Support Vector Machines

Dwarikanath Mahapatra & Chen Chen

November 5-7, 2014

Tutorial Outline

- Review SVMs
- Soft Margin SVM

Support Vector Machines: Review

Finding A Separating Hyperplane

Solution 1: using the Perceptron algorithm

Finding A Separating Hyperplane

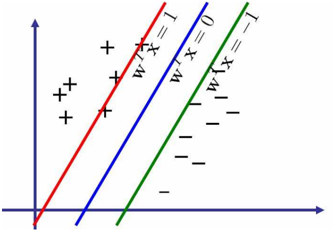
Solution 1: using the Perceptron algorithm

Solution 2: Maximum margin approach

A margin classifier is a classifier which is able to associate for each example a distance from the decision boundary. Linear Classifier has this property.

Why Maximize The Margin?

- Intuitively, it feels the safest.
- For a small error in the separating hyperplane, we do not suffer too many mistakes.
- Empirically, it works well.
- There is one global maximum, i.e. the problem is convex.

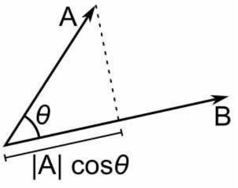


Deriving The Maximum Margin

Recall:

- For two vectors A and B, their inner product $\langle A, B \rangle$ is given by:
$$\langle A, B \rangle := \|A\| \|B\| \cos(\angle AB).$$
- If A and B are two vectors, the projection (C) of A on B is the vector that has the same slope as B with the length:

$$\|C\| = \|A\| \cos \theta$$

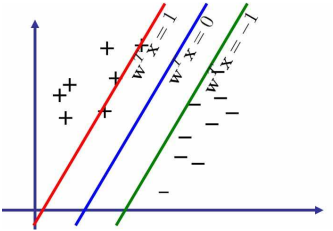


Deriving The Maximum Margin

Consider the figure below,
Let x_+ be a ‘positive’ sample, x_- be a ‘negative’ sample, and $x = x_+ - x_-$. The margin

$$d = \|x\| \cos\theta = \frac{\|x\|\|w\| \cos\theta}{\|w\|} = \frac{\langle w, x \rangle}{\|w\|} = \frac{\langle w, x_+ - x_- \rangle}{\|w\|} = \frac{2}{\|w\|}$$

where θ is the angle between x and the vector perpendicular to the hyperplane (**w**).



Hard Margin SVM

The objective of the hard margin support vector machine becomes:

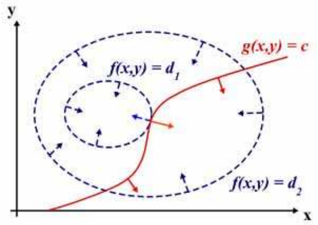
$$\begin{array}{ll} \text{minimize} & \|\mathbf{w}\| \\ \text{subject to} & y_i \mathbf{w}^T \mathbf{x} \geq 1 \quad \text{for } i = 1, \ldots, n. \end{array}$$

Use Lagrangian multipliers !!

Lagrangian functions

A strategy for finding the local maxima and minima of a function subject to equality constraints

- Maximize $f(x,y)$; subject to $g(x,y) = c$; f, g need to have continuous first partial derivatives.
- Introduce Lagrange variable λ and define the Lagrange function (lagrangian) given by $\Lambda(x,y,\lambda) = f(x,y) + \lambda \cdot (g(x,y) - c)$
- If $f(x_0, y_0)$ is a maximum of $f(x,y)$ then there exists λ_0 such that (x_0, y_0, λ_0) is a stationary point of Λ , i.e., $\nabla\Lambda = 0$



Hard Margin SVM Lagrangian

Primal form:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}w^T w \\ \text{subject to} & y_i \mathbf{w}^T \mathbf{x}_i \geq 1 \quad \text{for } i = 1, \ldots, m. \end{array}$$

* For convenience of derivation we use $\frac{1}{2}w^T w$

Lagrangian:

$$L(w, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{m} \alpha_i [y_i \mathbf{w}^T \mathbf{x}_i - 1]$$

Finding stationary points (Deriving with respect to the primal variable):

1. $\frac{\partial L(w,\alpha)}{\partial w} \rightarrow$ What is $\frac{\partial w^T w}{\partial w}$??

