1 Non-Separable Data

The mistake bound for the Perceptron algorithm that we derived in the previous lecture assumed that the input data were linearly separable with a certain margin $\gamma$. What if this is not the case?

We begin our analysis of the non-separable case by introducing a simple reformulation of the Perceptron algorithm, called the Dual Perceptron algorithm.

The Dual Perceptron Algorithm

1. For $t \leftarrow 1$ to $T$
   - Receive $x_t$. If $\sum_{i=1}^{t-1} \alpha_i (x_i \cdot x_t) \geq 0$ then predict $+1$, otherwise predict $-1$.
   - Receive correct label $y_t$. If there is a mistake then $\alpha_t \leftarrow y_t$ otherwise $\alpha_t \leftarrow 0$.

The Dual Perceptron algorithm makes exactly the same predictions as the Perceptron algorithm on a given sequence of inputs. The only difference is that the vector $w_t$ in the Perceptron algorithm is here maintained implicitly as $\sum_{i=1}^{t-1} \alpha_i x_i$. The key insight in this simple change of perspective is that the behaviour of the Perceptron algorithm only depends on the inner products of the input data. We will make use of this observation later on when we introduce the so-called “kernel trick”. For now we use it to derive a mistake bound in the non-separable case.

Consider a set of labelled examples $(x_1, y_1), \ldots, (x_T, y_T)$ that is not necessarily separable. Fix $\gamma > 0$ (and think of this as the desired margin). We define the loss $\ell_t$ of a threshold function with unit normal vector $u$ on the $t$-th input $(x_t, y_t)$ as

$$\ell_t = \max(0, \gamma - y_t (u \cdot x_t)).$$

Notice that $\ell_t$ is non-negative and is zero if and only if the threshold function has margin at least $\gamma$ at $(x_t, y_t)$.

The cumulative loss $L$ of $u$ over the sequence $(x_1, y_1), \ldots, (x_T, y_T)$ is defined by $L^2 = \sum_{t=1}^{T} \ell_t^2$. Then $L = 0$ if and only if $u$ represents a threshold function with margin at least $\gamma$ on the above sequence. We can now state our mistake bound, which specialises to the bound in the separable case by taking $L = 0$.

**Theorem 1.** Suppose that there is a linear threshold function with squared loss $L^2$ on $(x_1, y_1), \ldots, (x_T, y_T)$, where $||x_t|| \leq D$ for all $t$. Then the Perceptron algorithm makes at most $\left( \frac{D + L}{\gamma} \right)^2$ mistakes.

**Proof:** In short, the idea of the proof is to add new features to make the data linearly separable, and then to appeal to the mistake bound in the separable case.

Fix $\Delta > 0$. We map each vector $x_t \in \mathbb{R}^n$ to a new vector $\tilde{x}_t \in \mathbb{R}^{n+T}$, with $T$ extra features, where

$$\tilde{x}_t = (x_t, 0, \ldots, 0, \Delta, 0, \ldots, 0)$$

with $\Delta$ appearing in the $t$-th extra dimension. We leave each label $y_t$ unchanged.

Notice immediately that the predictions and mistakes of the (Dual) Perceptron algorithm are unaffected by us augmenting the inputs in this way, since $x_t \cdot x_s = \tilde{x}_t \cdot \tilde{x}_s$ for $1 \leq s < t \leq T$.

Now let $u \in \mathbb{R}$ be a unit vector realising the loss $L$ on the original data. Define

$$\tilde{u} = (u, \frac{y_1 \ell_1}{\Delta}, \ldots, \frac{y_T \ell_T}{\Delta}).$$

Then $||\tilde{u}||^2 = ||u||^2 + \frac{L^2}{\Delta^2} = 1 + \frac{L^2}{\Delta^2}$. Also, we have

$$y_t(\tilde{u} \cdot \tilde{x}_t) = y_t(u \cdot x_t) + y_t^2 \ell_t$$

$$= y_t(u \cdot x_t) + \ell_t$$

$$\geq \gamma,$$

for all $t$. We conclude that the transformed sequence $(\tilde{x}_1, y_1), \ldots, (\tilde{x}_T, y_T)$ is separable with margin at least

$$\frac{\gamma}{\sqrt{1 + \frac{L^2}{\Delta^2}}}.$$ 

Since $||\tilde{x}_t||^2 = ||x_t||^2 + \Delta^2 \leq D^2 + \Delta^2$, applying the mistake bound in the separable case, we conclude the Perceptron makes at most

$$\frac{(D^2 + \Delta^2) \left(1 + \frac{L^2}{\Delta^2}\right)}{\gamma^2}$$

mistakes on the transformed (and hence the original) input.

