E0 370 Statistical Learning Theory **Lecture 18 (Nov 8, 2011)**

Online Classification: Perceptron and Winnow

Lecturer: Shivani Agarwal *Scribe: Shivani Agarwal*

1 Introduction

In this lecture we will start to study the online learning setting that was discussed briefly in the first lecture. Unlike the batch setting we have studied so far, where one is given a sample or 'batch' of training data and the goal is to learn from this data a model that can make accurate predictions in the future, in the online setting, learning takes place in a sequence of trials: on each trial, the learner must make a prediction or take some action, each of which can potentially result in some loss, and the goal is to update the prediction/decision model at the end of each trial so as to minimize the total loss incurred over a sequence of such trials.

Online learning is relevant for a variety of problems, including prediction problems (e.g. forecasting the weather the next day) and decision/allocation problems (e.g. investing in different stocks or mutual funds). We will start by considering online supervised learning problems, where on each trial, the learner receives an instance and must predict its label, following which the true label is revealed and a corresponding loss incurred; as noted above, the goal of the learner is to minimize the total loss over a sequence of trials. We will focus in this lecture on online binary classification problems, and in the next lecture on online regression problems. We will then discuss online learning from experts, a framework that can be useful for both online supervised learning problems and online decision/allocation problems; we will analyze this framework in some detail in a couple of lectures, and then will conclude in the last lecture with a brief discussion of how online learning algorithms and their analyses can be transported back into the batch setting.

The basic online binary classification setting can be described as follows:

Online (Binary) Classification
For $t = 1, \ldots, T$:
– Receive instance $x^t \in \mathcal{X}$
– Predict $\hat{y}^t \in \{\pm 1\}$
– Receive true label $y^t \in \{\pm 1\}$
– Incur loss $\ell(y^t, \hat{y}^t)$

The goal of a learning algorithm in this setting is to minimize the total loss incurred. Specifically, let $S = ((x^1, y^1), \ldots, (x^T, y^T))$. Then the *cumulative loss* of an algorithm $\mathcal{A}$ on the trial sequence S is given by

$$L_S^{\ell}[\mathcal{A}] \;=\; \sum_{t=1}^{T} \ell(y^t, \hat{y}^t)\,. \tag{1}$$

The goal is to design algorithms with small cumulative loss on any trial sequence (or any trial sequence satisfying certain properties); the analysis here is therefore worst-case, rather than probabilistic as in the batch setting. For binary classification with zero-one loss $\ell_{0\text{-}1}$, the cumulative loss of an algorithm over a trial sequence S corresponds to the number of prediction mistakes made by the algorithm on this sequence; bounds on the cumulative zero-one loss $L_S^{0\text{-}1}[\mathcal{A}]$ are therefore termed *mistake bounds*. In the following, we will study two classical algorithms for online binary classification, namely the perceptron and winnow algorithms, and discuss the mistake bounds that can be derived for them.

2 Perceptron

In its basic form, the perceptron algorithm applies to Euclidean instance spaces $\mathcal{X} \subseteq \mathbb{R}^n$, and maintains a linear classifier (represented by a weight vector) in such a space:

Algorithm **Perceptron**

Initial weight vector $\mathbf{w}^1 = \mathbf{0} \in \mathbb{R}^n$
For $t = 1, \ldots, T$:
- Receive instance $\mathbf{x}^t \in \mathcal{X} \subseteq \mathbb{R}^n$
- Predict $\hat{y}^t = \text{sign}(\mathbf{w}^t \cdot \mathbf{x}^t)$
- Receive true label $y^t \in \{\pm 1\}$
- Incur loss $\ell_{0\text{-}1}(y^t, \hat{y}^t)$
- Update: If $\hat{y}^t \neq y^t$ then
  $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t + y^t \mathbf{x}^t$
  else
  $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t$

