Project 2 – Support Vector Machines

Master DMKM

3/12/2015

Description Support Vector Machines (SVM)

are one of the most widely used methods for classification problems. An SVM classifier implies the solution of a quadratic or linear programming problem.

We could define the SVMs as a simple binary classification method using hyperplanes. The objective of our problem is to build an hyperplane which separates two classes, labeled with a $+1$ or -1, minimizing the error of classification and maximizing the margin between the two separation hyperplanes.

This idea is illustrated by Figure 1 and Figure 2. Figure 1 shows a possible separation of the two classes using two lines. From the dashed line to the right we classify the "circle" class. And from the continuous line to the left we classify the "star" class. Using this classification, only a "circle" is wrongly classified, but the two lines intersect. Figure 2 shows the classification that would be done by a SVM. This classification is better. The separation margin of the two classes (distance between lines) is larger and it is more difficult to make an error when classifying a new point.

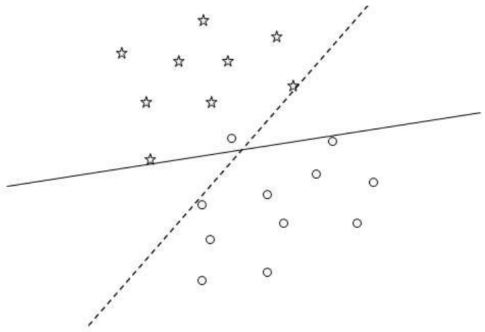

Figure 1: Example of bad classification

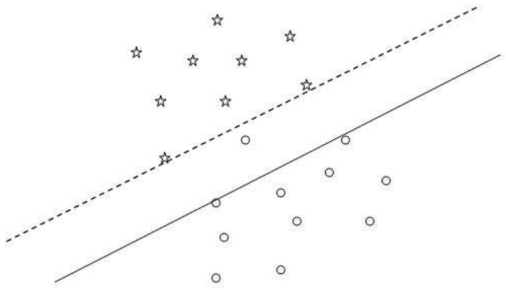

Figure 2: Example of good classification (SVM classification)

Problem modelling Our purpose is

to classify m points of $\mathbb{R}^n$, represented by the matrix $A \in \mathbb{R}^{m \times n}$. Each point belongs to the class $+1$ or -1 depending on its classification in a diagonal matrix $D \in \mathbb{R}^{m \times m}$: the diagonal element $D_{ii} = +1$ if the point i belongs to the class $+1$, otherwise $D_{ii} = -1$ and the point i belongs to the class -1. For this problem, the standard SVM with a linear kernel is characterized by the following quadratic optimization problem for some parameter $\nu > 0$:

$$\begin{aligned} \min_{(\omega,\gamma,y)\in\mathbb{R}^{n+1+m}} \quad & \nu e^T y + \frac{1}{2} \parallel \omega \parallel_2^2 \\ s.t \quad & D(A\omega - e\gamma) + y \geq e \\ & y \geq 0 \end{aligned}$$

Hint: $\omega \in \mathbb{R}^{n \times 1}$, $\gamma \in \mathbb{R}^{1 \times 1}$, $y \in \mathbb{R}^{m \times 1}$ and $e \in \mathbb{R}^{m \times 1}$.

As shown in Figure 3, ω is orthogonal to planes

$$x^T\omega = \gamma + 1$$
$$x^T\omega = \gamma - 1$$

γ determines its location relative to the origin (dashed line in the middle). The plane on the top delimits the $+1$ class (starts) and the plane on the bottom delimits the -1 class (circles). y_i with $i = 1, \ldots, m$ means the distance from point i to the plane of its true class if point i locates in the wrong class. When both classes are strictly *linearly separable*, then the values y can be zero. THe separating surface is the plane

$$x^T\omega = \gamma$$

the dashed line in the middle of the two delimiting planes. If the classes are linearly non-separable as in Figure 3, then both planes delimit the two classes with a "smooth margin" determined by the nonnegative variable y, therefore:

$$x^T\omega - \gamma + y_i \geq +1, \quad for \quad x^T = A_{i.} \quad and \quad D_{ii} = +1$$
$$x^T\omega - \gamma + y_i \leq -1, \quad for \quad x^T = A_{i.} \quad and \quad D_{ii} = -1$$

where $A_{i.}$ is the i-th row of A.

