Optimal Separating Hyperplane and the Support Vector Machine

Volker Tresp
Summer 2016
(Vapnik’s) Optimal Separating Hyperplane

- Let’s consider a linear classifier with $y_i \in \{-1, 1\}$
- If classes are linearly separable, the separating plane can be found
- Among all solutions one chooses the one that maximizes the margin $C$
Optimal Separating Hyperplane (2D)
Cost Function with Constraints

• Thus we want to find a classifier

\[
\hat{y}_i = \text{sign}(h_i)
\]

with

\[
h_i = \sum_{j=0}^{M-1} w_j x_{i,j}
\]

• The following inequality constraints need to be fulfilled at the solution

\[
y_i h_i \geq 1 \quad i = 1, \ldots, N
\]
Maximizing the Margin

• Of all possible solutions, one chooses the one that maximizes the margin

• This can be achieved by finding the solution where the sum of the squares of the weights is minimal

\[ w_{opt} = \arg \min_w \tilde{w}^T \tilde{w} = \arg \min_w \sum_{j=1}^{M-1} w_j^2 \]

where \( \tilde{w} = (w_1, \ldots, w_{M-1}) \). (this means that in \( \tilde{w} \) the offset \( w_0 \) is missing);

\( y_i \in \{-1, 1\} \)
Margin and Support-Vectors

- The margin becomes

\[ C = \frac{1}{||\tilde{w}_{opt}||} \]

- For the support vectors we have,

\[ y_i (x_i^T w_{opt}) = 1 \]
Optimal Separating Hyperplane (1D)

SVM

class 1

class 2

C
Optimization Problem

• The optimization problem is maximize

\[ \tilde{w}^T \tilde{w} \]

under the constraint that \( \forall i \)

\[ 1 - y_i(x_i^T w) \leq 0 \]

• We get the Lagrangian

\[ L_P = \frac{1}{2} \tilde{w}^T \tilde{w} + \sum_{i=1}^{N} \mu_i [1 - y_i(x_i^T w)] \]

• The Lagrangian is minimized with respect to \( w \) and maximized with respect to \( \mu_i \geq 0 \) (saddle point solution)
Solution

- The problem is solved via the Wolfe Dual and the solution can be written as

\[ \tilde{w} = \sum_{i=1}^{N} y_i \mu_i \tilde{x}_i = \sum_{i \in SV} y_i \mu_i \tilde{x}_i \]

Thus the sum is over the terms where the Lagrange multiplier is not zero, i.e., the support vectors.

- Also we can write the solution

\[ h(x) = w_0 + \sum_{j=1}^{M-1} w_j x_j = w_0 + \tilde{x}^T \tilde{w} = w_0 + \tilde{x}^T \sum_{i \in SV} y_i \mu_i \tilde{x}_i \]

\[ = w_0 + \sum_{i \in SV} y_i \mu_i \tilde{x}_i^T \tilde{x}_i = w_0 + \sum_{i \in SV} y_i \mu_i k(x_i, x) \]

Thus we get immediately a kernel solution with \( k(x_i, x) = \tilde{x}_i^T \tilde{x}_i \).
• The solution can be written as a weighted sum over support vector kernels!

• Naturally, if one works with basis functions, one gets

\[ k(x, x_i) = \phi(x)^T \phi(x_i) \]
Non-separable Classes

• If classes are not linearly separable one needs to extend the approach. On introduces the so-called slack variables $\xi_i$

• Find

$$w_{opt} = \arg \min_w \tilde{w}^T \tilde{w}$$

under the constraint that

$$y_i (x_i^T w) \geq 1 - \xi_i \quad i = 1, \ldots, N$$

and

$$\xi_i \geq 0 \quad \text{where} \quad \sum_{i=1}^{N} \xi_i \leq 1/\gamma$$

• The smaller $\gamma > 0$, the more slack is permitted. For $\gamma \to \infty$, one obtains the hard constraint
Optimization

- The optimal separating hyperplane is found via an evolved optimization of the quadratic cost function with linear constraints

- $\gamma$ is a hyperparameter
Optimization via Penalty Method

- We define

$$\arg \min_w 2\gamma \sum |1 - y_i(x_i^T w)|_+ + \tilde{w}^T \tilde{w}$$

where \(\sum |1 - y_i(x_i^T w)|_+\) is the penalty term.

Here, \(|arg|_+ = \max(arg, 0)|.

- With a finite \(\gamma\), slack is permitted.
Comparison of Cost Functions

- We consider a data point of class 1 and the contribution of one data point to the cost function/negative log-likelihood

- The contribution of $x_i$ to the cost is:
  - Least squares (blue): $\text{cost}(x_i, y_i = 1) = (1 - h(x_i))^2$
  - Perceptron (black) $\text{cost}(x_i, y_i = 1) = | - h(x_i) |$
  - Vapnik’s optimal hyperplane (green): $\text{cost}(x_i, y_i = 1) = \gamma | 1 - h(x_i) |$
  - Logistic Regression (magenta): $\text{cost}(x_i, y_i = 1) = \log(1 + \exp(-h(x_i)))$
  - Neural Network (red): $\text{cost}(x_i, y_i = 1) = (1 - \text{sig}(h(x_i)))^2$
\[ \text{cost}(x_i, y_i = 1) \]

\[ h(x_i) \]
Toy Example

- Data for two classes (red, green) are generated
- Classes overlap
- The true class boundary is in violet
- The continuous line is the separating hyperplane found by the linear SVM
- All data points within the dotted region are support vectors (62% of all data points)
- $\gamma$ is large (little slack is permitted)
Training Error: 0.270
Test Error: 0.288
Bayes Error: 0.210

$\gamma = 10000$
Toy Example (cont’d)

• Linear SVM with small $\gamma$: the solution has many more support vectors (85% of all data points)

• The test error is almost the same
\[ \gamma = 0.01 \]
Toy Example (cont’d)

• With polynomial kernels

• The test error is reduced since the fit is better

• Note that although the support vectors are close to the separating plane in the basis function space, this is not necessarily true in input space
SVM - Degree-4 Polynomial in Feature Space

Training Error: 0.180
Test Error: 0.245
Bayes Error: 0.210
Toy Example (cont’d)

• Gaussian kernels give the best results

• Most data points are support vectors
SVM - Radial Kernel in Feature Space

Training Error: 0.160
Test Error: 0.218
Bayes Error: 0.210
• The ideas of searching for solutions with a large margin has been extended to many other problems
Optimal Separating Hyperplane and Perceptron

• Vapnik was interested in solutions which focus on the currently misclassified examples such as the Perceptron. He could preserve this idea also in the Optimal Separating Hyperplanes

• The Perceptron requires

$$y_i h_i \geq 0 \quad i = 1, \ldots, N$$

which leads to non-unique solutions (even when weight decay is added)

• The important insight of Vapnik was that requiring

$$y_i h_i \geq 1 \quad i = 1, \ldots, N$$

leads to unique solutions when weight decay is added