Notice that the algorithm makes an update to its model (weight vector) only when there is a mistake in its prediction; online algorithms satisfying this property are said to be *conservative*. To get an intuitive feel for the algorithm, observe that if the true label y^t on trial t is $+1$ and the algorithm predicts $\hat{y}^t = -1$, then it means $\mathbf{w}^t \cdot \mathbf{x}^t < 0$; in order to improve the prediction on this example, the algorithm must increase the value of this dot product. Indeed, we have $\mathbf{w}^{t+1} \cdot \mathbf{x}^t = \mathbf{w}^t \cdot \mathbf{x}^t + \|\mathbf{x}^t\|_2^2 \geq \mathbf{w}^t \cdot \mathbf{x}^t$. Similarly, it can be verified that when $y^t = -1$ and the algorithm predicts $\hat{y}^t = +1$, the update has the effect of decreasing the value of the dot product. Thus the updates make sense intuitively. More formally, one can prove the following classical mistake bound for the perceptron algorithm in the linearly separable case:

Theorem 2.1 (Perceptron Convergence Theorem; Block, 1962; Novikoff, 1962)**.** Let $S = ((\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^T, y^T)) \in (\mathbb{R}^n \times \{\pm 1\})^T$. Let $R_2 = \max\{\|\mathbf{x}^t\|_2 \mid t \in [T]\}$ and let $\gamma > 0$. Then for any $\mathbf{u} \in \mathbb{R}^n$ such that $y^t(\mathbf{u} \cdot \mathbf{x}^t) \geq \gamma \; \forall t \in [T]$,

$$L_S^{0\text{-}1}[\text{Perceptron}] \;\leq\; \frac{R_2^2 \|\mathbf{u}\|_2^2}{\gamma^2}\,.$$

Proof. Denote $L_S^{0\text{-}1}[\text{Perceptron}] = k$. Consider measuring the 'progress towards $\mathbf{u}$' or 'closeness to $\mathbf{u}$' on each trial in terms of $\mathbf{w}^t \cdot \mathbf{u}$. For each trial t on which there is a mistake, we have

$$\mathbf{w}^{t+1} \cdot \mathbf{u} - \mathbf{w}^t \cdot \mathbf{u} \;=\; y^t \mathbf{x}^t \cdot \mathbf{u} \;\geq\; \gamma\,. \tag{2}$$

For all other trials t, we have $\mathbf{w}^{t+1} \cdot \mathbf{u} - \mathbf{w}^t \cdot \mathbf{u} = 0$. Therefore summing over $t = 1, \ldots, T$ gives

$$\mathbf{w}^{T+1} \cdot \mathbf{u} - \mathbf{w}^1 \cdot \mathbf{u} \;\geq\; k\,\gamma\,. \tag{3}$$

Noting that $\mathbf{w}^1 = \mathbf{0}$ and using Cauchy-Schwartz, we have

$$k \;\leq\; \frac{\mathbf{w}^{T+1} \cdot \mathbf{u}}{\gamma} \;\leq\; \frac{\|\mathbf{w}^{T+1}\|_2 \|\mathbf{u}\|_2}{\gamma}\,. \tag{4}$$

Now for each trial t on which there is a mistake,

$$\begin{aligned} \|\mathbf{w}^{t+1}\|_2^2 &= \|\mathbf{w}^t\|_2^2 + 2y^t \mathbf{w}^t \cdot \mathbf{x}^t + \|\mathbf{x}^t\|_2^2 && (5)\\ &\leq \|\mathbf{w}^t\|_2^2 + R_2^2 \qquad (\text{since } y^t \mathbf{w}^t \cdot \mathbf{x}^t \leq 0 \text{ for a mistake trial})\,. && (6) \end{aligned}$$

For all other trials t, $\|\mathbf{w}^{t+1}\|_2^2 - \|\mathbf{w}^t\|_2^2 = 0$. Therefore summing over $t = 1, \ldots, T$ and noting again $\mathbf{w}^1 = \mathbf{0}$, we get

$$\|\mathbf{w}^{T+1}\|_2^2 \;\leq\; k R_2^2\,. \tag{7}$$

Substituting in Eq. (4) gives

$$k \;\leq\; \frac{\sqrt{k} R_2 \|\mathbf{u}\|_2}{\gamma}\,. \tag{8}$$

Squaring both sides yields the result. □

One can also show the following weaker mistake bound in the general (non-separable) case:

