Appearing in the Proceedings of European Conference on Machine Learning (ECML) 2010

Online Knowledge-Based Support Vector Machines

Gautam Kunapuli*, Kristin P. Bennett†, Amina Shabbeer†, Richard Maclin‡ and Jude Shavlik*

*University of Wisconsin-Madison, †Rensselaer Polytechnic Insitute, and ‡University of Minnesota, Duluth

Abstract. Prior knowledge, in the form of simple advice rules, can greatly speed up convergence in learning algorithms. Online learning methods predict the label of the current point and then receive the correct label (and learn from that information). The goal of this work is to update the hypothesis taking into account not just the label feedback, but also the prior knowledge, in the form of soft polyhedral advice, so as to make increasingly accurate predictions on subsequent examples. Advice helps speed up and bias learning so that generalization can be obtained with less data. Our passive-aggressive approach updates the hypothesis using a hybrid loss that takes into account the margins of both the hypothesis and the advice on the current point. Encouraging computational results and loss bounds are provided.

1 Introduction

We propose a novel online learning method that incorporates advice into passive-aggressive algorithms, which we call the *Adviceptron*. Learning with advice and other forms of inductive transfer have been shown to improve machine learning by introducing bias and reducing the number of samples required. Prior work has shown that advice is an important and easy way to introduce domain knowledge into learning; this includes work on knowledge-based neural networks [15] and prior knowledge via kernels [12]. More specifically, for SVMs [16], knowledge can be incorporated in three ways [13]: by modifying the data, the kernel or the underlying optimization problem. While we focus on the last approach, we direct readers to a recent survey [9] on prior knowledge in SVMs.

Despite advances to date, research has not addressed how to incorporate advice into incremental SVM algorithms from either a theoretical or computational perspective. In this work, we leverage the strengths of Knowledge-Based Support Vector Machines (KBSVMs) [6] to effectively incorporate advice into the passive-aggressive framework introduced by Crammer et al., [4]. Our work explores the various difficulties and challenges in incorporating prior knowledge into online approaches and serves as a template to extending these techniques to other online algorithms. Consequently, we present an appealing framework for generalizing KBSVM-type formulations to online algorithms with simple, closed-form weight-update formulas and known convergence properties.

We focus on the binary classification problem and demonstrate the incorporation of advice that leads to a new algorithm called the passive-aggressive *Adviceptron*. In the Adviceptron, as in KBSVMs, advice is specified for convex, polyhedral regions in the input space of data. As shown in Fung et al., [6], advice takes the form of (a set of) simple, possibly conjunctive, implicative rules. Advice can be specified about *every* potential data point in the input space which satisfies certain advice constraints, such as the rule

$$(\texttt{feature}_7 \geq 5) \wedge (\texttt{feature}_{12} \geq 4) \Rightarrow (\texttt{class} = +1),$$

which states that the class should be $+1$ when $\texttt{feature}_7$ is at least 5 *and* $\texttt{feature}_{12}$ is at least 4. Advice can be specified for individual features as above and for linear combinations of features, while the conjunction of multiple rules allows more complex advice sets. However, just as label information of data can be noisy, the advice specification can be noisy as well. The purpose of advice is twofold: first, it should help the learner reach a good solution with *fewer training data points*, and second, advice should help the learner reach a potentially *better solution* (in terms of generalization to future examples) than might have been possible learning from data alone.

We wish to study the generalization of KBSVMs to the online case within the well-known framework of passive-aggressive algorithms (PAAs, [4]). Given a loss function, the algorithm is *passive* whenever the loss is zero, i.e., the data point at the current round t is correctly classified. If misclassified, the algorithm updates the weight vector ($\mathbf{w}^t$) *aggressively*, such that the loss is minimized over the new weights ($\mathbf{w}^{t+1}$). The update rule that achieves this is derived as the optimal solution to a constrained optimization problem comprising two terms: a loss function, and a *proximal term* that requires $\mathbf{w}^{t+1}$ to be as close as possible to $\mathbf{w}^t$. There are several advantages of PAAs: first, they readily apply to standard SVM loss functions used for batch learning. Second, it is possible to derive closed-form solutions and consequently, simple update rules. Third, it is possible to formally derive relative loss bounds where the loss suffered by the algorithm is compared to the loss suffered by some arbitrary, fixed hypothesis.

We evaluate the performance of the Adviceptron on two real-world tasks: diabetes diagnosis, and *Mycobacterium tuberculosis* complex (MTBC) isolate classification into major genetic lineages based on DNA fingerprints. The latter task is an essential part of tuberculosis (TB) tracking, control, and research by health care organizations worldwide [7]. MTBC is the causative agent of tuberculosis, which remains one of the leading causes of disease and morbidity worldwide. Strains of MTBC have been shown to vary in their infectivity, transmission characteristics, immunogenicity, virulence, and host associations depending on their phylogeographic lineage [7]. MTBC biomarkers or DNA fingerprints are routinely collected as part of molecular epidemiological surveillance of TB. Classification of strains of MTBC into genetic lineages can help implement suitable control measures. Currently, the United States Centers for Disease Control and Prevention (CDC) routinely collect DNA fingerprints for all culture positive TB patients in the United States. Dr. Lauren Cowan at the CDC has developed

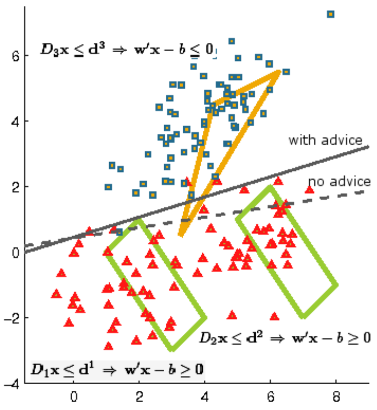


Fig. 1: Knowledge-based classifier which separates data and advice sets. If the advice sets were perfectly separable we have hard advice, (3). If subregions of advice sets are misclassified (analogous to subsets of training data being misclassified), we soften the advice as in (4). We revisit this data set in our experiments.

expert rules for classification which synthesize "visual rules" widely used in the tuberculosis research and control community [1, 3]. These rules form the basis of the expert advice employed by us for the TB task. Our numerical experiments demonstrate the Adviceptron can speed up learning better solutions by exploiting this advice. In addition to experimental validation, we also derive regret bounds for the Adviceptron.

We introduce some notation before we begin. Scalars are denoted by lowercase letters (e.g., y, τ), all vectors by lowercase bold letters (e.g., $\mathbf{x}$, $\boldsymbol{\eta}$) and matrices by uppercase letters (e.g., D). Inner products between two vectors are denoted $\mathbf{x}'\mathbf{z}$. For a vector $\mathbf{p}$, the notation $\mathbf{p}_+$ denotes the componentwise plus-function, $\max(p_j, 0)$ and $\mathbf{p}_\star$ denotes the componentwise step function. The step function is defined for a scalar component p_j as $(p_j)_\star = 1$ if $p_j > 0$ and $(p_j)_\star = 0$ otherwise.

