Support Vector Machines (and the Kernel Trick)

Ali Gooya

Reference: Christopher Bishop's *PRML Chapter 7*

Outline

Introduction to Support Vector Machines

Support vector machines are **non-probabilistic** binary linear classifiers.

The use of basis functions and the kernel trick mitigates the constraint of the SVM being a linear classifier

– in fact SVMs are particularly associated with the kernel trick.

Only a subset of data-points are required to define the SVM classifier

- these points are called support vectors.

SVMs are very popular classifiers and applications include

- text classification
- outlier detection
- face detection
- database marketing
- and many others.

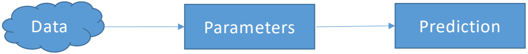
SVMs are also used for multi-class classification.

Also have support vector regression.

Look for the Bishop's Ch.7 for these applications.

Kernel methods and SVMs

- Parametric ML methods surrogate training data with optimal model parameters:
  - Examples: Linear regression models, MLPs, CNNs



- Kernel methods use kernels $k(\mathbf{x}, \mathbf{x}_n)$ to gauge similarities between the test point and training data points $\mathbf{x}_n$, and predict using:

$$y(\mathbf{x}) = \sum_n t_n k(\mathbf{x}, \mathbf{x}_n)$$

- This can be computationally challenging for large training data. SVMs are **sparse** kernel machines in a way that

$$y(\mathbf{x}) = \sum_n a_n t_n k(\mathbf{x}, \mathbf{x}_n)$$

- Where most of a_n coefficients become zero (hence sparsity).

The Separable Case

There are two classes which are assumed (for now) to be linearly separable.

Training data $\mathbf{x}_1, \ldots, \mathbf{x}_n$ with corresponding targets, $t_1, \ldots, t_n$ with $t_i \in \{-1, 1\}$.

We consider a classification rule of the form

$$\begin{aligned} h(\mathbf{x}) &= \text{sign}\left(\mathbf{w}^\top \mathbf{x} + b\right) \\ &= \text{sign}\left(y(\mathbf{x})\right) \end{aligned}$$

where $y(\mathbf{x}) := \mathbf{w}^\top \mathbf{x} + b$.

Note we can re-scale $(\mathbf{w}, b)$ without changing the decision boundary.

Therefore choose $(\mathbf{w}, b)$ so that training points closest to boundary satisfy $y(\mathbf{x}) = \pm 1$

- see Figure 7.1 from Bishop.

Let $\mathbf{x}_1$ be closest point from class with $t_1 = -1$ so that $\mathbf{w}^\top \mathbf{x}_1 + b = -1$.

And let $\mathbf{x}_2$ be closest point from class with $t_2 = 1$ so that $\mathbf{w}^\top \mathbf{x}_2 + b = 1$.

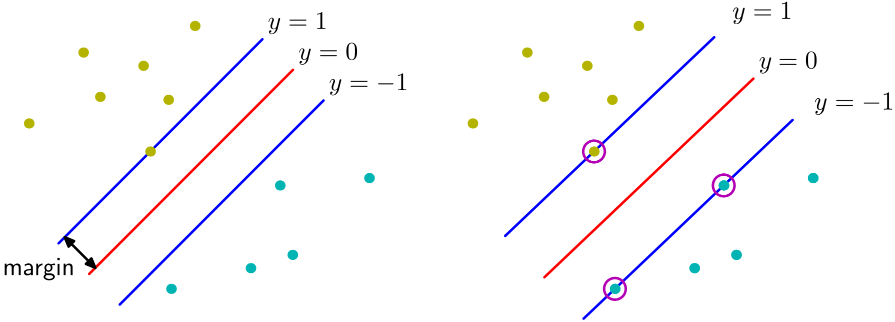


Figure 7.1 from Bishop: The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points, as shown on the left figure.
Maximizing the margin leads to a particular choice of decision boundary, as shown on the right. The location of this boundary is determined by a subset of the data points, known as support vectors, which are indicated by the circles.

Geometry of Maximizing the Margin

Recall the perpendicular distance of a point $\mathbf{x}$ from the hyperplane, $\mathbf{w}^\top \mathbf{x} + b = 0$, is given by

$$|\mathbf{w}^\top \mathbf{x} + b| / ||\mathbf{w}||.$$

Therefore distance of closest points in each class to the classifier is $1/||\mathbf{w}||$.

An SVM seeks the maximum margin classifier that separates all the data

- seems like a good idea
- but can also be justified by statistical learning theory.

Maximizing the margin, $1/||\mathbf{w}||$, is equivalent to minimizing $f(\mathbf{w}) := \frac{1}{2}\mathbf{w}^\top \mathbf{w}$.

Therefore obtain the following primal problem for the separable case:

$$\min_{\mathbf{w}, b} \quad f(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\top \mathbf{w} \tag{1}$$

$$\text{subject to} \quad t_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1, \quad i = 1, \ldots, n \tag{2}$$

Note that (2) ensures that all the training points are correctly classified.

The Primal Problem

The primal problem is a quadratic program with linear inequality constraints

- moreover it is convex and therefore has a unique minimum.

From the problem's geometry should be clear that only the points closest to the boundary are required to define the optimal hyperplane

- these are called the support vectors
- and will see that the solution can be expressed using only these points.