Theorem 2.2 (Freund and Schapire; 1999)**.** Let $S = ((\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^T, y^T)) \in (\mathbb{R}^n \times \{\pm 1\})^T$. Let $R_2 = \max\{\|\mathbf{x}^t\|_2 \mid t \in [T]\}$ and let $\gamma > 0$. Then for any $\mathbf{u} \in \mathbb{R}^n$,

$$L_S^{0\text{-}1}[\text{Perceptron}] \;\leq\; \left(\frac{R_2 \|\mathbf{u}\|_2 + \sqrt{\sum_{t=1}^T \left((\gamma - y^t(\mathbf{u} \cdot \mathbf{x}^t))_+\right)^2}}{\gamma} \right)^2 .$$

Details of the proof can be found in [1]. We conclude our discussion of the perceptron algorithm by observing that the algorithm can be re-written so as to use only dot products between instances seen by the algorithm, which facilitates a natural extension to a kernel-based variant for arbitrary instance spaces $\mathcal{X}$:

Algorithm **Kernel Perceptron**

Kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
For $t = 1, \ldots, T$:
- Receive instance $x^t \in \mathcal{X}$
- Predict $\hat{y}^t = \text{sign}\left(\sum_{r=1}^{t-1} \alpha_r K(x^r, x^t)\right)$
- Receive true label $y^t \in \{\pm 1\}$
- Incur loss $\ell_{0\text{-}1}(y^t, \hat{y}^t)$
- Update: If $\hat{y}^t \neq y^t$ then
  $\alpha_t \leftarrow y_t$
  else
  $\alpha_t \leftarrow 0$

A mistake bound similar to that for the linear perceptron algorithm can be shown in this case too:

Theorem 2.3 (Kernel Perceptron Convergence Theorem)**.** Let $S = ((x^1, y^1), \ldots, (x^T, y^T)) \in (\mathcal{X} \times \{\pm 1\})^T$ and let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function on $\mathcal{X}$. Let $R_2 = \max\{\sqrt{K(x^t, x^t)} \mid t \in [T]\}$ and let $\gamma > 0$. Then for any $u \in \mathcal{X}$ such that $y^t K(u, x^t) \geq \gamma \; \forall t \in [T]$,

$$L_S^{0\text{-}1}[\text{Perceptron}(K)] \;\leq\; \frac{R_2^2 \, K(u,u)}{\gamma^2} .$$

We leave the proof details as an exercise.

3 Winnow

The winnow algorithm also maintains a linear classifier in a Euclidean instance space; in this case, however, the updates to the weight vector are multiplicative rather than additive:

Algorithm **Winnow**

Learning rate parameter $\eta > 0$
Initial weight vector $\mathbf{w}^1 = (\frac{1}{n}, \ldots, \frac{1}{n}) \in \mathbb{R}^n$
For $t = 1, \ldots, T$:
- Receive instance $\mathbf{x}^t \in \mathcal{X} \subseteq \mathbb{R}^n$
- Predict $\hat{y}^t = \text{sign}(\mathbf{w}^t \cdot \mathbf{x}^t)$
- Receive true label $y^t \in \{\pm 1\}$
- Incur loss $\ell_{0\text{-}1}(y^t, \hat{y}^t)$
- Update: If $\hat{y}^t \neq y^t$ then
  $\forall i \in [n]$: $\quad w_i^{t+1} \leftarrow \dfrac{w_i^t \exp(\eta y^t x_i^t)}{Z_t}$
  where $Z_t = \sum_{j=1}^n w_j^t \exp(\eta y^t x_j^t)$
  else
  $\mathbf{w}^{t+1} \leftarrow \mathbf{w}^t$

Here too, one can observe that when a mistake is made on some trial t, the effect of the update is to move the dot product $\mathbf{w}^{t+1} \cdot \mathbf{x}^t$ in the right direction compared to $\mathbf{w}^t \cdot \mathbf{x}^t$. Formally, we have the following mistake bound (for trial sequences that are linearly separable by a non-negative weight vector):