2 Knowledge-Based SVMs

We now review knowledge-based SVMs [6]. Like classical SVMs, they learn a linear classifier ($\mathbf{w}'\mathbf{x} = b$) given data $(\mathbf{x}^t, y_t)_{t=1}^T$ with $\mathbf{x}^t \in \mathbb{R}^n$ and labels $y_t \in \{\pm 1\}$. In addition, they are also given *prior knowledge* specified as follows: *all points* that satisfy constraints of the polyhedral set $D_1\mathbf{x} \le \mathbf{d}^1$ belong to class $+1$. That is, the advice specifies that $\forall \mathbf{x}$, $D_1\mathbf{x} \le \mathbf{d}^1 \;\Rightarrow\; \mathbf{w}'\mathbf{x} - b \ge 0$. Advice can also be given about the other class using a second set of constraints: $\forall \mathbf{x}$, $D_2\mathbf{x} \le \mathbf{d}^2 \;\Rightarrow\; \mathbf{w}'\mathbf{x} - b \le 0$. Combining both cases using *advice labels*, $z = \pm 1$, advice is given by specifying $(D, \mathbf{d}, z)$, which denotes the implication

$$D\mathbf{x} \le \mathbf{d} \;\Rightarrow\; z(\mathbf{w}'\mathbf{x} - b) \ge 0. \tag{1}$$

We assume that m advice sets $(D_i, \mathbf{d}^i, z_i)_{i=1}^m$ are given in addition to the data, and if the i-th advice set has k_i constraints, we have $D_i \in \mathbb{R}^{k_i \times n}$, $\mathbf{d}^i \in \mathbb{R}^{k_i}$ and

$z_i = \{\pm 1\}$. Figure 1 provides an example of a simple two-dimensional learning problem with both data and polyhedral advice.

Note that due to the implicative nature of the advice, it says *nothing* about points that do not satisfy $D\mathbf{x} \leq \mathbf{d}$. Also note that the notion of margin can easily be introduced by requiring that $D\mathbf{x} \leq \mathbf{d} \Rightarrow z(\mathbf{w}'\mathbf{x} - b) \geq \gamma$, i.e., that the advice sets (and all the points contained in them) be separated by a margin of γ analogous to the notion of margin for individual data points. Advice in implication form cannot be incorporated into an SVM directly; this is done by exploiting theorems of the alternative [11]. Observing that $p \Rightarrow q$ is equivalent to $\neg p \vee q$, we require that the latter be true; this is same as requiring that the negation $(p \wedge \neg q)$ be false or that the system of equations

$$\{D\mathbf{x} - \mathbf{d}\,\tau \leq 0,\ z\mathbf{w}'\mathbf{x} - zb\,\tau < 0,\ -\tau < 0\} \text{ has no solution } (\mathbf{x}, \tau). \tag{2}$$

The variable τ is introduced to bring the system to nonhomogeneous form. Using the nonhomogeneous Farkas theorem of the alternative [11] it can be shown that (2) is equivalent to

$$\{D'\mathbf{u} + z\mathbf{w} = 0,\ -\mathbf{d}'\mathbf{u} - zb \geq 0,\ \mathbf{u} \geq 0\} \text{ has a solution } \mathbf{u}. \tag{3}$$

The set of (hard) constraints above incorporates the advice specified by *a single rule/advice set*. As there are m advice sets, each of the m rules is added as the equivalent set of constraints of the form (3). When these are incorporated into a standard SVM, the formulation becomes a hard-KBSVM; the formulation is hard because the advice is assumed to be linearly separable, that is, always feasible. Just as in the case of data, linear separability is a very limiting assumption and can be relaxed by introducing slack variables ($\boldsymbol{\eta}^i$ and ζ_i) to soften the constraints (3). If $\mathcal{P}$ and $\mathcal{L}$ are some convex regularization and loss functions respectively, the full soft-advice KBSVM is

$$\begin{array}{ll} \underset{(\boldsymbol{\xi}, \mathbf{u}^i, \boldsymbol{\eta}^i, \zeta_i) \geq 0, \mathbf{w}, b}{\text{minimize}} & \mathcal{P}(\mathbf{w}) + \lambda\, \mathcal{L}_{data}(\boldsymbol{\xi}) + \mu \displaystyle\sum_{i=1}^{m} \mathcal{L}_{advice}(\boldsymbol{\eta}^i,\, \zeta_i) \\ \text{subject to} & Y(X\mathbf{w} - b\mathbf{e}) + \boldsymbol{\xi} \geq \mathbf{e}, \\ & D_i'\mathbf{u}^i + z_i\mathbf{w} + \boldsymbol{\eta}^i = 0, \\ & -\mathbf{d}^{i\prime}\mathbf{u}^i - z_i b + \zeta_i \geq 1,\ i = 1, \ldots, m, \end{array} \tag{4}$$

where X is the $T \times n$ set of data points to be classified with labels $\mathbf{y} \in \{\pm 1\}^T$, $Y = diag(\mathbf{y})$ and $\mathbf{e}$ is a vector of ones of the appropriate dimension. The variables $\boldsymbol{\xi}$ are the standard slack variables that allow for soft-margin classification of the data. There are two regularization parameters $\lambda,\, \mu \geq 0$, which tradeoff the data and advice errors with the regularization.

While converting the advice from implication to constraints, we introduced new variables for each advice set: the *advice vectors* $\mathbf{u}^i \geq 0$. The advice vectors perform the same role as the dual multipliers α in the classical SVM. Recall that points with non-zero α's are the *support vectors* which additively contribute to $\mathbf{w}$. Here, for *each* advice set, the constraints of the set which have non-zero $\mathbf{u}^i$s are called *support constraints.*

3 Passive-Aggressive Algorithms with Advice

We are interested in an online version of (4) where the algorithm is given T labeled points $(\mathbf{x}^t, y_t)_{t=1}^T$ sequentially and required to update the model hypothesis, $\mathbf{w}^t$, as well as the advice vectors, $\mathbf{u}^{i,t}$, at every iteration. The batch formulation (4) can be extended to an online passive-aggressive formulation by introducing proximal terms for the advice variables, $\mathbf{u}^i$:

$$\begin{aligned}
&\underset{\xi,\mathbf{u}^i,\boldsymbol{\eta}^i,\zeta_i,\mathbf{w}}{\arg\min}\ \frac{1}{2}\|\mathbf{w}-\mathbf{w}^t\|^2 + \frac{1}{2}\sum_{i=1}^m \|\mathbf{u}^i - \mathbf{u}^{i,t}\|^2 + \frac{\lambda}{2}\xi^2 + \frac{\mu}{2}\sum_{i=1}^m \left(\|\boldsymbol{\eta}^i\|^2 + \zeta_i^2\right) \\
&\text{subject to } y_t\mathbf{w}'\mathbf{x}^t - 1 + \xi \geq 0, \\
&\qquad \left.\begin{aligned} D_i'\mathbf{u}^i + z_i\mathbf{w} + \boldsymbol{\eta}^i &= 0 \\ -\mathbf{d}^{i\prime}\mathbf{u}^i - 1 + \zeta_i &\geq 0 \\ \mathbf{u}^i &\geq 0 \end{aligned}\right\}\ i = 1,\dots,m.
\end{aligned} \tag{5}$$

Notice that while L_1 regularization and losses were used in the batch version [6], we use the corresponding L_2 counterparts in (5). This allows us to derive passive-aggressive closed-form solutions. We address this illustrative and effective special case, and leave the general case of dynamic online learning of advice and weight vectors for general losses as future work.

Directly deriving the closed-form solutions for (5) is impossible owing to the fact that satisfying the many inequality constraints at optimality is a combinatorial problem which can only be solved iteratively. To circumvent this, we adopt a two-step strategy when the algorithm receives a new data point $(\mathbf{x}^t, y_t)$: first, fix the advice vectors $\mathbf{u}^{i,t}$ in (5) and use these to update the weight vector $\mathbf{w}^{t+1}$, and second, fix the newly updated weight vector in (5) to update the advice vectors and obtain $\mathbf{u}^{i,t+1}$, $i = 1,\dots,m$. While many decompositions of this problem are possible, the one considered above is arguably the most intuitive and leads to an interpretable solution and also has good regret minimizing properties. In the following subsections, we derive each step of this approach and in the section following, analyze the regret behavior of this algorithm.

