How to prevent over fitting? II

- Regularization: penalize large coefficient values

\[
\tilde{E}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \{ f(x_i, \mathbf{w}) - y_i \}^2 + \frac{\lambda}{2} \| \mathbf{w} \|^2
\]

“ridge” regression

- In practice use validation data to choose lambda (not test data)
- Convex cost function, closed form solution
### Polynomial Coefficients

<table>
<thead>
<tr>
<th>$w^*_0$</th>
<th>$\ln \lambda = -\infty$</th>
<th>$\ln \lambda = -18$</th>
<th>$\ln \lambda = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.35</td>
<td>0.35</td>
<td>0.13</td>
</tr>
<tr>
<td>$w^*_1$</td>
<td>232.37</td>
<td>4.74</td>
<td>-0.05</td>
</tr>
<tr>
<td>$w^*_2$</td>
<td>-5321.83</td>
<td>-0.77</td>
<td>-0.06</td>
</tr>
<tr>
<td>$w^*_3$</td>
<td>48568.31</td>
<td>-31.97</td>
<td>-0.05</td>
</tr>
<tr>
<td>$w^*_4$</td>
<td>-231639.30</td>
<td>-3.89</td>
<td>-0.03</td>
</tr>
<tr>
<td>$w^*_5$</td>
<td>640042.26</td>
<td>55.28</td>
<td>-0.02</td>
</tr>
<tr>
<td>$w^*_6$</td>
<td>-1061800.52</td>
<td>41.32</td>
<td>-0.01</td>
</tr>
<tr>
<td>$w^*_7$</td>
<td>1042400.18</td>
<td>-45.95</td>
<td>0.00</td>
</tr>
<tr>
<td>$w^*_8$</td>
<td>-557682.99</td>
<td>-91.53</td>
<td>0.00</td>
</tr>
<tr>
<td>$w^*_9$</td>
<td>125201.43</td>
<td>72.68</td>
<td>0.01</td>
</tr>
</tbody>
</table>
Summary Point: How to set parameters?

Use a validation set

Divide the total dataset into three subsets:

- **Training data** is used for learning the parameters of the model.

- **Validation data** is not used for learning, but is used to determine the hyper-parameters, e.g. for deciding the type of model and the amount of regularization.

- **Test data** is used to get a final, unbiased estimate of how well the learning machine works. Expect this estimate to be worse than on the validation data.

Can re-divide the total dataset to get another unbiased estimate of the true error rate.
• Again, need to control the complexity of the (discriminant) function
What comes next?

• Learning by optimizing a cost function:

\[ \tilde{E}(w) = \frac{1}{2} \sum_{i=1}^{N} \left( f(x_i, w) - y_i \right)^2 + \frac{\lambda}{2} \|w\|^2 \]

loss function    regularization

• In general Minimize with respect to \( f \in \mathcal{F} \)

\[ \sum_{i=1}^{N} l(f(x_i), y_i) + \lambda R(f) \]

• choose loss function for: classification, regression, ranking, clustering …
• choose regularization function
The “Lasso” or $L_1$ norm regularization

- LASSO = Least Absolute Shrinkage and Selection

Minimize with respect to $w \in \mathbb{R}^d$

$$\sum_{i=1}^{N} (y_i - f(x_i, w))^2 + \lambda \sum_{j}^{d} |w_j|$$

- This is a quadratic optimization problem
- There is a unique solution
- $p$-Norm definition: $\|w\|_p = \left( \sum_{j=1}^{d} |w_j|^p \right)^{\frac{1}{p}}$
Sparsity property of the Lasso

- contour plots for $d = 2$

\[ \sum_{i=1}^{N} (y_i - f(x_i, w))^2 \]

- Minimum where contours of loss and regularizer are tangent

- For the lasso case, minima occur at “corners”

- Consequently one of the weights is zero

- In high dimensions many weights can be zero
Lasso in action

Ridge Regression

L1-Regularized Least Squares

percent of lambdaMax

coefficient values

percent of lambdaMax

regularization parameter lambda
Sparse weight vectors

• Weights being zero is a method of “feature selection” – zeroing out the unimportant features