The above expression is minimised by taking $\Delta = \sqrt{DL}$, yielding a mistake bound $\left(\frac{D+L}{\gamma}\right)^2$.

\[\square\]

### 2 The Winnow Algorithm

Recall the example of financial analysts from the last lecture. Each day $n$ financial analysts predict whether the market will rise or fall. We represent the prediction on day $t$ as a vector $x_t \in \{-1, +1\}^n$. The actual outcome is represented by $y_t \in \{-1, +1\}$. We assume that there is a subset of $k$ experts, a majority of whom always give the correct prediction. We saw that the Perceptron algorithm gave a mistake bound of $nk$ for learning a linear classifier in setting. This bound depends both on the dimension $n$ of the problem and $k$, which can be viewed as the number of “relevant” features.

We will now introduce the Winnow algorithm, which was designed to have better performance in cases in which the number of relevant features is small relative to the total number of features. Winnow looks similar to the Perceptron algorithm, but uses multiplicative weight updates instead of additive updates.

Like the Perceptron algorithm, Winnow maintains a current weight vector $w_t$ at each round $t$. In the case of the Winnow algorithm $w_t$ is always a probability distribution over the set of features. We will use $w_{t,i}$ to denote the $i$-th component of $w_t$ and $x_{t,i}$ to denote the $i$-th component of $x_t$. We again assume that the set of labels is $\{-1, +1\}$.
The Winnow Algorithm (with parameter $\eta$)

1. $w_1 \leftarrow \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$.

2. For each example $x_t$,
   - If $w_t \cdot x_t \geq 0$, then predict $+1$, else predict $-1$
   - If the algorithm makes a mistake then
     \begin{align*}
     Z_t &\leftarrow \sum_{i=1}^{n} w_{t,i} e^{\eta y_t x_{t,i}} \\
     \text{For each } i, w_{t+1,i} &\leftarrow \frac{w_{t,i} e^{\eta y_t x_{t,i}}}{Z_t}
     \end{align*}

As we can see, if we make a mistake on input $x_t$ then we increase the weights of all features $i$ for which $x_{t,i}$ has the same sign as $y_t$. (In the financial experts example, these correspond to experts that predicted correctly in round $t$.) Correspondingly, we decrease the weight of all features $i$ for which $x_{t,i}$ has a different sign to $y_t$. After this we renormalise, ensuring that $||w_{t+1}||_1 = 1$.

Before stating a mistake bound for the Winnow algorithm we first recall three norms on $\mathbb{R}^n$. Each norm corresponds to a certain notion of the length of a vector $x \in \mathbb{R}^n$:

\[
||x||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}
\]
\[
||x||_1 = \sum_{i=1}^{n} |x_i|
\]
\[
||x||_\infty = \max_{i=1,\ldots,n} |x_i|
\]

The norm $|| \cdot ||_2$ is called the \textit{Euclidean norm}. We have hitherto been referring to this simply as $|| \cdot ||$.

**Theorem 2.** Let $x_1, \ldots, x_T \in \mathbb{R}^n$ with $||x_t||_\infty \leq D$ for all $t = 1, \ldots, T$. Let $\gamma > 0$ and $u \geq 0$ be such that $||u||_1 = 1$ and $y_t(u \cdot x_t) \geq \gamma$ for all $t$. If $\eta = \gamma / D^2$ then the number of mistakes made by the Winnow algorithm is at most $2(D/\gamma)^2 \ln n$.

