqrt{\frac{\|\mathbf{w}_t\|_2^2\|\mathbf{x}_t\|_2^2 - (\mathbf{w}_t\cdot\mathbf{x}_t)^2}{\beta^2\|\mathbf{x}_t\|_2^2 - 1}}$. If $\beta^2\|\mathbf{x}_t\|_2^2 - 1 > \|\mathbf{w}_t\|_2^2\|\mathbf{x}_t\|_2^2 - (\mathbf{w}\cdot\mathbf{x}_t)^2$, this corresponds to the first case, $\lambda = 0$. We can easily show that constraint $\|\mathbf{w}\|_2^2 \leq \beta^2$ is always feasible and inactive if it is true:

$$\begin{aligned}
\|\mathbf{w}\|_2^2 &= \frac{\|\mathbf{w}_t\|_2^2\|\mathbf{x}_t\|_2^2 - (\mathbf{w}\cdot\mathbf{x}_t)^2 + 1}{\|\mathbf{x}_t\|_2^2} \\
&< \frac{\beta^2\|\mathbf{x}_t\|_2^2 - 1 + 1}{\|\mathbf{x}_t\|_2^2} \\
&= \beta^2.
\end{aligned}$$

Since both our objective function and constraint functions are convex, we know our solution is globally optimal. Combining the above two cases, we conclude the proof. ∎

Proof of Lemma 3.3

Proof. We focus on the Problem 3.1. The proof can be similarly applied to Problem 3.9. We show that if L trained with S outputs $\mathbf{w}$ then L trained with MS outputs the weight vector $M\mathbf{w}$ by inducting on the size of S.

Let $L[S]$ denote the weight vector returned by L trained with S.

When $|S| = 0$, both weight vectors are zero vectors. Thus, the claim is true and the score functions will be the same since they always return 0.

Assume when $|S| = k$, $L[MS] = ML[S]$. Now, consider $|S| = k+1$. Let $S = S' \cup \{x_{k+1}\}$ and $MS = (MS') \cup \{Mx_{k+1}\}$. We know $L[MS'] = ML[S']$, since the size of S' is k. Since we are using the linear product as the score function, for L trained with MS' we have the score for Mx_{k+1}: $L[MS'] \cdot (Mx_{k+1}) = (ML[S']) \cdot (Mx_{k+1}) = L[S'] (MM) x_{k+1} = L[S'] \cdot x_{k+1}$. Thus the prediction of Mx_{k+1} given by L with MS' is the same with the prediction of x_{k+1} given by L with S'.

If there is no need to update weight, we obviously have $L[MS] = ML[S]$. If we need to update the weight, the update will be $L[MS] = \frac{1}{1+\alpha}(L[MS'] + \tau'_t y_t (M\mathbf{x}_{k+1}))$, where $\tau'_t = \frac{\ell'+\alpha}{\|M\mathbf{x}_{k+1}\|^2} = \frac{\ell+\alpha}{\|\mathbf{x}_{k+1}\|^2} = \tau_t$. Thus $L[MS] = M\frac{1}{1+\alpha}(L[S'] + \tau_t y_t (\mathbf{x}_{k+1})) = ML[S]$.

As a summary, $L[MS] = ML[S]$ given any dataset. Since we are using the linear product as the score function, we always have $L[S, x] = L[MS, Mx]$. ∎

Proof of Theorem 3.4

Proof. Let $\Delta_t = \|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2$. We can prove the bound by lower and upper bounding $\sum_t \Delta_t$. As the above, we know $\sum_t \Delta_t \leq D^2$.

Obviously, only if $t \in I_T$, $\Delta_t \neq 0$. We will only consider this case here. Let $\mathbf{w}'_t$ be the solution of Problem 3.9. We define vector $\mathbf{v}_t \in \mathbb{R}^n$ such as

$$\mathbf{v}_{ti} = \begin{cases} \mathbf{w}'_{ti} & \text{if } \sum_{j=1}^{i} \mathbf{w}'^2_{tj} < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{5.23}$$

Then $\mathbf{w}_{t+1} = \mathbf{w}'_t - \mathbf{v}_t$. Δ_t can be rewritten as

$$(\|\mathbf{w}_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}'_t - \mathbf{u}_t\|_2^2) \tag{5.24}$$

$$+(\|\mathbf{w}'_t - \mathbf{u}_t\|_2^2 - \|\mathbf{w}'_t - \mathbf{u}_{t+1}\|_2^2) \tag{5.25}$$

$$+(\|\mathbf{w}'_t - \mathbf{u}_{t+1}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2) \tag{5.26}$$

We have shown $\chi_t + \gamma_t \geq \frac{1}{R^2} - 2\mu\beta$ and $\|\mathbf{w}'_t - \mathbf{u}_{t+1}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{u}_{t+1}\|_2^2 \geq -2\sqrt{\sigma}D$. Thus,

$$\Delta_t \geq \frac{1}{R^2} - 2\mu\beta - 2\sqrt{\sigma}D \tag{5.27}$$

Since we know $\frac{1}{R^2} - 2\mu\beta - 2\sqrt{\sigma}D > 0$, we get the result in the theorem. ∎

References

[1] V. R. Carvalho and W. W. Cohen. Single-pass online learning: performance, voting schemes and online feature selection. In *Proc. of KDD-06*, pages 548–553, 2006.

[2] Y. Censor and S. Zenios. *Parallel Optimization: Theory, Algorithms, and Applications*. Oxford University Press, New York, NY, USA, 1997.

[3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[4] K. Crammer and Y. Singer. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951 – 991, 2003.

[5] O. Dekel, S. Shalev-Shwartz, and Y. Singer. The Forgetron: A kernel-based perceptron on a fixed budget. In *Advances in NIPS 18*, pages 259–266. 2006.

[6] O. Dekel and O. Shamir. Learning to classify with missing and corrupted features. In *Proc. of ICML-08*, pages 216–223, 2008.

[7] Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proc. of COLT-96*, pages 325–332, 1996.

[8] C. Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2002.

[9] A. Globerson and S. Roweis. Nightmare at test time: robust learning by feature deletion. In *Proc. of ICML-06*, pages 353–360, 2006.

[10] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[11] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2):151–178, 1998.

[12] J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165– 2176, 2004.

[13] J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[14] Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1–3):361–387, 2002.

[15] A. Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. In *Proc. of ICML-04*, pages 78–85, 2004.

[16] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Neurocomputing: foundations of research*, pages 89–114, 1988.

[17] J. Shen, J. Irvine, X. Bao, M. Goodman, S. Kolibaba, A. Tran, F. Carl, B. Kirschner, S. Stumpf, and T. Dietterich. Detecting and correcting user activity switches: Algorithms and interfaces. In *Proc. of IUI-09*, 2009.

[18] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B*, 58(1):267–288, 1996.

[19] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of ICML-97*, pages 412–420, 1997.