Vector Calculus: Derivatives

The partial derivative of a function $g : R^n \to R$ of a vector $w \in R^n$, with respect to the vector itself is given by:

$$\frac{\partial g(w)}{\partial w} = \begin{bmatrix} \frac{\partial g(w)}{\partial w_1} \\ \frac{\partial g(w)}{\partial w_2} \\ \vdots \\ \frac{\partial g(w)}{\partial w_n} \end{bmatrix}.$$

$\langle w, w \rangle = w_1^2 + w_2^2 + \ldots + w_n^2$. The partial derivative of $\langle w, w \rangle$ is given by

$$\frac{\partial \langle w, w \rangle}{\partial w} = 2w$$

Hard Margin SVM Dual

Finding extreme points:

$$\tfrac{\partial L(w,\alpha)}{\partial w} = w - \sum_{i=1}^{m} \alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^{m} \alpha_i y_i x_i \quad .$$

Substituting $w = \sum_{i=1}^{m} \alpha_i y_i x_i$ into $L(w, \alpha)$:

$$L(w,\alpha) = \tfrac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{m} \alpha_i$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad .$$

Hard Margin SVM Dual

The Hard Margin SVM Dual becomes:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

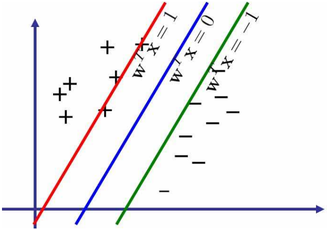
.

subject to:
$\forall i \quad \alpha_i \geq 0$

Support Vectors

* The separating hyperplane is determined only by the points which are closest to it (the support vectors).
The points which do not lie on the boundary do not contribute to the classification

In theory, you may have millions of data points, but only three support vectors.



Soft Margin SVM Motivation

- Non-separable datasets - in real life most datasets are in fact non separable

  Non linearly separable Use kernel trick
  Non separable introduce slack variables

- Sensitivity to outliers

Soft Margin SVM

Key Ideas:

1. Introduction of slack variables
   (see optimization techniques for more details on slack variables).
2. Penalize the misclassified examples

Hard Margin SVM: The margin term is defined by the hard constraints:

$$\forall i \; y_i \mathbf{w}^T \mathbf{x}_i \geq 1, y_i \in \{+1, -1\}$$

Soft margin SVM: soften the constraints:

$$\forall i \; y_i \mathbf{w}^T \mathbf{x}_i \geq 1 - \xi_i, \xi_i \geq 0$$

This new constraint permits a functional margin that is less than 1.

Note: Slack variables apply only to training data. Classification of test points depends only on which side of the hyperplane they are on.

Soft Margin SVM

Penalty Cost: we still prefer a hyperplane which correctly classifies the data.

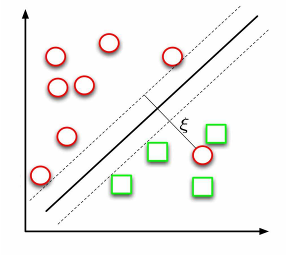
Introduce $C\xi_i$ penalty for each data point *i* which falls:

- Within the margin ($0 < \xi \leq 1$)
- On the wrong side of the hyperplane ($\xi > 1$).

→ The penalty is proportional to the amount by which
the example is misclassified.

Target: Minimize the sum of the total penalties ξ_i over all *i* (This is an upper bound for the training classification error).

Soft Margin SVM



Soft margin SVM becomes

$$\min_{w,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$

Subject to:

$$y_i \mathbf{w}^T \mathbf{x}_i \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0 \text{ for all } i = 1, \ldots, m.$$

- There are many ways to combine two cost terms into a cost function: Every expression that becomes larger as either of the two terms increases is valid.
- Soft margin SVM: Choose the simple way, add the terms.

  cost of solution = margin costs + C * slack costs (in L_1 norm)
- C is a constant which controls the amount of constraint violations vs. margin maximization. The value of C is found using cross validation.

Soft Margin SVM Lagrangian

Exercise:

$$L(w, \xi, \alpha, \beta) =$$

$$\frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \alpha_i [y_i \mathbf{w}^T \mathbf{x}_i - 1 + \xi_i] - \sum_{i=1}^{m} \beta_i \xi_i$$

SVM Summary

- By definition, SVM is the maximum margin classifier defined in terms of the support vector approach.
- Real-world SVM implementations usually combine three techniques:
  1. Maximum margin classifier (this is where convex optimization comes in)
  2. Soft margin technique (slack variables)
  3. Kernel trick
- Three very simple reason why SVMs are so popular:
  1. Of proven merit.
  2. Lots of experience, literature etc exists.
  3. Several easy-to-use, freely accessible, well-tested implementations are available (libsvm, svmlite, shogun etc.)