It might seem at first as though the mistake bound in the above theorem is strictly worse than that for the Perceptron algorithm. But notice that the meaning of $D$ and $\gamma$ has changed since we use different norms to measure $x_t$ and $u$. We can compare the Perceptron and Winnow algorithms as follows:

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>radius of inputs $x_t$</th>
<th>target vector $u$</th>
<th>updates</th>
<th>mistake bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptron</td>
<td>$</td>
<td></td>
<td>\cdot</td>
<td></td>
</tr>
<tr>
<td>Winnow</td>
<td>$</td>
<td></td>
<td>\cdot</td>
<td></td>
</tr>
</tbody>
</table>

Let’s apply Theorem 2 to the example of the financial analysts. In this case the radius of the set of input data is $D = 1$. The $k$ expert analysts define a linear classifier with normal vector

\[
u = \frac{1}{k} \left(0, \ldots, 0, 1, 0, \ldots, 0, 1, 0, \ldots, 0\right),\]

where the 0-1 vector on the right is the characteristic vector of the $k$ experts and $||u||_1 = 1$. The vector $u$ has margin $1/k$ over the set of all inputs. Applying Theorem 2, we obtain a mistake bound of $2k^2 \ln n$. This bound is better than the $kn$ bound for the Perceptron algorithm if $k$ is much smaller than $n$, i.e., if there are relatively few relevant features for the classification.
Proof of Theorem 2

The proof of Theorem 2 relies on the notion of Kullback-Liebler divergence or relative entropy $RE(p||q)$ of two probability distributions $p$ and $q$ on the same set $\{1, \ldots, n\}$, with $q_i > 0$ for $i = 1, \ldots, n$. This is defined by

$$RE(p||q) = \sum_{i=1}^{n} p_i \ln \left( \frac{p_i}{q_i} \right).$$

The Kullback-Liebler divergence plays an important role in information theory, and, as the name suggests, is closely related to the notion of entropy of a probability distribution. The function $RE(p||q)$ has some of the properties of a metric. For example, we have $RE(p||q) \geq 0$ since $-RE(p||q) = n \sum_{i=1}^{n} p_i \ln \left( \frac{q_i}{p_i} \right)$ by Jensen’s inequality, since $\ln$ is concave

$$\leq \ln \left( \sum_{i=1}^{n} p_i \cdot \frac{q_i}{p_i} \right) \leq \ln \left( \sum_{i=1}^{n} p_i \cdot q_i \right)$$

It can also be shown that $RE(p||q) = 0$ if and only if $p = q$. However in general we have neither symmetry ($RE(p||q) = RE(q||p)$), nor the triangle inequality.

Proof:(of Theorem 2) We define a potential function $\Phi_t \geq 0$ by

$$\Phi_t = \sum_{i=1}^{n} u_i \ln \left( \frac{u_i}{w_{t,i}} \right),$$

i.e., $\Phi_t$ is the relative entropy of $u$ and $w_t$. Then if we make a mistake in round $t$,

$$\Phi_{t+1} - \Phi_t = \sum_{i=1}^{n} u_i \ln \left( \frac{u_{t,i}}{w_{t+1,i}} \right) = \ln Z_t - \sum_{i=1}^{n} u_i \eta y_t x_{t,i} \leq \ln Z_t - \eta \gamma$$

We claim that $\ln Z_t \leq (\eta D)^2/2$. Given this, when $\eta = \gamma/D^2$ we have that $\Phi_{t+1} - \Phi_t \leq -\gamma^2/2D^2$ in any round $t$ in which there is a mistake. Since

$$\Phi_1 = \sum_{i=1}^{n} u_i \ln(nu_i) = \ln n + \sum_{i=1}^{n} u_i \ln(u_i) \leq \ln n,$$

the mistake bound of $2(D/\gamma)^2 \ln n$ follows.
It remains to prove the claimed upper bound on $\ln Z_t$, which is equivalent to showing that $Z_t \leq e^{(\eta D)^2/2}$.

To this end, consider a random variable $X$ that takes value $y_t x_t,i$ with probability $w_{t,i}$ for $i = 1, \ldots, n$. Then $E[X] = y_t (w_t \cdot x_t) \leq 0$ (by the assumption of a mistake in round $t$). Moreover we have $Z_t = E[e^{\eta X}] \leq E[e^{\eta Y}]$, where $Y = X - E[X]$. Now $Y$ is a random variable taking values in $[-2D, 2D]$ with $E[Y] = 0$, and so, applying Hoeffding’s Lemma\textsuperscript{1}, we have $Z_t \leq E[e^{\eta Y}] \leq e^{(\eta D)^2/2}$.

\textsuperscript{1}Recall that this states that for a random variable $Y$ taking values in the interval $[a, b]$, $E[e^{\eta Y}] \leq e^{\eta^2 (b-a)^2/2}$. 