Project Notebook: Q-SVM

May 31, 2019

1 Classical SVMs

The support vector machine framework (SVM) [Norvig] is currently the most popular approach for supervised learning. There are three properties that make SVMs attractive:

1. SVMs construct a maximum margin separator - a decision boundary with the largest possible distance to example points. This helps them generalize well.
2. SVMs create a linear separating hyperplate, but they have the ability to embed the data into a higher-dimensional space, using the so-called **kernel trick.** Oftern data che are not linearly separable in the original input space are easily separable in the higher-dimensional space. The high-dimensional linear separator is actually nonlinear in the original space. This means that the hypothesis space is greatly expanded over methods that use strictly linear representations.
3. SVMs are a non-parametric method: the models need to retain training examples and potentially need to store them all. In practice they often end up retaining only a small fraction of the number of examples, i.e. approximately as few as a small constant times the number of dimensions. Thus SVMs combine the advantages of nonparametric and parametric models: they have to flexibility to represent complex functions, but they are resistant to overfitting.

(Raschka) SVMs could be considered as an extension of the perceptron: using the perceptron algorithm, we minimize misclassification errors. In SVMs, our optimization objective is to maximize the margin.

The margin is defined as the distance between the separating hyperplane (decision boundary) and the training samples which are closest to this hyperplane, which are the so-called support vectors.

1.1 Maximum margin intuition:

To get an intuition about margin maximization, let's take a closer look at those *positive* and *negative* hyperplanes that are parallel to the decision boundary, which can be expressed as follows:

$$w_0 + \boldsymbol{w}^T \boldsymbol{x}_+ = 1$$
$$w_0 + \boldsymbol{w}^T \boldsymbol{x}_- = -1$$

If we subtract those two linear equations from each other, we get:

$$\boldsymbol{w}^T \left(\boldsymbol{x}_+ - \boldsymbol{x}_-\right) = 2$$

Normalizing the expression with the length of $\boldsymbol{w}$ we obtain (please note that w_0 shouldn't be included in the vector)

$$\frac{\boldsymbol{w}^T \left(\boldsymbol{x}_+ - \boldsymbol{x}_-\right)}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|}$$

The left expression can be interpreted as the distance between the positive and negative hyperplane. Now the objective function of the SVM becomes the maximization of this margin. Alternatively, we could minimize the reciprocal of the right expression using quadratic programming:

$$\frac{\|\boldsymbol{w}\|^2}{2}$$

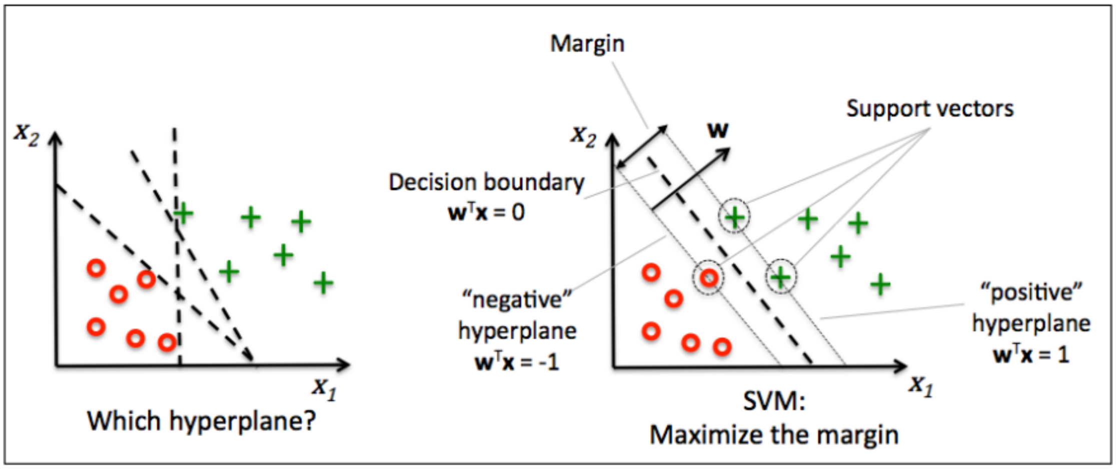

Figure 1: SVMs: selection of the decision boundary which maximzes the margin. ref. [Raschka]

1.2 Dealing with misclassification using slack variables

This method was introduced by Vladimir Vapnik in 1995 and let to the so-called soft-margin classification. The motivation for introducing the slack variable ξ was that the linear constraints need to be relaxed for nonlinearly separable data to allow convergence of the optimization in the presence of misclassifications under the appropriate cost penalization.

The positive-values slack variable is simply added to the linear constraints:

$$\begin{cases} \boldsymbol{w}^T\boldsymbol{x}^{(i)} \geq 1 & \text{if } y^{(i)} = 1 - \xi^{(i)} \\ \boldsymbol{w}^T\boldsymbol{x}^{(i)} \leq -1 & \text{if } y^{(i)} = 1 + \xi^{(i)} \end{cases}$$

so the new objective to be minimized (subject to the preciding constraints) becomes:

$$\frac{1}{2}\left\|\boldsymbol{w}\right\|^2 + C\sum_i \xi^{(i)}$$

where C is an hyper-parameter in order to control the penalty for misclassification

2 Kernel-based SVMs

[Raschka] Another reason why SVMs enjoy high popularity among machine learning practitioners is that they can be easily kernelized to solve nonlinear classification problems. Considering the infamous XOR classification problem (linear function has VC dimension of 3 in a 2-dim feature space: that model could shatter three points, but not four.)

We would not be able to separate samples frome the positive and negative class very well using a linear hyperplane as the decision boundary via the linear SVM model that we discussed earlier.

The basic idea behind kernel methods to deal with such linearly inseparable data is to create linear combinations of the original feature to project them onto a higher dimensional space via a mapping function $\phi(\cdot)$ where it becomes linearly separable.

However, one problem with this mapping approach is that the construction of the new features is computationally very expensive, especially if we are dealing with high-dimensional.data. This is where the so-called kernel-trick comes into play. Although we didn't go into much detail about how to solve the quadratic programming task to train an SVM, in practice all we need is to replace the dot product with mapped feature dot product.

$$\left(x^{(i)}\right)^T x^{(i)} \mapsto \phi\left(x^{(i)T}\right)\phi\left(x^{(i)}\right)$$

In order to save the expensive step of calculating this dot product between two points explicitly, we define a so-called kernel function:

$$K\left(x^{(i)}, x^{(j)}\right) = \phi\left(x^{(i)T}\right)\phi\left(x^{(i)}\right)$$

One of the most widely used kernels is the Radial Basis Function kernel (RBF kernel) or Gaussian kernel:

$$K\left(x^{(i)}, x^{(j)}\right) = \exp\left\{-\frac{\left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|^2}{2\sigma^2}\right\} \equiv \exp\left\{-\gamma\left\|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\right\|^2\right\}$$

where $\gamma = \frac{1}{2\sigma^2}$ is a hyper-parameter to be optimized.

3 Quantum SVM

4 Implementation

References

[RML-2014] **Rebentrost, P., Mohseni, M., & Lloyd, S.**
Quantum Support Vector Machine for Big Data Classification.
Physical Review Letters, 113(13) - (**2014**)
`doi:10.1103/physrevlett.113.130503`

[Norvig] Russell, Stuart J., and Peter Norvig. Artificial Intelligence A Modern Approach. 2016.