Support vector machine!

Meichen Lu (ml574@cam.ac.uk)

April 19, 2018

Contents

1 introduction

1.1 Motivation

1.2 Applications

- First applied for MNIST text recognition

2 Algorithms

2.1 Intuition

Begin with classification with two classes.

- Target is to find a boundary that is 'best' separates two classes. It means that we leave as big a **margin** as possible for both classes.
- To be able to compare different fitting, some kind of 'normalisation' is needed

2.2 More formal walk-through

For a linear decision boundary, we can express it as $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = 0$. Thus, one class will be $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b > 0$ and the other class $f(\boldsymbol{x}) < 0$. Introduce the class label as a random variable

y being 1 and -1 for the two scenarios such that $\gamma^{(i)} = y^{(i)}(\boldsymbol{w}^T \boldsymbol{x^{(i)}} + b)$ measures the distance of $x^{(i)}$ from the decision boundary. It is called **functional margin** or **geometric margin**.

Support vector the vectors x that lie on the boundary are called the support vectors. The following definition is introduced for computational convenience

$$yf(\boldsymbol{x}) = y(\boldsymbol{w}^T \boldsymbol{x} + b) = 1 \tag{1}$$

Next we need to find how to express the 'maximum margin' in maths. If we have two support vectors x^+ and x^- that belongs to the two classes, the width of the gap is thus $(x^+ - x^-) \cdot \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$. Substituting eq. 1, we find that: width $= \frac{2}{\|\boldsymbol{w}\|}$

$$\max \frac{2}{\|\boldsymbol{w}\|} \equiv \min \|\boldsymbol{w}\| \equiv \min \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w}$$

Kernel method For data that is not linearly separable, we can increase the dimensionality. If we apply a fixed feature-space transformation $\phi(x)$, we can convert the original input space $\boldsymbol{x}$ to a higher-dimensional space $\phi(x)$, where the contrast between two classes is exaggerated.

The functional margin becomes

$$y(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + b) = 1 \tag{2}$$

Lagrangian multiplier In order to solve this constrained optimization problem, we introduce Lagrange multipliers $a_n = 0$, with one multiplier an for each of the constraints.

$$L(\boldsymbol{w}, b, \boldsymbol{a}) = \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} - \sum_{n=1}^{N} a_n \{y_n(\boldsymbol{w^T} \boldsymbol{\phi}(\boldsymbol{x_n}) + b) - 1)\} \tag{3}$$

Taking the derivative of $L(\boldsymbol{w}, b, \boldsymbol{a})$ w.r.t $\boldsymbol{w}$ and b, we obtain

$$\boldsymbol{w} = \sum_{n=1}^{N} a_n y_n \boldsymbol{\phi}(\boldsymbol{x_n}) \tag{4}$$

$$0 = \sum_{n=1}^{N} a_n y_n \tag{5}$$

Eliminating $\boldsymbol{w}$ and b from $L(\boldsymbol{w}, b, \boldsymbol{a})$ gives the *dual representation*

$$\tilde{L}(\boldsymbol{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m \boldsymbol{\phi}(\boldsymbol{x_n})^T \boldsymbol{\phi}(\boldsymbol{x_m}) \tag{6}$$

We define $k(\boldsymbol{x_n}, \boldsymbol{x_m}) = \boldsymbol{\phi}(\boldsymbol{x_n})^T \boldsymbol{\phi}(\boldsymbol{x_m})$ to be the kernel function. After solving the dual problem, we can predict y using the sign of the $f(\boldsymbol{x})$

$$f(\boldsymbol{x}) = \sum_{n=1}^{N} a_n y_n k(\boldsymbol{x}, \boldsymbol{x_n}) + b \tag{7}$$

The Karush-Kuhn-Tucker (KKT) conditions are satisfied in this case:

$$a_n \leq 0 \tag{8}$$

$$y_n f(\boldsymbol{x_n} - 1) \leq 0 \tag{9}$$

$$a_n \{y_n f(\boldsymbol{x_n} - 1)\} = 0 \tag{10}$$

Therefore, either $a_n = 0$ or $y_n f(\boldsymbol{x_n}) = 1$. Any data point for which $a_n = 0$ (non-support vector) will not appear in the sum in Eq. 7 and hence plays no role in making predictions for new data points.

Regularization Cost function

$$\sum_{n=1}^{N} E_\infty(f(\boldsymbol{x_n})y_n - 1) + \lambda \|\boldsymbol{w}\| \tag{11}$$

where

$$E_\infty(z) = \begin{cases} 0, & \text{if } z \leq 0, \\ \infty, & \text{otherwise} \end{cases} \tag{12}$$