The objective function minimizes the one-norm of the variables y weighted by the parameter ν, plus the quadratic term $\frac{1}{2} \parallel \omega \parallel_2^2$. This last term can be written as $2/(\frac{2}{||\omega||_2})^2$, where $\frac{2}{||\omega||_2}$ is the margin or distance between the two separating hyperplanes. To minimize the inverse of the margin is equal to maximize the margin, that is, to obtain the most separated hyperplanes.

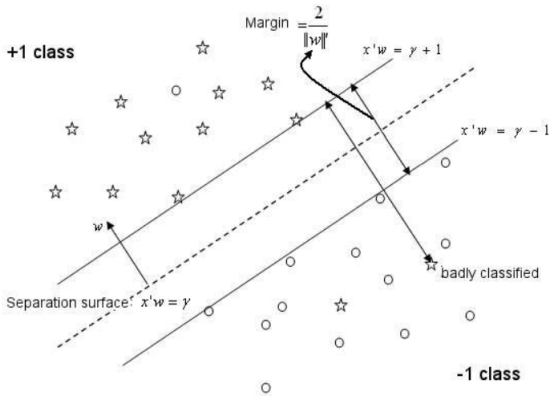

Figure 3: SVM components

No we will detail how the margin between the two planes $x^T\omega = a$ and $x^T\omega = b$ is computed and we will check that is equal to $\frac{2}{||\omega||_2}$: suppose that we have point x_1 on the first hyperplane ($x_1^T\omega = a$). The nearest point x_2 on the second hyperplane ($x_2^T\omega = b$) is obtained as $x_2^T = x_1^T + \alpha\omega^T$, where $\alpha \in \mathbb{R}$ has to be computed. Multiplying by ω we have:

$$\begin{aligned} x_2^T\omega &= x_1^T\omega + \alpha\omega^T\omega \\ b &= a + \alpha||\omega||_2^2 \\ \alpha &= \frac{b-a}{||\omega||_2^2} \end{aligned}$$

The distance or margin between two hyperplanes is the Euclidean distance of $\alpha\omega$, using the previous expression of $alpha$ we have:

$$\begin{aligned} ||\alpha\omega|| &= |\alpha| \cdot ||\omega||_2 \\ &= \frac{|b-a|}{||\omega||_2^2}||\omega||_2 \\ &= \frac{|b-a|}{||\omega||_2} \end{aligned}$$

In the case of SVM, $a = \gamma - 1$ and $b = \gamma + 1$, thus the numerator of the margin expression is $|b - a| = 2$, so the margin coincides with the expected result: $\frac{2}{||\omega||_2}$.

Particular case to be solved The task is to implement and solve

We provide you a small program for input data generation. The program is named *gensvmdat*, and you have to run it through the command:

$$gensvmdat \quad file \quad p \quad seed$$

where *file* is the name of the data file , p is the number of lines (or points) of the file, and *seed* is a seed for the random number generator. For example, if you type: "gensvmdat data.dat 10 12345", the file "data.dat" will be similar to Figure 4:

```
0.178 0.400 0.167 0.212   -1.0
0.062 0.054 0.276 0.048   -1.0
0.321 0.273 0.368 0.222   -1.0
0.220 0.202 0.044 0.344   -1.0
0.648 0.149 0.030 0.878   -1.0
0.655 0.528 0.506 0.916   1.0
0.188 0.119 0.058 0.720   -1.0
0.638 0.881 0.412 0.816   1.0
0.281 0.579 0.028 0.342   -1.0
0.633 0.304 0.390 0.954   1.0
```

Figure 4: Data sample generated by gensvmdat

Each row is a data point. The first four numbers of each line are random numbers between 0 and 1. If the result of adding up these values is greater than 2, the fifth value is 1. If it is lower than 2, the fifth value is -1. We want the SVM to generate a separation or classification plane between the 2 types of data. The program also produces, with a small probability, some badly classified points, which are marked with an "*" in the file.

The task is to formulate and solve the illustrated problem using CVXOPT and CVXPY. You can work in pair or individually. A report plus code should be submitted before the deadline, which is the 17 December 2015.