Support vector machines and large margin classification

Chapter e-8
Which hyperplane is better?

As you can see, many lines separate the data and the Perceptron Learning Algorithm (PLA) finds one of them. Do we care about which one PLA finds? All separators have $E_{in} = 0$, so the analysis in Chapter 2 gives the same $E_{out}$-bound for every separator. Well, the VC bound may say one thing, but surely our intuition says that the rightmost separator is preferred.

Let's try to pin down an argument that supports our intuition. In practice, there are measurement errors – noise. Place identical shaded regions around each data point, with the radius of the region being the amount possible measurement error. The true data point can lie anywhere within this 'region of uncertainty' on account of the measurement error. A separator is 'safe' with respect to the measurement error if it classifies the true data points correctly. That is, no matter where in its region of uncertainty the true data point lies, it is still on the correct side of the separator. The figure below shows the largest measurement errors which are safe for each separator.

As a separator that tolerate more measurement error is safer. The rightmost separator tolerates the largest error, whereas for the leftmost separator, even a small error in some data points could result in a misclassification. In Chapter 4, we saw that noise (for example measurement error) is the main cause of overfitting. Regularization helps us combat noise and avoid overfitting. In our example, the rightmost separator is more robust to noise without compromising $E_{in}$; it is better 'regularized'. Our intuition is justifiably.

We can also quantify noise tolerance from the viewpoint of these separators. Place a cushion on each side of the separator. We call such a separator with a cushion 'fat', and we see that its separation the data if a point lies within its cushion. Here is the largest cushion we can place around each of our three candidate separators.
Large margin classifiers

Perceptron: find hyperplane that separates the two classes

Support Vector Machine (SVM): separating hyperplane with a large margin

Intuitive concept that is backed by theoretical results (statistical learning theory)

Has its origins in the work of Vladimir Vapnik

The history of SVMs

Large margin linear classifiers


Large margin non-linear classifiers


SVMs for non-separable data


Since then - lots of other large margin algorithms
Bring back the bias

Before:

\[ \mathbf{x} \in \{1\} \times \mathbb{R}^d; \quad \mathbf{w} \in \mathbb{R}^{d+1} \]

\[ \mathbf{x} = \begin{bmatrix} 1 \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_0 \\ \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_d \end{bmatrix}. \]

signal = \mathbf{w}^T \mathbf{x}

Now:

\[ \mathbf{x} \in \mathbb{R}^d; \quad b \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^d \]

\[ \mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_d \end{bmatrix}; \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_d \end{bmatrix}. \]

bias \( b \)

signal = \mathbf{w}^T \mathbf{x} + b
The geometric margin

The margin of a linear discriminant:

\[
\frac{1}{2} \hat{w}^T (x_\oplus - x_\ominus)
\]

\(\hat{w}\) a unit vector in the direction of \(w\)
The geometric margin

Want to find:

\[ \frac{1}{2} \hat{w}^T (x_+ - x_-) \]

Suppose that \( x_+ \) and \( x_- \) are equidistant from the decision boundary:

\[ w^T x_+ + b = a \]
\[ w^T x_- + b = -a \]

Subtracting the two equations:

\[ w^T (x_+ - x_-) = 2a \]

Divide by the norm of \( w \):

\[ \hat{w}^T (x_+ - x_-) = \frac{2a}{||w||} \]
Canonical separating hyperplane

Hyperplane $h = (b, \mathbf{w})$

$h$ separates the data means:

$$y_n(\mathbf{w}^T \mathbf{x}_n + b) > 0$$

By rescaling the weights and bias,

$$\min_{n=1,\ldots,N} y_n(\mathbf{w}^T \mathbf{x}_n + b) = 1$$
The geometric margin

To get a well-defined value we will use the canonical representation of a hyperplane.

Under this assumption we have that the margin equals

$$\frac{1}{\|w\|}$$

Maximizing the margin is therefore equivalent to minimizing

$$\|w\|^2$$
Motivation

Theoretical motivation: The VC dimension, which measures the complexity of a hypothesis, increases with decreasing margin.
Linear SVMs

**Objective:** maximize the margin while correctly classifying all examples correctly

\[
\min_{\mathbf{w}, b} \frac{1}{2} ||\mathbf{w}||^2
\]

subject to: \( y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 \quad i = 1, \ldots, n. \)
Digression: constrained optimization

Before considering optimization problems with inequality constraints we will consider ones with equality constraints:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to:} & \quad g_i(x) = 0
\end{align*}
\]

And to make things even simpler, start with the case of a single constraint \( g(x) \)

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to:} & \quad g(x) = 0
\end{align*}
\]
Digression: constrained optimization

Images from http://en.wikipedia.org/wiki/Lagrange_multiplier
Digression: constrained optimization

Claim: A minimizer $x^*$ of the constrained optimization problem must have the property that $\nabla f(x^*)$ is orthogonal to the constraint surface.

