Support Vector Machines: Soft Margin and Duality

CS6780 – Advanced Machine Learning
Spring 2019
Thorsten Joachims
Cornell University

Reading: Schoelkopf/Smola Chapter 7.3, 7.5
Cristianini/Shawe-Taylor Chapter 2-2.1.1

Non-Separable Training Data

• Limitations of hard-margin formulation
  – For some training data, there is no separating hyperplane.
  – Complete separation (i.e. zero training error) can lead to suboptimal prediction error.

Soft-Margin Separation

Idea: Maximize margin and minimize training error.

Soft-Margin OP (Primal):
\[
\min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{N} \xi_i
\]
\[s.t. \ y_1 (\mathbf{w}^T \mathbf{x}_1 + b) \geq 1 - \xi_1 \wedge \xi_1 \geq 0
\]
\[y_2 (\mathbf{w}^T \mathbf{x}_2 + b) \geq 1 - \xi_2 \wedge \xi_2 \geq 0
\]

• Slack variable \( \xi_i \) measures by how much \((x_i, y_i)\) fails to achieve margin \( \delta \)
• \( \sum \xi_i \) is upper bound on number of training errors
• \( C \) is a parameter that controls trade-off between margin and training error.

Soft-Margin OP (Primal):
\[
\min_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^{N} \xi_i
\]
\[s.t. \ y_1 (\mathbf{w}^T \mathbf{x}_1 + b) \geq 1 - \xi_1 \wedge \xi_1 \geq 0
\]
\[y_2 (\mathbf{w}^T \mathbf{x}_2 + b) \geq 1 - \xi_2 \wedge \xi_2 \geq 0
\]

• \( \sum \xi_i \) is upper bound on number of training errors
• \( C \) is a parameter that controls trade-off between margin and training error.

Example Reuters “acq”: Varying C

<table>
<thead>
<tr>
<th>Training Sample</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>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training set 1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Training set 2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Test set 1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>Test set 2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Controlling Soft-Margin Separation

\( \sum \xi_i \) is upper bound on number of training errors
\( C \) is a parameter that controls trade-off between margin and training error.

Example: Margin in High-Dimension
**SVM Solution as Linear Combination**

- **Primal OP:**
  \[
  \text{minimize: } P(w, b, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\
  \text{subject to: } v_{i+1} = v_i + \xi_i, v_1 = 0, v_i \geq 0
  \]

  - **Theorem:** The solution \( \mathbf{w}^* \) can always be written as a linear combination
  \[
  \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i
  \]
  of the training vectors with \( 0 \leq \alpha_i \leq C \).

- **Properties:**
  - Factor \( \alpha_i \) indicates "influence" of training example \((x_i, y_i)\).
  - \((x_i, y_i)\) is a Support Vector, if and only if \( \alpha_i > 0 \).
  - \( \xi_i < 0 \), then \( \alpha_i = C \).
  - \( \text{If } 0 < \alpha < C, \text{ then } \xi_i = 0. \)
  - \( \text{If } 0 < \alpha < C, \text{ then } y_i(y_i \mathbf{w}^* - b) - 1 \).
  - SVM-light outputs \( \alpha_i \) using the "-a" option

**Dual SVM Optimization Problem**

- **Primal Optimization Problem**
  \[
  \text{minimize: } P(w, b, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \\
  \text{subject to: } v_{i+1} = v_i + \xi_i, v_1 = 0, v_i \geq 0
  \]

- **Dual Optimization Problem**
  \[
  \text{maximize: } D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\
  \text{subject to: } \sum_{i=1}^{n} \alpha_i = 0, v_{i+1} = 0, 0 \leq \alpha_i \leq C
  \]

- **Theorem:** If \( \mathbf{w}^* \) is the solution of the Primal and \( \alpha^* \) is the solution of the Dual, then
  \[
  \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i
  \]

**Leave-One-Out (i.e. n-fold CV)**

- **Training Set:** \( S = \{(x_1, y_1), ..., (x_n, y_n)\} \)
- **Approach:** Repeatedly leave one example out for testing.

\[
\begin{array}{c|c|c}
\text{Train on} & \text{Test on} & \text{Leave-one-out Error} \\
\hline
(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_i, y_i), (x_{i+1}, y_{i+1}), ..., (x_n, y_n) & (x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_{i-1}, y_{i-1}), (x_{i+1}, y_{i+1}), ..., (x_n, y_n) & \hline
\end{array}
\]

\( h_i \) is the rule learned on \( S \setminus \{(x_i, y_i)\} \)

- **Estimator:** \( \text{Err}_{\text{leave-out}}(A) = \frac{1}{n} \sum_{i=1}^{n} d(h_i(x_i), y_i) \)

- **Question:** Is there a cheaper way to compute this estimate?

**Necessary Condition for Leave-One-Out Error**

- **Lemma:** For SVM, \( |h_i(\mathbf{x}_i) - y_i| = |2\alpha_i R^2 + \xi_i - 1| \)

- **Input:**
  - \( \alpha_i \) dual variable of example \( i \)
  - \( \xi_i \) slack variable of example \( i \)
  - \( \|\mathbf{x}_i\| \leq R \) bound on length

- **Example:**

\[
\begin{array}{c|c|c}
\text{Value of } 2\alpha_i R^2 + \xi_i & \text{Leave-one-out Error} \\
\hline
0.0 & \text{Must be Correct} \\
0.7 & \text{Must be Correct} \\
1.5 & \text{Error} \\
0.1 & \text{Must be Correct} \\
1.2 & \text{Correct} \\
\end{array}
\]
Case 1: Example is not SV
Criterion: \((\alpha_i = 0) \rightarrow (\xi_i = 0) \rightarrow (2 \alpha_i R^2 + \xi_i < 1) \rightarrow Correct\)

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

Case 3: Example has Small Training Error
Criterion: \((\alpha_i = C) \text{ and } (\xi_i < 1-2CR^2) \rightarrow (2 \alpha_i R^2 + \xi_i < 1) \rightarrow Correct\)

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)

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Retraining Steps (%)</th>
<th>CPU-Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuters (n=6451)</td>
<td>0.58%</td>
<td>32.3</td>
</tr>
<tr>
<td>WebKB (n=2092)</td>
<td>20.42%</td>
<td>235.4</td>
</tr>
<tr>
<td>Ohsumed (n=10000)</td>
<td>2.56%</td>
<td>1132.3</td>
</tr>
</tbody>
</table>

Fast Leave-One-Out Estimation for SVMs
Lemma: Training errors are always Leave-One-Out Errors.
Algorithm:
- \((R,\alpha,\xi) = \text{trainSVM}(S_{train})\)
- FOR \((x,y) \in S_{train}\)
  - IF \(\xi > 1\) THEN loo++;
  - ELSE IF \((2 \alpha R^2 + \xi < 1)\) THEN loo = loo;
  - ELSE trainSVM\((S_{train} \setminus \{(x,y)\})\) and test explicitly

Experiment:
<table>
<thead>
<tr>
<th>Training Data</th>
<th>Retraining Steps (%)</th>
<th>CPU-Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reuters (n=6451)</td>
<td>0.58%</td>
<td>32.3</td>
</tr>
<tr>
<td>WebKB (n=2092)</td>
<td>20.42%</td>
<td>235.4</td>
</tr>
<tr>
<td>Ohsumed (n=10000)</td>
<td>2.56%</td>
<td>1132.3</td>
</tr>
</tbody>
</table>