3.1 Updating w Using Fixed Advice Vectors $\mathbf{u}^{i,t}$

At step t $(= 1,\dots,T)$, the algorithm receives a new data point $(\mathbf{x}^t, y_t)$. The hypothesis from the previous step is $\mathbf{w}^t$, with corresponding advice vectors $\mathbf{u}^{i,t}$, $i = 1,\dots,m$, one for each of the m advice sets. In order to update $\mathbf{w}^t$ based on the advice, we can simplify the formulation (5) by fixing the advice variables $\mathbf{u}^i = \mathbf{u}^{i,t}$. This gives a fixed-advice online passive-aggressive step, where the variables ζ_i drop out of the formulation (5), as do the constraints that involve those variables. We can now solve the following problem (the corresponding Lagrange multipliers for each constraint are indicated in parentheses):

$$\begin{aligned}
\mathbf{w}^{t+1} \ = \ &\underset{\mathbf{w},\xi,\boldsymbol{\eta}^i}{\text{minimize}}\ \frac{1}{2}\|\mathbf{w}-\mathbf{w}^t\|_2^2 + \frac{\lambda}{2}\xi^2 + \frac{\mu}{2}\sum_{i=1}^m \|\boldsymbol{\eta}^i\|_2^2 \\
&\text{subject to } y_t\mathbf{w}'\mathbf{x}^t - 1 + \xi \geq 0, \qquad (\alpha) \\
&\qquad D_i'\mathbf{u}^i + z_i\mathbf{w} + \boldsymbol{\eta}^i = 0,\ i = 1,\dots,m.\ (\boldsymbol{\beta}^i)
\end{aligned} \tag{6}$$

In (6), $D_i'\mathbf{u}^i$ is the classification hypothesis according to the i-th knowledge set. Multiplying $D_i'\mathbf{u}^i$ by the label z_i, the labeled i-th hypothesis is denoted $\mathbf{r}^i = -z_i D_i'\mathbf{u}^i$. We refer to the $\mathbf{r}^i$s as the *advice-estimates of the hypothesis* because they represent each advice set as a point in hypothesis space. We will see later that the next step when we update the advice using the fixed hypothesis can be viewed as representing the hypothesis-estimate of the advice as a point in that advice set. The effect of the advice on $\mathbf{w}$ is clearly through the equality constraints of (6) which force $\mathbf{w}$ at each round to be as close to *each of* the advice-estimates as possible by aggressively minimizing the error, $\boldsymbol{\eta}^i$. Moreover, Theorem 1 proves that the optimal solution to (6) can be computed in closed-form and that this solution requires only the centroid of the advice estimates, $\mathbf{r} = (1/m)\sum_{i=1}^m \mathbf{r}^i$. For fixed advice, the centroid or *average advice* vector $\mathbf{r}$, provides a compact and sufficient summary of the advice.

Update Rule 1 (Computing $\mathbf{w}^{t+1}$ from $\mathbf{u}^{i,t}$) *For λ, $\mu > 0$, and given advice vectors $\mathbf{u}^{i,t} \geq 0$, let $\mathbf{r}^t = 1/m\sum_{i=1}^m \mathbf{r}^{i,t} = -1/m\sum_{i=1}^m z_i D_i'\mathbf{u}^{i,t}$, with $\nu = 1/(1+m\mu)$. Then, the optimal solution of (6) which also gives the closed-form update rule is given by*

$$\mathbf{w}^{t+1} \;=\; \mathbf{w}^t + \alpha_t y_t \mathbf{x}^t + \sum_{i=1}^m z_i \boldsymbol{\beta}^{i,t} \;=\; \nu\,(\mathbf{w}^t + \alpha_t y_t \mathbf{x}^t) + (1-\nu)\,\mathbf{r}^t,$$

$$\alpha_t = \frac{\Big(\,1 - \nu\, y_t {\mathbf{w}^t}'\mathbf{x}^t - (1-\nu)\, y_t {\mathbf{r}^t}'\mathbf{x}^t\,\Big)_+}{\dfrac{1}{\lambda} + \nu\|\mathbf{x}^t\|^2}, \quad \frac{z_i\boldsymbol{\beta}^{i,t}}{\mu} = \mathbf{r}^{i,t} - \frac{\mathbf{w}^t + \alpha_{t\star}\,\lambda\, y_t \mathbf{x}^t + m\mu\,\mathbf{r}^t}{\dfrac{1}{\nu} + \alpha_{t\star}\,\lambda\|\mathbf{x}^t\|^2}. \tag{7}$$

The numerator of α_t is the *combined loss function*,

$$\ell_t = \max\Big(\,1 - \nu\, y_t {\mathbf{w}^t}'\mathbf{x}^t - (1-\nu)\, y_t {\mathbf{r}^t}'\mathbf{x}^t,\ 0\Big), \tag{8}$$

which gives us the condition upon which the update is implemented. This is exactly the hinge loss function where the margin is computed by a convex combination of the current hypothesis $\mathbf{w}^t$ and the current advice-estimate of the hypothesis $\mathbf{r}^t$. Note that for any choice of $\mu > 0$, the value of $\nu \in (0, 1]$ with $\nu \to 0$ as $\mu \to \infty$. Thus, ℓ_t is simply the hinge loss function applied to a *convex combination* of the margin of the hypothesis, $\mathbf{w}^t$ from the *current iteration* and the margin of the *average advice-estimate*, $\mathbf{r}^t$. Furthermore, if there is no advice, $m = 0$ and $\nu = 1$, and the updates above become exactly identical to online passive-aggressive algorithms for support vector classification [4]. Also, it is possible to eliminate the variables $\boldsymbol{\beta}^i$ from the expressions (7) to give a very simple update rule that depends only on α_t and $\mathbf{r}^t$:

$$\mathbf{w}^{t+1} = \nu(\mathbf{w}^t + \alpha_t y_t \mathbf{x}^t) + (1-\nu)\mathbf{r}^t. \tag{9}$$

This update rule is a convex combination of the current iterate updated by the data, $\mathbf{x}^t$ and the advice, $\mathbf{r}^t$.

3.2 Updating $\mathbf{u}^{i,t}$ Using the Fixed Hypothesis $\mathbf{w}^{t+1}$

When $\mathbf{w}$ is fixed to $\mathbf{w}^{t+1}$, the master problem breaks up into m smaller subproblems, the solution of each one yielding updates to each of the $\mathbf{u}^i$ for the i-th advice set. The i-th subproblem (with the corresponding Lagrange multipliers) is shown below:

$$\begin{aligned}\mathbf{u}^{i,t+1} = \arg\min_{\mathbf{u}^i,\boldsymbol{\eta},\zeta}\ & \frac{1}{2}\,\|\mathbf{u}^i - \mathbf{u}^{i,t}\|^2 + \frac{\mu}{2}\left(\|\boldsymbol{\eta}^i\|_2^2 + \zeta_i^2\right) \\ \text{subject to } & D_i'\mathbf{u}^i + z_i\mathbf{w}^t + \boldsymbol{\eta}^i = 0, && (\boldsymbol{\beta}^i) \\ & -\mathbf{d}^{i\prime}\mathbf{u}^i - 1 + \zeta_i \geq 0, && (\gamma_i) \\ & \mathbf{u}^i \geq 0. && (\boldsymbol{\tau}^i)\end{aligned} \tag{10}$$

The first-order gradient conditions can be obtained from the Lagrangian:

$$\mathbf{u}^i = \mathbf{u}^{i,t} + D_i\boldsymbol{\beta}^i - \mathbf{d}^i\gamma_i + \boldsymbol{\tau}^i, \qquad \boldsymbol{\eta}^i = \frac{\boldsymbol{\beta}^i}{\mu}, \qquad \zeta_i = \frac{\gamma_i}{\mu}. \tag{11}$$

The complicating constraints in the above formulation are the cone constraints $\mathbf{u}^i \geq 0$. If these constraints are dropped, it is possible to derive a closed-form intermediate solution, $\tilde{\mathbf{u}}^i \in \mathbb{R}^{k_i}$. Then, observing that $\boldsymbol{\tau}^i \geq 0$, we can compute the final update by projecting the intermediate solution onto $\mathbf{u}^i \geq 0$.

$$\mathbf{u}^{i,t+1} = \left(\mathbf{u}^{i,t} + D_i\boldsymbol{\beta}^i - \mathbf{d}^i\gamma_i\right)_+. \tag{12}$$

When the constraints $\mathbf{u}^i \geq 0$ are dropped from (10), the resulting problem can be solved (analogous to the derivation of the update step for $\mathbf{w}^{t+1}$) to give a closed-form solution which depends on the dual variables: $\tilde{\mathbf{u}}^{i,t+1} = \mathbf{u}^{i,t} + D_i\boldsymbol{\beta}^i - \mathbf{d}^i\zeta_i$. This solution is then projected into the positive orthant by applying the plus function: $\mathbf{u}^{i,t+1} = \tilde{\mathbf{u}}_+^{i,t+1}$. This leads to the advice updates, which need to applied to each advice vector $\mathbf{u}^{i,t}$, $i = 1, \ldots, m$ individually.

Update Rule 2 (Computing $\mathbf{u}^{i,t+1}$ from $\mathbf{w}^{t+1}$) *For $\mu > 0$, and given the current hypothesis $\mathbf{w}^{t+1}$, for each advice set, $i = 1, \ldots, m$, the update rule is given by*

$$\mathbf{u}^{i,t+1} = \left(\mathbf{u}^{i,t} + D_i\boldsymbol{\beta}^i - \mathbf{d}^i\gamma_i\right)_+, \quad \begin{bmatrix}\boldsymbol{\beta}^i \\ \gamma_i\end{bmatrix} = H_i^{-1}\mathbf{g}^i,$$

$$H_{i,t} = \begin{bmatrix} -(D_i'D_i + \frac{1}{\mu}I_n) & D_i'\mathbf{d}^i \\ \mathbf{d}^{i\prime}D_i & -(\mathbf{d}^{i\prime}\mathbf{d}^i + \frac{1}{\mu})\end{bmatrix}, \quad \mathbf{g}^{i,t} = \begin{bmatrix} D_i'\mathbf{u}^{i,t} + z_i\mathbf{w}^t \\ -\mathbf{d}^{i\prime}\mathbf{u}^{i,t} - 1\end{bmatrix}, \tag{13}$$

with the untruncated solution being the optimal solution to (10) without the cone constraints $\mathbf{u}^i \geq 0$.

Recall that, when updating the hypothesis $\mathbf{w}^t$ using new data points $\mathbf{x}^t$ and the fixed advice (i.e., $\mathbf{u}^{i,t}$ is fixed), each advice set contributes an estimate of the hypothesis ($\mathbf{r}^{t^i} = -z_iD_i'\mathbf{u}^{i,t}$) to the update. We termed the latter the *advice-estimate* of the hypothesis. Here, given that when there is an update, $\beta \neq 0$,

Algorithm 1 The Passive-Aggressive Adviceptron Algorithm

```
1: input: data $(\mathbf{x}^t, y_t)_{t=1}^T$, advice sets $(D_i, \mathbf{d}^i, z_i)_{i=1}^m$, parameters $\lambda$, $\mu > 0$
2: initialize: $\mathbf{u}^{i,1} = 0$, $\mathbf{w}^1 = 0$
3: let $\nu = 1/(1 + m\mu)$
4: for $(\mathbf{x}^t, y_t)$ do
5:     predict label $\hat{y}_t = \text{sign}(\mathbf{w}^{t\prime}\mathbf{x}^t)$
6:     receive correct label $y_t$
7:     suffer loss $\ell_t = 1 - \nu y_t \mathbf{w}^{t\prime}\mathbf{x}^t - (1-\nu) y_t \mathbf{r}^{t\prime}\mathbf{x}^t$ where $\mathbf{r}^t = -\frac{1}{m}\sum_{i=1}^m z_i D_i' \mathbf{u}^{i,t}$
8:     update hypothesis using $\mathbf{u}^{i,t}$, as defined in Update 1
           $\alpha = \ell_t / (\frac{1}{\lambda} + \nu \|\mathbf{x}^t\|^2), \quad \mathbf{w}^{t+1} = \nu\,(\,\mathbf{w}^t + \alpha\, y_t \mathbf{x}^t\,) + (1-\nu)\,\mathbf{r}^t$
9:     update advice using $\mathbf{w}^{t+1}$, $(H_i, \mathbf{g}^i)$ as defined in Update 2
           $(\boldsymbol{\beta}^i, \gamma_i) = H_i^{-1}\mathbf{g}^i, \quad \mathbf{u}^{i,t+1} = \left(\mathbf{u}^{i,t} + D_i\boldsymbol{\beta}^i - \mathbf{d}^i\gamma_i\right)_+$
10: end for
```

$\gamma_i > 0$, we denote $\mathbf{s}^i = \boldsymbol{\beta}^i/\gamma_i$ as the *hypothesis-estimate of the advice*. Since $\boldsymbol{\beta}^i$ and γ_i depend on $\mathbf{w}^t$, we can reinterpret the update rule (12) as

$$\mathbf{u}^{i,t+1} = \left(\mathbf{u}^{i,t} + \gamma_i (D_i \mathbf{s}^i - \mathbf{d}^i)\,\right)_+ . \tag{14}$$

Thus, the advice variables are refined using the hypothesis-estimate of that advice set according to the current $\mathbf{w}^t$; here the update is the *error or the amount of violation of the constraint $D_i\mathbf{x} \le \mathbf{d}^i$ by an ideal data point, $\mathbf{s}^i$ estimated by the current hypothesis, $\mathbf{w}^t$*. Note that the error is scaled by a factor γ_i.

Now, update Rules 1 and 2 can be combined together to yield the full passive-aggressive *Adviceptron* (Algorithm 1).

4 Analysis

In this section, we analyze the behavior of the passive-aggressive adviceptron by studying its regret behavior and loss-minimizing properties. Returning to (4) for a moment, we note that there are three loss functions in the objective, each one penalizing a slack variable in each of the three constraints. We formalize the definition of the three loss functions here. The loss function $L_\xi(\mathbf{w};\, \mathbf{x}^t,\, y_t)$ measures the error of the labeled data point $(\mathbf{x}^t,\, y_t)$ from the hyperplane $\mathbf{w}$; $L_\eta(\mathbf{w}^t, \mathbf{u}^i;\, D_i,\, z_i)$ and $L_\zeta(\mathbf{u}^i;\, \mathbf{d}^i,\, z_i)$ cumulatively measure how well $\mathbf{w}$ satisfies the advice constraints $(D_i,\, \mathbf{d}^i,\, z_i)$. In deriving Updates 1 and 2, we used the following loss functions:

$$L_\xi(\mathbf{w};\, \mathbf{x}^t,\, y_t) = (1 - y_t \mathbf{w}'\mathbf{x}^t)_+, \tag{15}$$

$$L_\eta(\mathbf{w}, \mathbf{u};\, D_i,\, z_i) = \|D_i'\mathbf{u} + z_i\mathbf{w}\|^2, \tag{16}$$

$$L_\zeta(\mathbf{u};\, \mathbf{d}^i,\, z_i) = (1 + \mathbf{d}^{i\prime}\mathbf{u})_+. \tag{17}$$

Also, in the context of (4), $\mathcal{L}_{data} = \frac{1}{2}L_\xi^2$ and $\mathcal{L}_{advice} = \frac{1}{2}(L_\eta + L_\zeta^2)$. Note that in the definitions of the loss functions, the arguments after the semi-colon are the data and advice, which are fixed.

Lemma 1. *At round t, if we define the updated advice vector* before projection *for the i-th advice set as $\tilde{\mathbf{u}}^i$, the following hold for all $\mathbf{w} \in \mathbb{R}^n$:*

1. $\tilde{\mathbf{u}}^i = \mathbf{u}^{i,t} - \mu \nabla_{\mathbf{u}^i} \mathcal{L}_{advice}(\mathbf{u}^{i,t})$,
2. $\|\nabla_{\mathbf{u}^i} \mathcal{L}_{advice}(\mathbf{u}^{i,t})\|^2 \leq \Big(\|D_i\|^2 L_\eta(\mathbf{u}^{i,t}, \mathbf{w}) \;+\; \|\mathbf{d}^i\|^2 L_\zeta^2(\mathbf{u}^{i,t}) \Big)$.

The first inequality above can be derived from the definition of the loss functions and the first-order conditions (11). The second inequality follows from the first condition using convexity: $\|\nabla_{\mathbf{u}^i}\mathcal{L}_{advice}(\mathbf{u}^{i,t})\|^2 = \|D_i \boldsymbol{\eta}^i - \mathbf{d}^i \gamma_i\|^2 = \|D_i(D_i'\mathbf{u}^{i,t} + z_i \mathbf{w}^{t+1}) + \mathbf{d}^i(\mathbf{d}^{i\prime}\mathbf{u}^{i,t} + 1)\|^2 \leq \|D_i(D_i'\mathbf{u}^{i,t} + z_i\mathbf{w}^{t+1})\|^2 + \|\mathbf{d}^i(\mathbf{d}^{i\prime}\mathbf{u}^{i,t} + 1)\|^2$. The inequality follows by applying $\|A\mathbf{x}\| \leq \|A\|\|\mathbf{x}\|$. We now state additional lemmas that can be used to derive the final regret bound. The proofs are in the appendix.

Lemma 2. *Consider the rules given in Update 1, with $\mathbf{w}^1 = 0$ and λ, $\mu > 0$. For all $\mathbf{w}^* \in \mathbb{R}^n$ we have*

$$\begin{aligned} &\|\mathbf{w}^* - \mathbf{w}^{t+1}\|^2 - \|\mathbf{w}^* - \mathbf{w}^t\|^2 \leq \\ &\qquad \nu\lambda L_\xi(\mathbf{w}^*)^2 - \frac{\nu\lambda}{1+\nu\lambda X^2} L_\xi(\widehat{\mathbf{w}}^t)^2 + (1-\nu)\|\mathbf{w}^* - \mathbf{r}^t\|^2. \end{aligned}$$

where $\widehat{\mathbf{w}}^t = \nu\mathbf{w}^t + (1-\nu)\mathbf{r}^t$, the combined hypothesis that determines if there is an update, $\nu = 1/(1+m\mu)$, and we assume that $\|\mathbf{x}^t\|^2 \leq X^2, \forall t = 1, \ldots, T$.

Lemma 3. *Consider the rules given in Update 2, for the i-th advice set with $\mathbf{u}^{i,1} = 0$, and $\mu > 0$. For all $\mathbf{u}^* \in \mathbb{R}_+^{k_i}$, we have*

$$\begin{aligned} \|\mathbf{u}^* - \mathbf{u}^{i,t+1}\|^2 - \|\mathbf{u}^* - \mathbf{u}^{i,t}\|^2 \;\leq\; & \mu L_\eta(\mathbf{u}^*, \mathbf{w}^t) + \mu L_\zeta(\mathbf{u}^*)^2 \\ & -\mu\,\big((1-\mu\Delta^2) L_\eta(\mathbf{u}^{i,t}, \mathbf{w}^t) + (1-\mu\delta^2) L_\zeta(\mathbf{u}^{i,t})^2 \,\big). \end{aligned}$$

where we assume that $\|D_i\|^2 \leq \Delta^2$ and $\|\mathbf{d}^i\|^2 \leq \delta^2$.

Lemma 4. *At round t, given the current hypothesis and advice vectors $\mathbf{w}^t$ and $\mathbf{u}^{i,t}$, for any $\mathbf{w}^* \in \mathbb{R}^n$ and $\mathbf{u}^{i,*} \in \mathbb{R}_+^{k_i}$, $i = 1, \ldots, m$, we have*

$$\|\mathbf{w}^* - \mathbf{r}^t\|^2 \leq \frac{1}{m}\sum_{i=1}^m L_\eta(\mathbf{w}^*, \mathbf{u}^{i,t}) = \frac{1}{m}\sum_{i=1}^m \|\mathbf{w}^* - \mathbf{r}^{i,t}\|^2$$

The overall loss suffered over one round $t = 1, \ldots, T$ is defined as follows:

$$R(\mathbf{w}, \mathbf{u}; c_1, c_2, c_3) = \left(c_1 L_\xi(\mathbf{w})^2 + \sum_{i=1}^m (c_2 L_\eta(\mathbf{w}, \mathbf{u}) + c_3 L_\zeta(\mathbf{u})^2) \right).$$

This is identical to the loss functions defined in the batch version of KBSVMs (4) and its online counterpart (10). The Adviceptron was derived such that it minimizes the latter. The lemmas are used to prove the following regret bound for the Adviceptron[1].

Theorem 1. *Let* $\mathfrak{S} = \{(\mathbf{x}^t,\, y_t)\}_{t=1}^T$ *be a sequence of examples with* $(\mathbf{x}^t,\, y_t) \in \mathbb{R}^n \times \{\pm 1\}$*, and* $\|\mathbf{x}^t\|_2 \leq X \quad \forall t$*. Let* $\mathfrak{A} = \{(D_i,\, \mathbf{d}^i,\, z_i)\}_{i=1}^m$ *be* m *advice sets with* $\|D_i\|_2 \leq \varDelta$ *and* $\|\mathbf{d}^i\|_2 \leq \delta$*. Then the following holds for all* $\mathbf{w}^* \in \mathbb{R}^n$ *and* $\mathbf{u}^i \in \mathbb{R}_+^{k_i}$*:*

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T} R\left(\mathbf{w}^t, \mathbf{u}^t; \frac{\lambda}{1+\nu\lambda X^2},\ \mu(1-\mu\varDelta^2),\ \mu(1-\mu\delta^2)\right)\\
&\leq \frac{1}{T}\sum_{t=1}^{T} R(\mathbf{w}^*, \mathbf{u}^*;\ \lambda, 0, \mu) + R(\mathbf{w}^*, \mathbf{u}^t;\ 0, \mu, 0) + R(\mathbf{w}^{t+1}, \mathbf{u}^*;\ 0, \mu, 0) \qquad (18)\\
&\qquad\qquad + \frac{1}{\nu T}\|\mathbf{w}^*\|^2 + \frac{1}{T}\sum_{i=1}^{M}\|\mathbf{u}^{i,*}\|^2.
\end{aligned}$$

If the last two R terms in the right hand side are bounded by $2R(\mathbf{w}^*, \mathbf{u}^*; 0, \mu, 0)$, then the regret behavior becomes similar to truncated-gradient algorithms [8].

5 Experiments

We performed experiments on three data sets: one artificial (see Figure 1) and two real world. Our real world data sets are Pima Indians Diabetes data set from the UCI repository [2] and *M. tuberculosis* spoligotype data set (both are described below). We also created a synthetic data set where one class of the data corresponded to a mixture of two small σ Gaussians and the other (overlapping) class was represented by a flatter (large σ) Gaussian. For this set, the learner is provided with three hand-made advice sets (see Figure 1).

5.1 `Diabetes` Data Set

The `diabetes` data set consists of 768 points with 8 attributes. For domain advice, we constructed two rules based on statements from the NIH web site on risks for Type-2 Diabetes[2]. A person who is obese, characterized by high body mass index (`BMI` ≥ 30) and high `bloodglucose` level (≥ 126) is at strong risk for diabetes, while a person who is at normal weight (`BMI` ≤ 25) and low `bloodglucose` level (≤ 100) is unlikely to have diabetes. As `BMI` and `bloodglucose` are features of the data set, we can give advice by combining these conditions into conjunctive rules, one for each class. For instance, the rule predicting that diabetes is false is (`BMI` ≤ 25) $\wedge$ (`bloodglucose` ≤ 100) $\Rightarrow \neg$`diabetes`.

5.2 `Tuberculosis` Data Set

These data sets consist of two types of DNA fingerprints of *M. tuberculosis* complex (MTBC): the spacer oglionucleotide types (spoligotypes) and Mycobacterial

[1] The complete derivation can be found at `http://ftp.cs.wisc.edu/machine-learning/shavlik-group/kunapuli.ecml10.proof.pdf`

[2] `http://diabetes.niddk.nih.gov/DM/pubs/riskfortype2`

Class	#isolates	#pieces of Positive Advice	#pieces of Negative Advice
East-Asian	4924	1	1
East-African-Indian	1469	2	4
Euro-American	25161	1	2
Indo-Oceanic	5309	5	5
M. africanum	154	1	3
M. bovis	693	1	3

Table 1: The number of isolates for each MTBC class and the number of positive and negative pieces of advice for each classification task. Each task consisted of 50 training examples drawn randomly from the isolates with the rest becoming test examples.

Interspersed Repetitive Units (MIRU) types of 37942 clinical isolates collected by the US Centers for Disease Control and Prevention (CDC) during 2004–2008 as part of routine TB surveillance and control. The spoligotype captures the variability in the direct repeat (DR) region of the genome of a strain of MTBC and is represented by a 43-bit long binary string constructed on the basis of presence or absence of spacers (non-repeating sequences interspersed between short direct repeats) in the DR. In addition, the number of repeats present at the 24th locus of the MIRU type (`MIRU24`) is used as an attribute.

Six major lineages of strains of the MTBC have been previously identified: the "modern" lineages: Euro-American, East-Asian and East-African-Indian and the "ancestral" lineages: *M. bovis*, *M. africanum* and Indo-Oceanic. Prior studies report high classification accuracy of the major genetic lineages using Bayesian Networks on spoligotypes and up to 24 loci of MIRU [1] on this dataset. Expert-defined rules for the classification of MTBC strains into these lineages have been previously documented [3, 14]. The rules are based on observed patterns in the presence or absence of spacers in the spoligotypes, and in the number of tandem repeats at MIRU of a single MIRU locus – `MIRU24`, associated with each lineage. The `MIRU24` locus is known to distinguish ancestral versus modern lineages with high accuracy for most isolates with a few exceptions.

The six TB classification tasks are to distinguish each lineage from the rest. The advice consists of positive advice to identify each lineage, as well as negative advice that rules out specific lineages. We found that incorporation of negative advice for some classes like *M. africanum* significantly improved performance. The number of isolates for each class and the number of positive and negative pieces of advice for each classification task are given in Table 1. Examples of advice are provided below[3].

```
Spacers(1-34) absent ⇒ East-Asian
At least one of Spacers(1-34) present ⇒ ¬East-Asian
Spacers(4-7, 23-24, 29-32) absent ∧ MIRU24≤1 ⇒ East-African-Indian
Spacers(4-7, 23-24) absent ∧ MIRU24≤1 ∧ at least one spacer of (29-32)
present ∧ at least one spacer of (33-36) present⇒ East-African-Indian
```

[3] The full rules can be found in `http://ftp.cs.wisc.edu/machine-learning/shavlik-group/kunapuli.ecml10.rules.pdf`

`Spacers(3, 9, 16, 39-43) absent` $\wedge$ `spacer 38 present` $\Rightarrow$ *`M. bovis`*
`Spacers(8, 9, 39) absent` $\wedge$ `MIRU24>1` $\Rightarrow$ *`M. africanum`*
`Spacers(3, 9, 16, 39-43) absent` $\wedge$ `spacer 38 present` $\Rightarrow \neg$ *`M. africanum`*

For each lineage, both negative and positive advice can be naturally expressed. For example, the positive advice for *M. africanum* closely corresponds to a known rule: `if spacers(8, 9, 13) are absent` $\wedge$ `MIRU24` $\leq 1 \Rightarrow$ *`M. africanum`*. However, this rule is overly broad and is further refined by exploiting the fact that *M. africanum* is an ancestral strain. Thus, the following rules out all modern strains: `if MIRU24` $\leq 1 \Rightarrow \neg$ *`M. africanum`*. The negative advice captures the fact that spoligotypes do not regain spacers once lost. For example, `if at least one of Spacers(8, 9, 39) is present` $\Rightarrow \neg$ *`M. africanum`*. The final negative rule rules out *M. bovis*, a close ancestral strain easily confused with *M. africanum*.

5.3 Methodology

The results for each data set are averaged over multiple randomized iterations (20 iterations for synthetic and `diabetes`, and 200 for the `tuberculosis` tasks). For each iteration of the synthetic and `diabetes` data sets, we selected 200 points at random as the training set and used the rest as the test set. For each iteration of the `tuberculosis` data sets, we selected 50 examples at random from the data set to use as a training set and tested on the rest. Each time, the training data was presented in a random order, one example at a time, to the learner to generate the learning curves shown in Figures 2(a)–2(h). We compare the results to well-studied incremental algorithms: standard passive-aggressive algorithms [4], margin-perceptron [5] and ROMMA [10]. We also compare it to the standard batch KBSVM [6], where the learner was given all of the examples used in training the online learners (e.g., for the synthetic data we had 200 data points to create the learning curve, so the KBSVM used those 200 points).

5.4 Analysis of Results

For both artificial and real world data sets, the advice leads to significantly faster convergence of accuracy over the no-advice approaches. This reflects the intuitive idea that a learner, when given prior knowledge that is useful, will be able to more quickly find a good solution. In each case, note also, that the learner is able to use the learning process to improve on the starting accuracy (which would be produced by advice only). Thus, the Adviceptron is able to learn effectively from both data *and* advice.

A second point to note is that, in some cases, prior knowledge allows the learner to converge on a level of accuracy that is not achieved by the other methods, which do not benefit from advice. While the results demonstrate that advice can make a significant difference when learning with small data sets, in many cases, large amounts of data may be needed by the advice-free algorithms to eventually achieve performance similar to the Adviceptron. This shows that advice can provide large improvements over just learning with data.

Finally, it can be seen that, in most cases, the generalization performance of the Adviceptron converges rapidly to that of the batch-KBSVM. However, the batch-KBSVMs take, on average, 15–20 seconds to compute an optimal solution as they have to solve a quadratic program. In contrast, owing to the simple, closed-form update rules, the Adviceptron is able to obtain identical test-set performance in under 5 seconds on average. Further scalability experiments represent one of the more immediate directions of future work. One minor point to note is regarding the results on East-Asian and *M. bovis* (Figures 2(e) and 2(h)): the advice (provided by a tuberculosis domain expert) was so effective that these problems were almost immediately learned (with few to no examples).

6 Conclusions and Related Work

We have presented a new online learning method, the Adviceptron, that is a novel approach that makes use of prior knowledge in the form of polyhedral advice. This approach is an online extension to KBSVMs [6] and differs from previous polyhedral advice-taking approaches and the neural-network-based KBANN [15] in two significant ways: it is an online method with closed-form solutions and it provides a theoretical mistake bound.

The advice-taking approach was incorporated into the passive-aggressive framework because of its many appealing properties including efficient update rules and simplicity. Advice updates in the adviceptron are computed using a projected-gradient approach similar to the truncated-gradient approaches by Langford et al., [8]. However, the advice updates are truncated far more aggressively. The regret bound shows that as long as the projection being considered is non-expansive, it is still possible to minimize regret.

