Large margin classifiers: Support Vector Machines

Chapter 7

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 Valdimir 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

The geometric margin

The margin of a linear discriminant function $f$ with respect to a labeled dataset $D$:

$$ m_D(f) = \frac{1}{2} \hat{w}^T (x_\oplus - x_\ominus) $$

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

Want to find:
\[ m_D(f) = \frac{1}{2} \hat{w}^T(x_\oplus - x_\ominus) \]

Suppose that \( x_\oplus \) and \( x_\ominus \) are equidistant from the decision boundary:
\[ f(x_\oplus) = w^T x_\oplus + b = a \]
\[ f(x_\ominus) = w^T x_\ominus + b = -a \]

Subtracting the two equations:
\[ w^T (x_\oplus - x_\ominus) = 2a \]

Divide by the norm of \( w \):
\[ \frac{w^T (x_\oplus - x_\ominus)}{||w||} = \frac{2a}{||w||} \]

Theoretical motivation

Theorem: Let \( D \) be an i.i.d. sample of size \( n \) that is linearly separable and let \( m_0(f) \) be the margin associated with \( f(x) = w^T x + b \) then:
\[ P(y \neq \text{sign}(f(x))) \leq \frac{4}{\sqrt{n} m_D(f)} \]

Linear SVMs

Objective: maximize the margin while still correctly classifying all examples correctly
\[ \min_{\hat{w},b} \frac{1}{2} ||\hat{w}||^2 \]
subject to: \( y_i (\hat{w}^T x_i + b) \geq 1 \quad i = 1, \ldots, n \).
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 \(g(x)\)

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

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:

\[
\nabla f(x^*) + \sum \lambda_i \nabla g_i(x^*) = 0
\]

The Lagrangian function:

\[
\Lambda(x, \lambda) = f(x) + \sum \lambda_i g_i(x) = f(x) + \lambda^T g(x)
\]

The above condition is obtained by setting

\[
\nabla_x \Lambda(x, \lambda) = 0 \quad \nabla_x \text{ Denote differentiation with respect to } x
\]

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

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

where $\lambda > 0$

Constrained optimization with inequality constraints

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:

$$\nabla \Lambda(x, \lambda) = 0$$
$$g_i(x) \geq 0$$
$$\lambda \geq 0$$
$$\lambda_i g_i(x) = 0$$

These are known as the KKT conditions

Lagrangian duality

Claim: The problem of minimizing $f(x)$ s.t. $g(x) \leq 0$ 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. $g(x) \leq 0$ 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:

$$\max_x \min_\lambda \Lambda(x, \lambda) \text{ such that } \lambda \geq 0$$

This is called the dual

Under certain conditions (convexity) the two problems have the same solution

Back to SVMs

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 $\alpha_i > 0$

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

$$\Rightarrow b = y_i - w^T x_i$$

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 $\alpha_i > 0$

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

$$\Rightarrow b = y_i - w^T x_i$$

The correspond $x_i$ are called support vectors
Support Vectors

Claim: The number of support vectors is an upper bound on the estimated Leave-One-Out error.

The dual

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

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

The dual:

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

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

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

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

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

Compare to the primal:

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

subject to: \( y_i (w^T x_i + b) \geq 1 \quad i = 1, \ldots, n \).
The non-separable case

In order to allow for misclassification 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.
\[ \sum_{i=1}^{n} \xi_i \text{ is a bound on the number of misclassified examples} \]

SVMs for non-separable data

Our optimization problem for the non-separable case:
\[
\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i \\
\text{subject to: } y_i(w^T x_i + b) \geq 1 - \xi_i, \ \xi_i \geq 0, \ i = 1, \ldots, n.
\]

Let's form the Lagrangian:
\[
\Lambda(w, b, \alpha, \xi) = \frac{1}{2} ||w||^2 + C \sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} \alpha_i [1 - y_i(w^T x_i + b)] + \sum_{i=1}^{n} \beta_i \xi_i
\]
Saddle point equations:
\[
\frac{\partial \Lambda}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0
\]
\[
\frac{\partial \Lambda}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0
\]
\[
\frac{\partial \Lambda}{\partial \xi_i} = C - \alpha_i - \beta_i = 0
\]

The dual

Plugging into the Lagrangian we get the following dual formulation:
\[
\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]
\[
\text{subject to: } \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i \geq 0 \Rightarrow \beta_i \geq 0, \ C - \alpha_i - \beta_i = 0
\]
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

\[
\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]

subject to: \(0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0\)

SVM 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)

SVM: dual and primal

Primal:
\[
\min_{w,b} \frac{1}{2}||w||^2 + C \sum_{i=1}^{n} \xi_i
\]
subject to: \(y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \ i = 1, \ldots, n.\)

Dual:
\[
\max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j
\]
subject to: \(0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0\)

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

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