Example: Margin in High-Dimension

<table>
<thead>
<tr>
<th>Training Sample $S_{train}$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$x_6$</th>
<th>$x_7$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, -1, 1)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(2, -1, 1)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(2, -1, 1)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(2, -1, 1)</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hyperplane 1</th>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$w_4$</th>
<th>$w_5$</th>
<th>$w_6$</th>
<th>$w_7$</th>
<th>$b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyperplane 2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hyperplane 3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Hyperplane 4</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hyperplane 5</td>
<td>0.95</td>
<td>0.05</td>
<td>0</td>
<td>0.05</td>
<td>0.05</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hyperplane 6</td>
<td>0.67</td>
<td>-0.67</td>
<td>0</td>
<td>0.33</td>
<td>0.33</td>
<td>-0.33</td>
<td>-0.33</td>
<td>0</td>
</tr>
</tbody>
</table>

(Batch) Perceptron Algorithm

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^N$, $y_i \in \{-1, 1\}$, $f \in \{1, 2, \ldots \}$

Algorithm:
- $w_0 = \vec{0}$, $k = 0$
- repeat
  - FOR $i = 1$ TO $n$
    - IF $y_i(w \cdot x_i) \leq 0$   # makes mistake
      - $w_{k+1} = w_k + y_i x_i$
      - $e = k + 1$
      - ENDIF
      - ENDIF
  - until $f$ iterations reached

Dual (Batch) Perceptron Algorithm

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x_i \in \mathbb{R}^N$, $y_i \in \{-1, 1\}$, $f \in \{1, 2, \ldots \}$

Dual Algorithm:
- $\forall i \in [1 \ldots n] \quad \alpha_i = 0$
- repeat
  - FOR $i = 1$ TO $n$
    - IF $y_i (w \cdot x_i) \leq C$
      - $\alpha_i = \alpha_i - 1$
      - ENDIF
    - ENDIF
- until $f$ iterations reached
- $w_0 = \vec{0}$, $k = 0$
- repeat
  - FOR $i = 1$ TO $n$
    - IF $y_i (w \cdot x_i) \leq C$
      - $\alpha_i = \alpha_i + 1$
      - ENDIF
    - ENDIF
- until $f$ iterations reached

SVM Solution as Linear Combination

- Primal OP:
  - minimize: $P(w, h, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i$
  - subject to: $\forall_{i=1}^{n} : y_i (w \cdot x_i + b) \geq 1 - \xi_i$

- Theorem: The solution $w^*$ can always be written as a linear combination $w^* = \sum_{i=1}^{n} \alpha_i y_i x_i$ of the training vectors with $0 \leq \alpha_i \leq C$

- Properties:
  - Factor $\alpha_i$ indicates “influence” of training example $(x_i, y_i)$
  - If $\alpha_i > 0$, then $x_i \in C$
  - If $\alpha_i = 0$, then $x_i \notin C$
  - $(x_i, y_i)$ is a Support Vector, if and only if $\alpha_i > 0$
  - If $0 < \alpha_i < C$, then $y_i w^* \cdot x_i = 1$
  - SVM light outputs $\alpha_i$ using the “-a” option

Dual SVM Optimization Problem

- Dual Optimization Problem
  - maximize: $D(\alpha) = \frac{1}{2} \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j\alpha_i \alpha_j (x_i \cdot x_j)$
  - subject to: $\sum_{i=1}^{n} \alpha_i y_i = 0$

- Theorem: If $w^*$ is the solution of the Primal and $\alpha^*$ is the solution of the Dual, then $w^* = \sum_{i=1}^{n} \alpha^*_i y_i x_i$
Leave-One-Out (i.e. n-fold CV)
• Training Set: $S = \{(x_1,y_1), \ldots, (x_n,y_n)\}$
• Approach: Repeatedly leave one example out for testing.

<table>
<thead>
<tr>
<th>Train on</th>
<th>Test on</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)$</td>
<td>$(x_{n+1},y_{n+1})$, $(x_{n+2},y_{n+2})$</td>
</tr>
<tr>
<td>$(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)$</td>
<td>$(x_{n+1},y_{n+1})$, $(x_{n+2},y_{n+2})$</td>
</tr>
<tr>
<td>$(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)$</td>
<td>$(x_{n+1},y_{n+1})$, $(x_{n+2},y_{n+2})$</td>
</tr>
<tr>
<td>$(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)$</td>
<td>$(x_{n+1},y_{n+1})$, $(x_{n+2},y_{n+2})$</td>
</tr>
</tbody>
</table>

• Estimate: $err_{loo}(A) = \frac{1}{n} \sum_{i=1}^{n} A(h_i(x_i),y_i)$
• Question: Is there a cheaper way to compute this estimate?

Necessary Condition for Leave-One-Out Error
• Lemma: For SVM, $[h_i(x_i) \neq y_i] \Rightarrow [2 \alpha_i R^2 + \xi_i \geq 1]$
• Input:
  – $\alpha_i$: dual variable of example $i$
  – $\xi_i$: slack variable of example $i$
  – $\|x_i\| \leq R$ bound on length
• Example:

<table>
<thead>
<tr>
<th>Value of $2 \alpha_i R^2 + \xi_i$</th>
<th>Leave-one-out Error?</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Must be Correct</td>
</tr>
<tr>
<td>0.7</td>
<td>Must be Correct</td>
</tr>
<tr>
<td>3.5</td>
<td>Error</td>
</tr>
<tr>
<td>0.1</td>
<td>Must be Correct</td>
</tr>
<tr>
<td>1.3</td>
<td>Correct</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Case 1: Example is not SV
Criterion: $(\alpha_i = 0)$ and $(\xi_i = 0)$, so $(2 \alpha_i R^2 + \xi_i < 1)$: Correct

Case 2: Example is SV with Low Influence
Criterion: $(\alpha_i < 0.5/R^2 < C)$ and $(\xi_i = 0)$, so $(2 \alpha_i R^2 + \xi_i < 1)$: Correct

Case 3: Example has Small Training Error
Criterion: $(\alpha_i = C)$ and $(\xi_i < 1-2CR^2)$, so $(2 \alpha_i R^2 + \xi_i < 1)$: Correct

Experiment: Reuters Text Classification
Experiment Setup
– 6451 Training Examples
– 6451 Test Examples to estimate true Prediction Error
– Comparison between Leave-One-Out upper bound and error on Test Set (average over 10 train/test splits)