Support Vector Machines

Christos Dimitrakakis

November 21, 2023

Outline

Hyperplane

If $x \in \mathbb{R}^n$, then an affine subspace of dimension $n-1$ is a hyperplane.

Definition

A hyperplane in $\mathbb{R}^n$ is the set of points satisfying

$$\{\boldsymbol{x} : \beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x} = 0\}$$

Separating hyperplane

Consider a dataset $(\boldsymbol{x}_t, y_t)$ of points with $y_t \in \{-1, 1\}$. If

$$(\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_t) y_t \qquad \forall t$$

then the hyperplane separates the dataset.

The maximal margin hyperplane

The margin

For any a dataset $(\boldsymbol{x}_t, y_t)$, and hyperplane we can define the margin

$$M = \min_t (\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_t) y_t$$

as the minimum distance between the hyperplane and a correctly classied point.

The maximal margin hyperplane

Similarly, there is some $\beta_0, \boldsymbol{\beta}$ that achieves the maximum separation:

$$\max_{\beta_0, \boldsymbol{\beta}} \min_t (\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_t) y_t$$

The maximum margin classifier

The optimisation problem

We can write the problem like this

$$
\begin{aligned}
&\max_{\beta_0, \boldsymbol{\beta}, M} M && \text{(maximise the margin)}\\
&\text{s.t.} \|\boldsymbol{\beta}\| = 1 && \text{(invariance)}\\
&y_t(\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_t) \geq M \qquad \forall t \in [T]. && \text{(margin for all examples)}
\end{aligned}
$$

And we can divide by $\|\boldsymbol{\beta}\|$ to remove the norm constraint:

$$
y_t(\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_t) \geq M\|\boldsymbol{\beta}\|, \qquad \forall t \in [T]
$$

Setting $\|\boldsymbol{\beta}\| = 1/M$, we can rewrite this as

$$
\begin{aligned}
&\min_{\beta_0, \boldsymbol{\beta}} \|\boldsymbol{\beta}\|^2\\
&\text{s.t.} y_t(\beta_0 + \boldsymbol{\beta}^\top \boldsymbol{x}_t) \geq 1 && \forall t
\end{aligned}
$$

Quadratic programming

A quadratic program has the form:

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|^2$$
$$\text{s.t.} \boldsymbol{\beta}^\top x_t \geq 1 \forall t.$$

A constrained optimisation problem

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta})$$
$$\text{s.t.} g_i(\boldsymbol{\beta}) = 0 \forall i$$
$$h_i(\boldsymbol{\beta}) \geq 0 \forall i.$$

We can use the Lagrange method of multipliers to solve these problems.

Lagrange methods

Lagrange multipliers

For any local minimum $\boldsymbol{\beta}^*$, there is a unique vector $\boldsymbol{\lambda}^* \in \mathbb{R}^n$ so that

$$\nabla f(\boldsymbol{\beta}^*) + \sum_i \lambda_i^* \nabla h_i(\boldsymbol{\beta}^*) = 0.$$

The Lagrangian function

We first augment the original cost function to the Langrangian

$$L(\boldsymbol{\beta}, \boldsymbol{\lambda}) = f(\boldsymbol{\beta}) + \sum_{i=1}^{n} \lambda_i h_i(\boldsymbol{\beta})$$

For any local minimum $\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*$, $\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*) = 0$, $\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{\beta}^*, \boldsymbol{\lambda}^*) = 0$.

The Lagrange dual function

The dual function D is always concave:

$$D(\boldsymbol{\lambda}) = \inf_{\boldsymbol{\beta}} L(\boldsymbol{\beta}, \boldsymbol{\lambda}).$$

Support vector machines

- Hyperplanes do not always work
- How about a non-linear boundary?
- Instead of mapping the inputs through a non-linearity, map inner products to a kernel

Kernelised linear functions

We can rewrite

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

in terms of a kernel $K : X \times X \to \mathbb{R}$:

$$f(\boldsymbol{x}) = \beta_0 + \sum_{t=1}^{T} \alpha_t K(\boldsymbol{x}, \boldsymbol{x}_t), \qquad K(\boldsymbol{x}, \boldsymbol{x}_t) = \boldsymbol{x}^\top \boldsymbol{x}_t$$

because we can find $\boldsymbol{\alpha}$ so that

$$\sum_i \sum_t (\alpha_t x_{t,i}) x_i = \sum_i \beta_i x_i.$$

In fact it is sufficient to have: $\sum_t \alpha_t x_{t,i} = \beta_i$.

Kernels

Radial Basis Function

A simple type of non-linear layer in neural networks:

$$f(x) = \sum_i \alpha_i K(\boldsymbol{x}, \boldsymbol{c}_i), \qquad K(\boldsymbol{x}, \boldsymbol{c}_i) = \exp(-\|\boldsymbol{x} - \boldsymbol{c}_i\|^2),$$

where $\boldsymbol{c}_t$ are fixed centroids

Kernels in SVMs

Instead of fixed kernels, use the training data:

$$f(x) = \sum_t \alpha_t K(\boldsymbol{x}, \boldsymbol{x}_t),$$

Some common kernel choices

- Linear: $K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x}^\top \boldsymbol{x}$.
- RBFs: $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\|bx - \boldsymbol{x}'\|^2)$
- Polynomial: $K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^\top \boldsymbol{x})^d$.

Kernels as features*

Some kernels can be rewritten in terms of a feature mapping $\phi : X \to Z$

$$K(\boldsymbol{x}, \boldsymbol{x}') = \phi^\top(\boldsymbol{x})\phi(\boldsymbol{x})$$

- The mapping ϕ is implicit, and never computed.
- The dimension of Z can be infinite.
- So-called Mercer kernels are symmetric: $K(\boldsymbol{x}, \boldsymbol{x}') = K(\boldsymbol{x}', \boldsymbol{x})$.

Mercer kernels

$K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a Mercer kernel, if for any $\{\boldsymbol{x}_t : t \in [T]\}$, the kernel matrix $\boldsymbol{K} \in \mathbb{R}^{n \times n}$

$$\boldsymbol{K} \triangleq [K(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j \in [T]}$$

is symmetric positive semi-definite, i.e.

$$\boldsymbol{z}^\top \boldsymbol{K} \boldsymbol{z} \geq 0 \qquad \forall \boldsymbol{z} \in \mathbb{R}^n.$$