We have presented a bound on the effectiveness of this method and a proof of that bound. In addition, we performed several experiments on artificial and real world data sets that demonstrate that a learner with reasonable advice can significantly outperform a learner without advice. We believe our approach can serve as a template for other methods to incorporate advice into online learning methods.

One drawback of our approach is the restriction to certain types of loss functions. More direct projected-gradient approach or other related online convex programming [17] approaches can be used to develop algorithms with similar properties. This also allows for the derivation of general algorithms for different loss functions. KBSVMs can also be extended to kernels as shown in [6], and is yet another direction of future work.

Acknowledgements

The authors would like to thank Dr. Lauren Cowan of the CDC for providing the TB dataset and the expert-defined rules for lineage classification. The authors gratefully acknowledge support of the Defense Advanced Research Projects Agency under DARPA grant FA8650-06-C-7606 and the National Institute of Health under NIH grant

1-R01-LM009731-01. Views and conclusions contained in this document are those of the authors and do not necessarily represent the official opinion or policies, either expressed or implied of the US government or of DARPA.

References

1. M. Aminian, A. Shabbeer, and K. P. Bennett. A conformal Bayesian network for classification of Mycobacterium tuberculosis complex lineages. *BMC Bioinformatics*, 11(Suppl 3):S4, 2010.
2. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
3. K. Brudey, J.R. Driscoll, L. Rigouts, W.M. Prodinger, A. Gori, S.A. Al-Hajoj, C. Allix, L. Aristimuño, J. Arora, V. Baumanis, et al. *Mycobacterium tuberculosis* complex genetic diversity: Mining the fourth international spoligotyping database (spoldb 4) for classification, population genetics and epidemiology. *BMC Microbiology*, 6(1):23, 2006.
4. K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *J. of Mach. Learn. Res.*, 7:551–585, 2006.
5. Y. Freund and R. E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.
6. G. Fung, O. L. Mangasarian, and J. W. Shavlik. Knowledge-based support vector classifiers. In S. Becker, S. Thrun, and K. Obermayer, editors, *NIPS*, volume 15, pages 521–528, 2003.
7. S. Gagneux and P. M. Small. Global phylogeography of Mycobacterium tuberculosis and implications for tuberculosis product development. *The Lancet Infectious Diseases*, 7(5):328–337, 2007.
8. J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *J. Mach. Learn. Res.*, 10:777–801, 2009.
9. F. Lauer and G. Bloch. Incorporating prior knowledge in support vector machines for classification: A review. *Neurocomp.*, 71(7-9):1578–1594, 2008.
10. Y. Li and P. M. Long. The relaxed online maximum margin algorithm. *Mach. Learn.*, 46(1/3):361–387, 2002.
11. O. L. Mangasarian. *Nonlinear Programming*. McGraw-Hill, 1969.
12. B. Schölkopf, P. Simard, A. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In *NIPS*, volume 10, pages 640–646, 1998.
13. B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization Optimization and Beyond*. MIT Press, Cambridge, MA, USA, 2001.
14. A. Shabbeer, L. Cowan, J. R. Driscoll, C. Ozcaglar, S. L Vandenberg, B. Yener, and K. P Bennett. TB-Lineage: An online tool for classification and analysis of strains of *Mycobacterium tuberculosis* Complex. *Unpublished manuscript*, 2010.
15. G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1-2):119–165, 1994.
16. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.
17. M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. 20th Int. Conf. on Mach. Learn.*, 2003.

Appendix

Proof of Lemma 2

The progress at trial t is $\Delta_t = \frac{1}{2}\|\mathbf{w}^* - \mathbf{w}^{t+1}\|^2 - \frac{1}{2}\|\mathbf{w}^* - \mathbf{w}^t\|^2 = \dfrac{1}{2}\|\mathbf{w}^{t+1} - \mathbf{w}^t\|^2 + (\mathbf{w}^t - \mathbf{w}^*)'(\mathbf{w}^{t+1} - \mathbf{w}^t)$. Substituting $\mathbf{w}^{t+1} - \mathbf{w}^t = \nu\, \alpha_t y_t \mathbf{x}^t + (1-\nu)(\mathbf{r}^t - \mathbf{w}^t)$, from the update rules, we have

$$\varDelta_t \leq \frac{1}{2}\nu^2\alpha_t^2\|\mathbf{x}^t\|^2 + \nu\alpha_t\left(\nu\, y_t{\mathbf{w}^t}'\mathbf{x}^t + (1-\nu)\, y_t{\mathbf{r}^t}'\mathbf{x}^t - y_t{\mathbf{w}^*}'\mathbf{x}^t\right) + \frac{1}{2}(1-\nu)\,\|\mathbf{r}^t - \mathbf{w}^*\|^2.$$

The loss suffered by the adviceptron is defined in (8). We focus only on the case when the loss is > 0. Define $\widehat{\mathbf{w}}^t = \nu\mathbf{w}^t + (1-\nu)\mathbf{r}^t$. Then, we have $1 - L_\xi(\widehat{\mathbf{w}}^t) = \nu\, y_t{\mathbf{w}^t}'\mathbf{x}^t + (1-\nu)\, y_t{\mathbf{r}^t}'\mathbf{x}^t$. Furthermore, by definition, $L_\xi(\mathbf{w}^*) \geq 1 - y_t{\mathbf{w}^*}'\mathbf{x}^t$. Using these two results,

$$\varDelta_t \;\leq\; \frac{1}{2}\nu^2\alpha_t^2\|\mathbf{x}^t\|^2 + \nu\,\alpha_t(L_\xi(\mathbf{w}^*) - L_\xi(\mathbf{w}^t)) + \frac{1}{2}(1-\nu)\,\|\mathbf{r}^t - \mathbf{w}^*\|^2. \qquad (19)$$

Adding $\frac{1}{2}\,\nu\,(\frac{\alpha_t}{\sqrt{\lambda}} - \sqrt{\lambda}\ell_t^*)^2$ to the left-hand side of the and simplifying, using Update 1:

$$\varDelta_t \leq \frac{1}{2}\,\frac{\nu\, L_\xi(\mathbf{w}^t)^2}{\left(\dfrac{1}{\lambda} + \nu\,\|\mathbf{x}^t\|^2\right)} - \frac{1}{2}\,\nu\,\lambda L_\xi(\mathbf{w}^*)^2 + \frac{1}{2}(1-\nu)\,\|\mathbf{r}^t - \mathbf{w}^*\|^2.$$

Rearranging the terms above and using $\|\mathbf{x}^t\|^2 \leq X^2$ gives the bound. □

Proof of Lemma 3

Let $\widetilde{\mathbf{u}}^{i,t} = \mathbf{u}^{i,t} + D_i\boldsymbol{\beta}^i - \mathbf{d}^i\gamma_i$ be the update before the projection onto $\mathbf{u} \geq 0$. Then, $\mathbf{u}^{i,t+1} = \widetilde{\mathbf{u}}^{i,t}_+$. We also write $\mathcal{L}_{advice}(\mathbf{u}^{i,t})$ compactly as $\mathcal{L}(\mathbf{u}^{i,t})$. Then,

$$\begin{aligned}
\frac{1}{2}\|\mathbf{u}^* - \mathbf{u}^{i,t+1}\|^2 &\leq \frac{1}{2}\|\mathbf{u}^* - \widetilde{\mathbf{u}}^{i,t}\|^2 \\
&= \frac{1}{2}\|\mathbf{u}^* - \mathbf{u}^{i,t}\|^2 + \frac{1}{2}\|\mathbf{u}^{i,t} - \widetilde{\mathbf{u}}^{i,t}\|^2 + (\mathbf{u}^* - \mathbf{u}^{i,t})'(\mathbf{u}^{i,t} - \widetilde{\mathbf{u}}^{i,t}) \\
&= \frac{1}{2}\|\mathbf{u}^* - \mathbf{u}^{i,t}\|^2 + \frac{\mu^2}{2}\|\nabla_{\mathbf{u}^i}\mathcal{L}(\mathbf{u}^{i,t})\|^2 + \mu(\mathbf{u}^* - \mathbf{u}^{i,t})'\nabla_{\mathbf{u}^i}\mathcal{L}(\mathbf{u}^{i,t})
\end{aligned}$$