Theorem 3.1. Let $S = ((\mathbf{x}^1, y^1), \ldots, (\mathbf{x}^T, y^T)) \in (\mathbb{R}^n \times \{\pm 1\})^T$. Let $R_\infty = \max\{\|\mathbf{x}^t\|_\infty \mid t \in [T]\}$ and let $\gamma > 0$. Then for any $\mathbf{u} \in \mathbb{R}^n$ such that $u_i \geq 0 \ \forall i \in [n]$ and $y^t(\mathbf{u} \cdot \mathbf{x}^t) \geq \gamma \ \forall t \in [T]$,

$$L_S^{0\text{-}1}[\text{Winnow}(\eta)] \ \leq \ \frac{\|\mathbf{u}\|_1 \ln n}{\eta\gamma - \|\mathbf{u}\|_1 \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right)}\,.$$

Moreover, if R_∞, $\|\mathbf{u}\|_1$, and γ are known, then one can select η^* to yield

$$L_S^{0\text{-}1}[\text{Winnow}(\eta^*)] \ \leq \ 2\left(\frac{R_\infty^2 \|\mathbf{u}\|_1^2}{\gamma^2}\right) \ln n\,.$$

Proof. Denote $L_S^{0\text{-}1}[\text{Winnow}(\eta)] = k$, and let $\mathbf{p} = \mathbf{u}/\|\mathbf{u}\|_1$ so that $\mathbf{p} \in \Delta_n$, where Δ_n is the probability simplex in $\mathbb{R}^n$. Consider again measuring the 'progress towards $\mathbf{u}$ (or $\mathbf{p}$)' on each trial; in this case, we will measure the 'distance of $\mathbf{w}^t$ from $\mathbf{p}$' in terms of the KL-divergence, $\text{KL}(\mathbf{p}\|\mathbf{w}^t) = \sum_{i=1}^n p_i \ln\left(\frac{p_i}{w_i^t}\right)$. For each trial t on which there is a mistake, we have

$$\begin{aligned}
\text{KL}(\mathbf{p}\|\mathbf{w}^t) - \text{KL}(\mathbf{p}\|\mathbf{w}^{t+1}) &= \sum_{i=1}^n p_i \ln\left(\frac{w_i^{t+1}}{w_i^t}\right) && (9)\\
&= \sum_{i=1}^n p_i \ln\left(\frac{e^{\eta y^t x_i^t}}{Z_t}\right) && (10)\\
&= \eta y^t \sum_{i=1}^n p_i x_i^t - \sum_{i=1}^n p_i \ln Z_t && (11)\\
&= \eta y^t (\mathbf{p} \cdot \mathbf{x}^t) - \ln Z_t && (12)\\
&\geq \frac{\eta\gamma}{\|\mathbf{u}\|_1} - \ln Z_t\,. && (13)
\end{aligned}$$

Now, $Z_t = \sum_{i=1}^n w_i^t e^{\eta y^t x_i^t}$. Noting that $y^t x_i^t \in [-R_\infty, R_\infty]$ for all i, t, we can bound Z_t as follows (using convexity of the mapping $t \mapsto e^{\eta t}$):

$$\begin{aligned}
Z_t &\leq \sum_{i=1}^n w_i^t \left(\left(\frac{1 + y^t x_i^t / R_\infty}{2}\right) e^{\eta R_\infty} + \left(\frac{1 - y^t x_i^t / R_\infty}{2}\right) e^{-\eta R_\infty}\right) && (14)\\
&= \left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right) \sum_{i=1}^n w_i^t + \left(\frac{e^{\eta R_\infty} - e^{-\eta R_\infty}}{2}\right) y^t \sum_{i=1}^n w_i^t x_i^t && (15)\\
&= \left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right) + \left(\frac{e^{\eta R_\infty} - e^{-\eta R_\infty}}{2}\right) y^t \mathbf{w}^t \cdot \mathbf{x}^t && (16)\\
&\leq \left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right) \quad \text{(since } e^{\eta R_\infty} - e^{-\eta R_\infty} > 0\text{, and } y^t \mathbf{w}^t \cdot \mathbf{x}^t \leq 0 \text{ for mistake trials } t\text{).} && (17)
\end{aligned}$$

Therefore, on each mistake trial t, we have

$$\text{KL}(\mathbf{p}\|\mathbf{w}^t) - \text{KL}(\mathbf{p}\|\mathbf{w}^{t+1}) \ \geq \ \frac{\eta\gamma}{\|\mathbf{u}\|_1} - \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right)\,. \tag{18}$$