• (The SVM classifier also has this property (sparse alpha in the dual representation))

• Ridge regression does not
More loss functions for regression

- **quadratic (square) loss** $\ell(y, f(x)) = \frac{1}{2}(y - f(x))^2$

- **$\varepsilon$-insensitive loss** $\ell(y, f(x)) = \max((|r| - \varepsilon), 0)$

- **Hüber loss** (mixed quadratic/linear): robustness to outliers:
  $\ell(y, f(x)) = h(y - f(x))$
  $h(r) = \begin{cases} r^2 & \text{if } |r| \leq c \\ 2c|r| - c^2 & \text{otherwise.} \end{cases}$

- All of these are convex.
Part II Outline

- Linear classifiers
  - Linear separability

- Support Vector Machine (SVM) classifier
  - Wide margin
  - Cost function
  - Hard and soft margins
  - Optimization

- Logistic Regression classifier
  - Cost function
Linear Classifiers and the SVM
Binary Classification

Given training data \((x_i, y_i)\) for \(i = 1 \ldots N\), with \(x_i \in \mathbb{R}^d\) and \(y_i \in \{-1, 1\}\), learn a classifier \(f(x)\) such that

\[
f(x_i) \begin{cases} 
  \geq 0 & y_i = +1 \\
  < 0 & y_i = -1
\end{cases}
\]

i.e. \(y_i f(x_i) > 0\) for a correct classification.
Linear separability

linearly separable

not linearly separable
A linear classifier has the form

\[ f(x) = w^\top x + b \]

- in 2D the discriminant is a line
- \( w \) is the normal to the line, and \( b \) the bias
- \( w \) is known as the weight vector
A linear classifier has the form

\[ f(x) = w^\top x + b \]

- in 3D the discriminant is a plane, and in nD it is a hyperplane

For a K-NN classifier it was necessary to `carry’ the training data
For a linear classifier, the training data is used to learn \( w \) and then discarded
Only \( w \) is needed for classifying new data
What is the best $w$?

- maximum margin solution: most stable under perturbations of the inputs
How to find the maximum margin?

linearly separable data
Support Vector Machine

linearly separable data

\[ f(x) = \sum_{i} \alpha_i y_i (x_i^T x) + b \]

\[ w^T x + b = 0 \]
How to find the maximum margin?

linearly separable data
SVM – sketch derivation

- Since \( w^T x + b = 0 \) and \( c(w^T x + b) = 0 \) define the same plane, we have the freedom to choose the normalization of \( w \)

- Choose normalization such that \( w^T x_+ + b = +1 \) and \( w^T x_- + b = -1 \) for the positive and negative support vectors respectively

- Then the margin is given by

\[
\frac{w}{||w||} \cdot (x_+ - x_-) = \frac{w^T (x_+ - x_-)}{||w||} = \frac{2}{||w||}
\]
Support Vector Machine

linearly separable data

Margin = \frac{2}{||w||}

\begin{align*}
\mathbf{w}^T \mathbf{x} + b &= 1 \\
\mathbf{w}^T \mathbf{x} + b &= 0 \\
\mathbf{w}^T \mathbf{x} + b &= -1
\end{align*}
SVM – Optimization

- Learning the SVM can be formulated as an optimization:

\[
\max_w \frac{2}{\|w\|} \quad \text{subject to} \quad w^\top x_i + b \begin{cases} \geq 1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{cases} \quad \text{for } i = 1 \ldots N
\]

- Or equivalently

\[
\min_w \|w\|^2 \quad \text{subject to} \quad y_i (w^\top x_i + b) \geq 1 \quad \text{for } i = 1 \ldots N
\]

- This is a quadratic optimization problem subject to linear constraints and there is a unique minimum

Write this more concisely as

\[
\min_w \|w\|^2 \quad \text{subject to} \quad y_i f(x_i) \geq 1 \quad \text{for } i = 1 \ldots N
\]

where \( f(x) = w^\top x + b \).
Linear separability again: What is the best $w$?

- the points can be linearly separated but there is a very narrow margin

- but possibly the large margin solution is better, even though one constraint is violated

In general there is a trade off between the margin and the number of mistakes on the training data