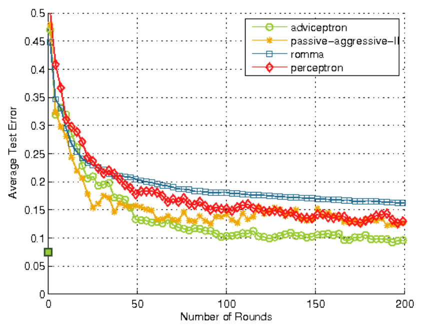
The first inequality is due to the non-expansiveness of projection and the next steps follow from Lemma 1.1. Let $\varDelta_t = \frac{1}{2}\|\mathbf{u}^* - \mathbf{u}^{i,t+1}\|^2 - \frac{1}{2}\|\mathbf{u}^* - \mathbf{u}^{i,t}\|^2$. Using Lemma 1.2, we have

$$\begin{aligned}
\varDelta_t \leq\, & \frac{\mu^2}{2}\left(\|D_i\|^2 L_\eta(\mathbf{u}^{i,t},\mathbf{w}^t) + \|\mathbf{d}^i\|^2 L_\zeta(\mathbf{u}^{i,t})^2\right) \\
& \quad + \mu(\mathbf{u}^* - \mathbf{u}^{i,t})'\left(\frac{1}{2}\nabla_{\mathbf{u}^i} L_\eta(\mathbf{u}^{i,t},\mathbf{w}^t) + L_\zeta(\mathbf{u}^{i,t})\nabla_{\mathbf{u}^i} L_\zeta(\mathbf{u}^{i,t})\right) \\
\leq\, & \frac{\mu^2}{2}\left(\|D_i\|^2 L_\eta(\mathbf{u}^{i,t},\mathbf{w}^t) + \|\mathbf{d}^i\|^2 L_\zeta(\mathbf{u}^{i,t})^2\right) \\
& \quad + \frac{\mu}{2}\left(L_\eta(\mathbf{u}^*,\mathbf{w}^t) - L_\eta(\mathbf{u}^{i,t},\mathbf{w}^t)\right) + \frac{\mu}{2}\left(L_\zeta(\mathbf{u}^*)^2 - L_\zeta(\mathbf{u}^{i,t})^2\right)
\end{aligned}$$

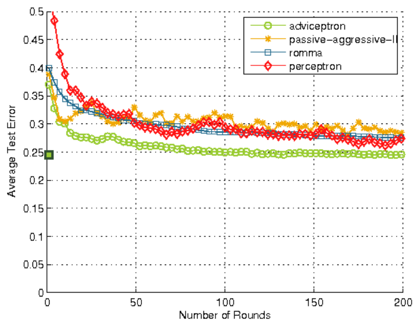
where the last step follows from the convexity of the loss function L_η and the fact that

$$\begin{aligned}
& L_\zeta(\mathbf{u}^{i,t})(\mathbf{u}^* - \mathbf{u}^{i,t})'\nabla_{\mathbf{u}^i} L_\zeta(\mathbf{u}^{i,t}) \\
& \qquad \leq L_\zeta(\mathbf{u}^{i,t})\left(L_\zeta(\mathbf{u}^*) - L_\zeta(\mathbf{u}^{i,t})\right) \quad \text{(convexity of } L_\zeta\text{)} \\
& \qquad \leq L_\zeta(\mathbf{u}^{i,t})\left(L_\zeta(\mathbf{u}^*) - L_\zeta(\mathbf{u}^{i,t})\right) + \frac{1}{2}\left(L_\zeta(\mathbf{u}^*) - L_\zeta(\mathbf{u}^{i,t})\right)^2.
\end{aligned}$$

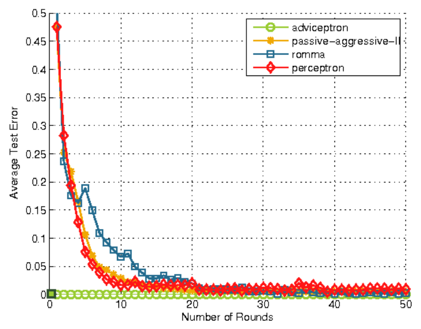
Rearranging the terms and bounding $\|D_i\|^2$ and $\|\mathbf{d}^i\|^2$ proves the lemma. □



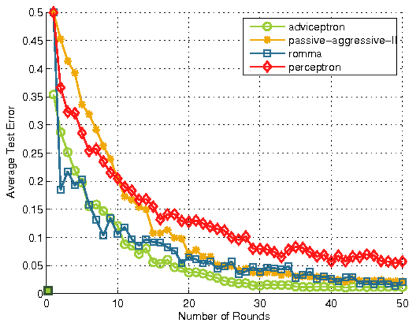
(a) Synthetic Data



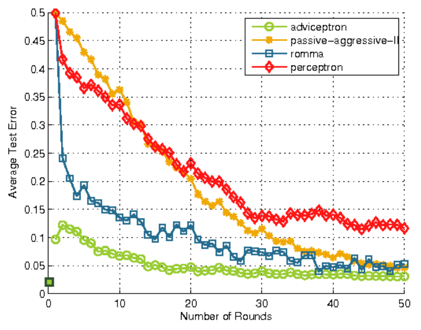
(b) Diabetes



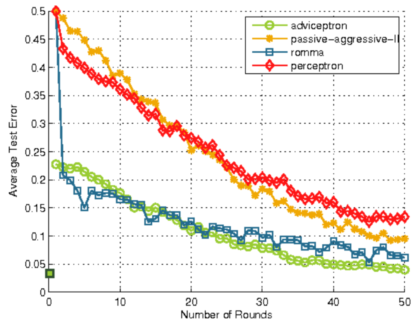
(c) Tuberculosis: East-Asian



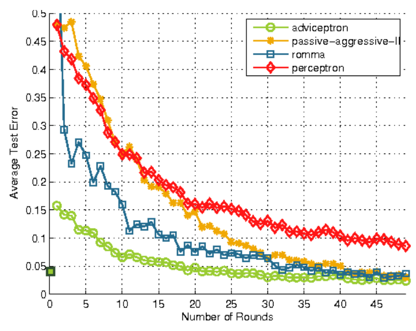
(d) Tuberculosis: East-African-Indian



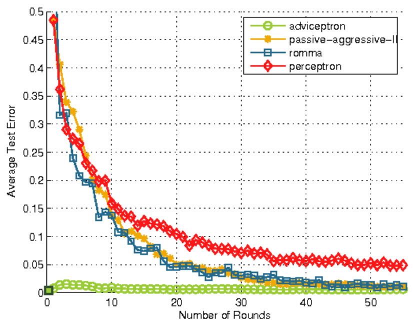
(e) Tuberculosis: Euro-American



(f) Tuberculosis: Indo-Oceanic



(g) Tuberculosis: *M. africanum*



(h) Tuberculosis: *M. bovis*

Fig. 2: Results comparing the Adviceptron to standard passive-aggressive, ROMMA and perceptron, where one example is presented at each round. The baseline KBSVM results are shown as a square on the y-axis for clarity; in each case, batch-KBSVM uses the entire training set available to the online learners.