On all other trials t,

$$\text{KL}(\mathbf{p}\|\mathbf{w}^t) - \text{KL}(\mathbf{p}\|\mathbf{w}^{t+1}) \ = \ 0\,. \tag{19}$$

Therefore summing over $t = 1, \ldots, T$, we have

$$\text{KL}(\mathbf{p}\|\mathbf{w}^1) - \text{KL}(\mathbf{p}\|\mathbf{w}^{T+1}) \ \geq \ k\left(\frac{\eta\gamma}{\|\mathbf{u}\|_1} - \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right)\right)\,. \tag{20}$$

Now,

$$\mathrm{KL}(\mathbf{p}\|\mathbf{w}^1) \leq \ln n\,. \tag{21}$$

$$\mathrm{KL}(\mathbf{p}\|\mathbf{w}^{T+1}) \geq 0\,. \tag{22}$$

This yields the desired bound

$$k \leq \frac{\|\mathbf{u}\|_1 \ln n}{\eta\gamma - \|\mathbf{u}\|_1 \ln\left(\frac{e^{\eta R_\infty} + e^{-\eta R_\infty}}{2}\right)}\,. \tag{23}$$

Now if R_∞, $\|\mathbf{u}\|_1$, and γ are known, then one can minimize the right hand side above w.r.t. η; this yields

$$\eta^* = \frac{1}{2R_\infty} \ln\left(\frac{R_\infty\|\mathbf{u}\|_1 + \gamma}{R_\infty\|\mathbf{u}\|_1 - \gamma}\right)\,. \tag{24}$$

With this choice of η^*, one gets

$$k \leq \frac{\ln n}{g\left(\frac{\gamma}{R_\infty\|\mathbf{u}\|_1}\right)}\,, \tag{25}$$

where $g(\epsilon) = \frac{1+\epsilon}{2}\ln(1+\epsilon) + \frac{1-\epsilon}{2}\ln(1-\epsilon)$ (note that $\gamma/R_\infty\|\mathbf{u}\|_1 \leq 1$, since $\gamma \leq y^t(\mathbf{u}^t \cdot \mathbf{x}^t) \leq \|\mathbf{u}\|_1 R_\infty$). One can show that $g(\epsilon) \geq \epsilon^2/2$, which when applied to the above yields the desired result. □

4 Comparison of the Two Algorithms

To understand the relative strengths of the two algorithms, consider the following two examples, where $k \ll n$:

Example 1 (Sparse target vector, dense instances)**.** Let $\mathbf{u} \in \{0,1\}^n$ with at most k non-zero components, and let $\mathbf{x}^t \in \{\pm 1\}^n\ \forall t$. Thus $\|\mathbf{u}\|_1 \leq k$, $\|\mathbf{u}\|_2 \leq \sqrt{k}$, $R_2 = \sqrt{n}$, and $R_\infty = 1$.

Example 2 (Dense target vector, sparse instances)**.** Let $\mathbf{u} = \mathbf{1} \in \mathbb{R}^n$, and let $\mathbf{x}^t \in \{0,1\}^n\ \forall t$ such that each $\mathbf{x}^t$ has at most k non-zero components. Thus $\|\mathbf{u}\|_1 = n$, $\|\mathbf{u}\|_2 = \sqrt{n}$, $R_2 \leq \sqrt{k}$, and $R_\infty = 1$.

In Example 1, the mistake bound we get for perceptron is $\frac{nk}{\gamma^2}$, while that for winnow is $\frac{2k^2}{\gamma^2}\ln n$. On the other hand, in Example 2, the mistake bound we get for perceptron is $\frac{kn}{\gamma^2}$, whereas that for winnow is $\frac{2n^2}{\gamma^2}\ln n$. Thus, for a sparse target vector that depends on only a small number of relevant features, winnow gives a better mistake bound; for dense target vectors and sparse instances, perceptron has a better bound.

5 Next Lecture

In the next lecture we will see both additive and multiplicative update algorithms for online regression, and will derive bounds on their *regret*, which measures the cumulative loss of the algorithm with respect to the best possible loss within some class of predictors.

Acknowledgments. The proof of the mistake bound for winnow is based on a proof described by Sham Kakade and Ambuj Tewari in their lecture notes for a course taught at TTI Chicago in Spring 2008.

References

[1] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.