Therefore there exists $\lambda \neq 0$ such that

$$\nabla f(x^*) + \lambda \nabla g(x^*) = 0$$

$\lambda$ is known as a Lagrange multiplier.

Lagrange multipliers

When there are multiple equality constraints:

$$\nabla f(x^*) + \sum_i \lambda_i \nabla g_i(x^*) = 0$$

The Lagrangian function:

$$\Lambda(x, \lambda) = f(x) + \sum_i \lambda_i g_i(x) = f(x) + \lambda^T g(x)$$

The above condition is obtained by setting

$$\nabla_x \Lambda(x, \lambda) = 0$$

And the condition $$\nabla_\lambda \Lambda(x, \lambda) = 0$$
leads to the constraint equations.

Conclusion: the solution is a stationary point of the Lagrangian.
Inequality constraints

minimize \( f(x) \)
subject to: \( g(x) \leq 0 \)

Two possible scenarios:
- \( g(x) < 0 \) - the constraint is inactive
- \( g(x) = 0 \) - the constraint is active

If the constraint is inactive the stationarity condition is \( \nabla f(x) = 0 \)
This corresponds to a stationary point of the Lagrangian with \( \lambda = 0 \)
When the constraint is active, we have \( \lambda \neq 0 \)
Both cases can be summarized by the condition
\[
\lambda g(x) = 0
\]

The sign of \( \lambda \) is important: \( f(x) \) will be minimized only if its gradient is oriented away from the region \( g(x) < 0 \), i.e.
\[
\nabla f(x^*) = -\lambda \nabla g(x^*) \text{ where } \lambda > 0
\]
Conclusion:

Our constrained optimization problem of minimizing $f(x)$ such that $g(x) \leq 0$ is solved by $x, \lambda$ that satisfy:

$$\nabla \Lambda(x, \lambda) = 0$$

$$g(x) \leq 0$$

$$\lambda \geq 0$$

$$\lambda g(x) = 0$$

These are known as the KKT conditions
Constrained optimization with inequality constraints

With multiple constraints:

Our constrained optimization problem of minimizing \( f(x) \) such that \( g_i(x) \leq 0 \) is solved by \( x, \lambda \) that satisfy:

\[
\nabla \Lambda(x, \lambda) = 0 \\
g_i(x) \leq 0 \\
\lambda \geq 0 \\
\lambda_i g_i(x) = 0
\]

These are known as the KKT conditions
Lagrangian duality

Claim: The problem of minimizing $f(x)$ s.t. $g_i(x) \leq 0$ can be expressed as:

$$\min_x \max_\lambda \Lambda(x, \lambda) \text{ such that } \lambda \geq 0$$

We can see this by performing the inner maximization:

$$\max_\lambda f(x) + \lambda^T g(x) = \begin{cases} f(x) & g(x) \leq 0 \\ \infty & g(x) > 0 \end{cases}$$

Solution is a saddle point
Lagrangian duality

Claim: The problem of minimizing $f(x)$ s.t. $g_i(x) \leq 0$ can be expressed as:

$$\min_x \max_{\lambda} \Lambda(x, \lambda) \text{ such that } \lambda \geq 0$$

Instead of using the primal formulation let’s consider:

$$\max_{\lambda} \min_x \Lambda(x, \lambda) \text{ such that } \lambda \geq 0$$

This is called the dual

Under certain conditions (convexity) the two problems have the same solution
Lagrangian for the SVM problem:

\[ \Lambda(w, b, \alpha) = \frac{1}{2} ||w||^2 + \sum_{i=1}^{n} \alpha_i [1 - y_i (w^T x_i + b)] \]

Necessary conditions for the saddle point:

\[ \frac{\partial \Lambda}{\partial w} = w + \sum_{i=1}^{n} \alpha_i (-y_i x_i) = 0 \]

\[ \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \]

\[ \frac{\partial \Lambda}{\partial b} = \sum_{i=1}^{n} \alpha_i y_i = 0 \]

How do we get \( b \)?
Support Vectors

Let's use the KKT conditions:

$$\alpha_i [1 - y_i (w^T x_i + b)] = 0$$

Implication:

Pick an $i$ such that $\alpha_i > 0$

$$y_i (w^T x_i + b) = 1$$

$$\Rightarrow \quad b = y_i - w^T x_i$$
Support Vectors

Let’s use the KKT conditions:

\[ \alpha_i \left[ 1 - y_i (\mathbf{w}^T \mathbf{x}_i + b) \right] = 0 \]

Implication:
Pick an \( i \) such that \( \alpha_i > 0 \)

\[ y_i (\mathbf{w}^T \mathbf{x}_i + b) = 1 \]

\[ \Rightarrow b = y_i - \mathbf{w}^T \mathbf{x}_i \]

