Learning From Data
Lecture 24
The Optimal Hyperplane and Overfitting

Why is the fattest hyperplane the best?
Non-separable Data

M. Magdon-Ismail
CSCI 4100/6100

The Optimal Hyperplane
The fattest hyperplane that separates the data tolerates most measurement error.

Quadratic Programming:
minimize $b, w$
subject to:
$y_n(w^T x_n + b) \geq 1$ for $n = 1, \ldots, N$.

Support vectors: the data points that sit on the cushion.
Using only support vectors, the classifier does not change.

Evidence that Larger Margin is Better

(1) Experimental: larger margin gives lower $E_{out}$; bias drops a little and var a lot.

(2) Bound for $d_{vc}$ can be less than $d + 1$ - fat hyperplanes generalize better.

(3) $E_{cv}$ bound does not explicitly depend on $d$.

The optimal hyperplane performs ‘automatic’ regularization.

Overfitting and the Optimal Hyperplane: 4/17
Larger Margin is Better

Generate a random separable data set \((N = 20)\)

Select 50,000 random separating hyperplanes \(h\)

Compute \(E_{\text{out}}\) and \(\rho(h)/\rho(\text{SVM})\)

Average over several thousands of random data sets

Bigger margin is generally better

Biggest is not best.

Bias and Variance

Bias and Variance experiment

\[ \begin{array}{ccc}
\text{Random} & \text{SVM} \\
\text{bias} & 0.02 & 0.015 \\
\text{var} & 0.059 & 0.038 \\
E_{\text{out}} & 0.079 & 0.053 \\
\end{array} \]

\(\rho\) (Random hyperplane) / \(\rho\) (SVM)

Fat Hyperplanes Shatter Fewer Points

Theorem.
\[ d_{vc}(\gamma) \leq \left\lceil \frac{R^2}{\gamma^2} \right\rceil + 1 \]

Fat hyperplanes can’t shatter.

Theorem on \(d_{vc}(\gamma)\)
Fat Hyperplanes Shatter Fewer Points

Theorem. \( d_{vc}(\gamma) \leq \left\lceil \frac{R^2}{\gamma^2} \right\rceil + 1 \)

A Bound on \( E_{cv} \)

\[ E_{cv} = \frac{1}{N} \sum_{n=1}^{N} e_n \] (unbiased estimate of \( E_{out}(N-1) \))

\[ E_{cv} \leq \frac{\# \text{ support vectors}}{N} \] (no explicit dependence on \( d \))

Summary of Hyperplanes and Generalization

<table>
<thead>
<tr>
<th>Algorithm For Selecting Separating Hyperplane</th>
<th>General</th>
<th>PLA</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_{vc} = d + 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( E_{cv} \leq \frac{R^2}{N\rho^2} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( d_{vc}(\rho) \leq \left\lceil \frac{\rho^2}{\gamma^2} \right\rceil + 1 )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( E_{cv} \leq \frac{# \text{ support vectors}}{N} )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Generalization performance controlled by quantities not explicitly depending on \( d \)

Non-Separable Data

tolerate error

nonlinear transform
Soft Margin SVM

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \\
\text{subject to} & \quad y_n (w^T x_n + b) \geq 1 - \xi_n \\
& \quad \xi_n \geq 0 \quad \text{for } n = 1, \ldots, N
\end{align*}
\]

Tolerate error

Trades off ‘soft in-sample error’ \( \sum_{n=1}^{N} \xi_n \) with weight norm \( \frac{1}{2} w^T w \)

\( C \) plays the role of a regularization parameter (\( \lambda \sim \frac{1}{C} \))

Choice of \( C \) is important - similar to choice of \( \lambda \) in regularization

Non-Separable Data

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n \\
\text{subject to} & \quad y_n (w^T x_n + b) \geq 1 - \xi_n \\
& \quad \xi_n \geq 0 \quad \text{for } n = 1, \ldots, N
\end{align*}
\]

\( C = 1 \)

\( C = 500 \)

Nonlinear Transform and SVM

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \Phi^T \Phi w + C \sum_{n=1}^{N} \xi_n \\
\text{subject to} & \quad y_n (\Phi^T \Phi x_n + b) \geq 1 - \xi_n \\
& \quad \xi_n \geq 0 \quad \text{for } n = 1, \ldots, N
\end{align*}
\]

\( \Phi_2 + \text{SVM} \)

\( \Phi_3 + \text{SVM} \)

\( \Phi_3 + \text{pseudoinverse algorithm} \)

Observations:
1. \( \Phi_2 \) has almost 2 times the parameters of \( \Phi_3 \)
2. \( \Phi_2 \)-SVM does not display significant overfitting compared to \( \Phi_3 \)-regression
3. \#support vectors did not double
4. Can go to higher dimensions if \#support vectors stays small or margin stays large
Going to Even Higher Dimension

In higher dimension, can control overfitting with \# support vectors or margin $\rho$

What about:
  
  Efficiency?
  
  Infinitely many dimensions?