Perceptron, again
Homogeneous linear classifiers

• Homogeneous linear classifier: \( w \in \mathbb{R}^d \) (weight vector)

\[
f_w(x) = f_{w,0}(x) = \begin{cases} 
+1, & \langle x, w \rangle > 0 \\
-1, & \langle x, w \rangle \leq 0 
\end{cases}
\]
Perceptron (Rosenblatt, ‘58)

Input: training data $S$

• Let $w_1 = 0$.
• For $t = 1, 2, ...$:
  • If there is $(x_t, y_t) \in S$ such that $f_{w_t}(x_t) \neq y_t$, then:
    • Update: $w_{t+1} := w_t + y_t x_t$
  • Else: return $w_t$

If $S$ is separable with margin $\gamma > 0$, and $R := \max_{(x, y) \in S} ||x||$, then Perceptron terminates after $\left(\frac{R}{\gamma}\right)^2$ updates with linear separator for $S$. 
Online Perceptron

**Input**: training data $S$ as an input stream.

- Let $w = \vec{0}$.
- For each $(x, y) \in S$:
  - If $f_w(x) \neq y$, then:
    - **Update**: $w := w + yx$
- Return $w$
Online Perceptron

• **Always terminates**: in fact, just makes a *single pass through the data*!
• Does it return a linear separator (assuming one exists)? Maybe not.

• **However**:

\[
\text{If } S \text{ is separable with margin } \gamma > 0, \text{ and } R := \max_{(x,y) \in S} \|x\|, \text{ then Online Perceptron makes at most } \left(\frac{R}{\gamma}\right)^2 \text{ mistakes (and updates).}
\]
What good is a mistake bound?

• **Mistake bound**: upper-bound on number of mistakes made by an *online learning algorithm* on an arbitrary sequence of examples.

• **Online learning algorithm** (for our purposes): algorithm that operates on a stream of examples, and always has a “current classifier” in hand.

• **Amazing fact**: online learning algorithms with small mistake bounds can be used to produce classifiers with small classification error!
Voted-Perceptron (Freund and Schapire, ‘99)

**Input:** training data $S$ as an *input stream*.

- Let $w_1 = 0$, $c_1 = 1$, $t = 0$.
- For each $(x, y) \in S$:
  - If $f_{w_t}(x) \neq y$, then:
    - Update: $w_{t+1} := w_t + yx$, $c_{t+1} := 1$, $t := t + 1$.
  - Else: $c_t := c_t + 1$
- Return $((w_1, c_1), (w_2, c_2), ..., (w_t, c_t))$

$c_t$ represents # of examples that $w_t$ correctly classifies (plus one).
A.K.A. “survival time”.
Voted-Perceptron (Freund and Schapire, ‘99)

What is the final classifier based on \((w_1, c_1), (w_2, c_2), \ldots, (w_t, c_t)\)?

**Input:** test point \(x\)

- **Compute score:** \(z := \sum_{s=1}^{t} c_s f_{w_s}(x)\)
- **Compute prediction:** \(\hat{y} := \text{sign}(z)\)

\(c_s\) represents # of examples that \(w_s\) correctly classifies (plus one).

A.K.A. “survival time”.
Voted-Perceptron: classification error

• Assume $S$ is a sequence of $n$ i.i.d. examples $(x, y)$ from $P$.
• Also assume there exists $w_\star$ with $\|w_\star\| = 1$ and $\gamma, R > 0$ such that
  $$\Pr_{(x,y)\sim P} (y\langle w_\star, x \rangle \geq \gamma \wedge \|x\| \leq R) = 1.$$  

• If $\hat{f}$ denote the classifier returned by Voted-Perceptron on input $S$, then:
  $$\mathbb{E}[\text{err}(\hat{f})] \leq \frac{2(R/\gamma)^2}{n + 1}$$
Other variants

• What determines final classifier?
  1. Just run Online Perceptron and return final $w$
  2. Voted-Perceptron, based on survival times $c_i$
  3. Averaged-Perceptron: \( \hat{\mathbf{w}} := \sum_{i=1}^{t} c_i \mathbf{w}_i \)

• How to use the training data?
  1. Make a single pass through $S$.
  2. Make multiple passes through $S$. 

Experimental results

- Using OCR digits data, binary classification problem of distinguishing “9” from other digits.
- # training examples: 60000 (about 6000 are from class “9”).

<table>
<thead>
<tr>
<th># passes</th>
<th>0.1</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test error (online-P)</td>
<td>0.079</td>
<td>0.064</td>
<td>0.057</td>
<td>0.063</td>
<td>0.058</td>
<td>0.059</td>
</tr>
<tr>
<td>Test error (voted-P)</td>
<td>0.045</td>
<td>0.039</td>
<td>0.038</td>
<td>0.038</td>
<td>0.038</td>
<td>0.037</td>
</tr>
<tr>
<td>Test error (average-P)</td>
<td>0.045</td>
<td>0.039</td>
<td>0.038</td>
<td>0.038</td>
<td>0.038</td>
<td>0.037</td>
</tr>
</tbody>
</table>