The correspond \( x_i \) are called support vectors
Support Vectors

Claim: The fraction of support vectors is an upper bound on the estimated Leave-One-Out error (see page 17 in chapter 8)

\[ E_{cv}(SVM) = \frac{1}{N} \sum_{n=1}^{N} e_n \leq \frac{\# \text{ support vectors}}{N}. \]
The dual

\[ \Lambda(w, b, \alpha) = \frac{1}{2} ||w||^2 + \sum_{i=1}^{n} \alpha_i \left[ 1 - y_i (w^T x_i + b) \right] \]

The dual:

\[ W(\alpha) = \frac{1}{2} \left( \sum_{i=1}^{n} \alpha_i y_i x_i \right)^T \sum_{j=1}^{n} \alpha_j y_j x_j \]

\[ + \sum_{i=1}^{n} \alpha_i - b \sum_{i=1}^{n} \alpha_i y_i \]

\[ - \sum_{i=1}^{n} \alpha_i y_i \left( \sum_{j=1}^{n} \alpha_j y_j x_j^T x_i \right) \]
The dual

\[
W(\alpha) = \frac{1}{2} \left( \sum_{i=1}^{n} \alpha_i y_i x_i \right)^{\top} \sum_{j=1}^{n} \alpha_j y_j x_j \\
+ \sum_{i=1}^{n} \alpha_i - b \sum_{i=1}^{n} \alpha_i y_i \\
- \sum_{i=1}^{n} \alpha_i y_i \left( \sum_{j=1}^{n} \alpha_j y_j x_j^{\top} x_i \right) \\
= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^{\top} x_j
\]
The dual

\[
\begin{align*}
\text{maximize} & \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{subject to:} & \quad \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i = 0
\end{align*}
\]

Comments: quadratic programming problem (no local minima!) Usually a sparse solution (many alphas equal to 0)

Compare to the primal:

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|^2 \\
\text{subject to:} & \quad y_i (w^T x_i + b) \geq 1 \quad i = 1, \ldots, n.
\end{align*}
\]
The non-seaprable case

In order to allow for misclassifications we replace the constraints

$$y_i (w^T x_i + b) \geq 1$$

with

$$y_i (w^T x_i + b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$ are called slack variables

Need to incorporate the slack variables in the optimization problem because we want to discourage overuse of the slacks.

$$\sum_{i=1}^{n} \xi_i$$ is a bound on the number of misclassified examples
SVMs for non-separable data

Our optimization problem for the non-separable case:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i$$

subject to: $y_i(w^T x_i + b) \geq 1 - \xi_i, \; \xi_i \geq 0, \; i = 1, \ldots, n.$
SVMs for non-separable data

Our optimization problem for the non-separable case:

$$\min_{\mathbf{w}, b} \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i$$

subject to:  $$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \, \xi_i \geq 0, \, \, i = 1, \ldots, n.$$ 

Let's form the Lagrangian:

$$\Lambda(\mathbf{w}, b, \alpha, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i [1 - \xi_i - y_i (\mathbf{w}^T \mathbf{x}_i + b)] + \sum_{i=1}^{n} \beta_i \xi_i$$

Saddle point equations:

$$\frac{\partial \Lambda}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0$$

$$\frac{\partial \Lambda}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0$$

$$\frac{\partial \Lambda}{\partial \xi_i} = C - \alpha_i - \beta_i = 0$$
The dual

Plugging into the Lagrangian we get the following dual formulation:

\[
\begin{align*}
\text{maximize } & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{subject to: } & \alpha_i \geq 0, \quad \sum_{i=1}^{n} \alpha_i y_i \geq 0 \\
& \beta_i \geq 0, \quad C - \alpha_i - \beta_i = 0
\end{align*}
\]

Beta appears only in the constraints. Replace it with the constraint

\[0 \leq \alpha_i \leq C\]
The dual

The final form of the dual becomes

$$\text{maximize } \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to: $0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0$
SVM: dual and primal

Primal:
\[
\begin{align*}
\text{minimize} \quad & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to:} \quad & y_i (w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, n.
\end{align*}
\]

Dual:
\[
\begin{align*}
\text{maximize} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j \\
\text{subject to:} \quad & 0 \leq \alpha_i \leq C, \quad \sum_{i=1}^{n} \alpha_i y_i = 0
\end{align*}
\]

Dual: simpler constraints; will allow us to use SVMs as non-linear classifiers
SVM solvers

Primal:
- Limited to linear SVMs
- Fast
- Software: LibLinear

Dual:
- Interior point methods (generic solvers for quadratic programming problems)
- SVM-specific solvers: SMO (optimize two alphas at a time)
- Software: LibSVM (a flavor of SMO)
- Approximate solvers (e.g. LASVM)
SMO

Sequential Minimal Optimization (SMO): A solver for the SVM dual problem.

When you choose two variables, the resulting problem can be solved analytically!

Issues and tricks:

- Which two variables to choose?
- Shrinking: temporarily remove variables that are less likely to be chosen (at upper/lower bounds). Need occasional “unshrinking”.

Platt, John (1998), Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines