Point classification Problem

CSL758, Advanced Algorithms

by

Ayush Nayyar(2004cs50212)
Shailen Agrawal(2004cs50521)

0.1 Introduction

We have n points $p_1, p_2, p_3....p_n$ in d dimensional space where each p_i is of the form $({x_1}^i, {x_2}^i, ...{x_d}^i)$
We assume that the partitioning plane passes through the origin. i.e. the plane seperating the blue and the red points is of the form

$$\sum_{i=1}^{d} a_i x_i = 0$$

Not the only does the plane seperate the points, it seperates it by some margin i.e.
For the blue points we must have $\sum_{i=1}^{d} a_i x_i > \delta$ and for the red points we must have $\sum_{i=1}^{d} a_i x_i < -\delta$
So a harder problem is to find the best hyperplane, i.e. the plane for which δ is maximum
The formal problem statement of such a hyperplane with margin is : for every point the distance of p_i from the plane is greater than δ i.e. $|(a.p_i)| > \delta$ where $a = (a_1, a_2, ...a_d)$

Our algorithm will not compute the best plane but the fact that points are well seperated will help in reducing the mistakes and also in the analysis.

0.2 Algorithm

Aim : To come up with $(w_1, w_2, ...w_d)$ s.t $\sum_{i=1}^{d} w_i.x_i = 0$ seperates the red points from the blue points

1. start with $(w_1, w_2, ...w_d) = (0,0,0,....0)$
2. Repeat for each p_i

- If p_i is blue and $(w.p_i) < 0$ then update w = w+ p_i
- If p_i is red and $(w.p_i) > 0$ then update w = w- p_i

Till all p_i conditions are correctly satisfied.

3. Return $(w_1, w_2, ...w_d)$

The updation we do is local, i.e. we just try to satisfy the points in question. w_0 = (0,0,0...0)

$$\mathrm{w}_t \;\; - w_{t-1} = f_i p_i \;(1)$$

if we make a mistake at the t^{th} iteration and where $f_i = 1$ if p_i is blue and $f_i = -1$ if p_i is red

0.3 Analysis

Bound the number of mistakes committed by the algorithms

An incorrectly classified point is a mistake and we can commit a number of mistakes with the same point, so by bounding the number of mistakes we bound the no. of steps the algorithm performs.

$$\mathrm{w}_{final} \;\; - w_0 = \sum f_i \;(2)$$

Let us assume the vector of the best hyperplane is a i.e. for the blue points we must have $\sum_{i=1}^{d} a_i x_i > \delta$ and for the red points we must have $\sum_{i=1}^{d} a_i x_i < -\delta$
Note that $\forall$ i the product $(f_i p_i).a$ is always positive and greater than δ . Because for Blue points $f_i = 1,\ and\ (a.p) > \delta$ and for Red points $f_i = -1, and\ (a.p) < -\delta$.

$$(\mathrm{w}_{final} - w_0, a) = (\sum f_i p_i, a)$$
$$(\mathrm{w}_{final}, a) = (\sum f_i p_i, a)$$
$$\geq (no.\ of\ mistakes)\delta$$
$$= \mathrm{M}.\delta$$

$$Also\ |w_{final}| \geq (w_{final}, a) \tag{3}$$

A mistake in t means $w_t \ - w_{t-1} = f_i$

$$(w_t, w_t) = |w_t|^2$$
$$= \ (w_{t-1} + f_i p_i, w_{t-1} + f_i p_i)$$
$$= \ |w_{t-1}|^2 + 2(w_{t-1}, f_i p_i) + |f_i p_i|^2 = |w_{t-1}|^2 + something\ always\ negative + something\ always\ 1$$

which means

$$|w_t|^2 - |w_{t-1}|^2 \leq 1$$
$$|w_{final}|^2 - |w_0|^2 \leq no.of mistakes(M)$$
$$|w_{final}| \leq \sqrt{M}$$
$$\delta M \leq \sqrt{M}$$
$$M \leq \frac{1}{\delta^2}$$

So the number of mistakes made by our algorithm is $\leq \frac{1}{\delta^2}$

For a general partitioning hyperplane which is not passing through the origin, we increase the dimension of the points.

$$\sum_i a_i x_i = b$$
$$-b.1 + a_i x_1 + a_2 x_2 \ldots . a_d x_d > 0 \text{ for blue points and } -b.1 + a_i x_1 + a_2 x_2 \ldots . a_d x_d < 0 \text{ for red points}$$

To accomodate this we convert all points p to the form $(1, x_1, x_2, \ldots x_d)$ and the best hyperplane now becomes $(-b, a_1, a_2 \ldots . a_d)$.

0.4 Finding the best hyperplane

So now we want to find the hyperplane which has the broadest margin.
maximise 2δ such that $\sum a_i x_i \geq b + \delta$ for blue points and $\sum a_i x_i \leq b - \delta$ and $|a|_2 = 1$. Let's try to eliminate δ
Call

$$\frac{a_i}{\delta} = w_i$$
$$\frac{b}{\delta} = b'$$

Maximising δ means minimising $|w|_2$.So our input to the linear program is as follows :

$$\text{minimise } |w|_2$$
$$\sum w_i {x_i}^j \geq b' + 1 \text{ for } p_j \text{ blue}$$
$$\sum w_i {x_i}^j \leq b' - 1 \text{ for } p_j \text{ red}$$

0.5 When points are not perfectly seperable

We need to allow for a few misclassifications. So now we try to minimise $|w|_2 + C(\sum z_j)$ where $C(z_j)$ is the cost associated with mistake z_j under the constraints

$$\sum w_i {x_i}^j \geq b' + 1 - z_j \text{ for } p_j \text{ blue}$$
$$\sum w_i {x_i}^j \leq b' - 1 - z_j \text{ for } p_j \text{ red}$$

So this problem can be solved by giving the above constraints as inputs to a linear program.