HomeWork 4 Solutions

Machine Learning

December 2, 2015

1 Warm up: Support Vector Machines

1.1

The equation of the optimal hyper plane is x = 2 and the support vectors are

{ {(3,1),+1} , {(3,-1),+1} and {(1,0),-1} }

1.2

Yes

1.3

No, not necessarily. Assume that the initial weight vector is represented by equation x = 1.5 , it already classifies the given data correctly and thus the perceptron algorithm will not modify it any further.

2 Kernel and Perceptron Algorithm

2.1

To show that any K-DNF is separable after feature transformation, we use the following transformation.
Say we have a K-DNF where each conjunction is represented by k_i.
So for example $k_1 = x_1 \wedge x_2 \wedge ... \wedge x_k$
the K-DNF = $k_1 \vee k_2 \vee ..k_n$
The feature transformation would be
$\phi(x) = [k_1, k_2, ...k_n]$
This transformation makes a K-DNF linearly separable as you can define
$w = [1, 1, 1, ..1]$ and bias $b = -1$ and check for the $sign(w^t \phi(x) + b)$

2.2

We need to find a way to compute this without actually computing $\phi(x_1) and \phi(x_2)$
If we observe the summation it must be clear that only the terms where $c(x_1)$and $c(x_2)$ are both true contribute to the value of $K(x_1, x_2)$,

i.e $c(x_1)c(x_2) = 1$
or $c(x_1) = 1$ and $c(x_2) = 1$
This implies that relevant conjunction "c" must contains only those variables that are same for both x_1 and x_2. Thus, we just need to count the number of relevant conjunctions "c" from the set C.

$$K(x_1, x_2) = \binom{Equal(x_1, x_2)}{k}$$

where $Equal(x_1, x_2)$ is the number of features where both x_1 and x_2 have the same value. It needs to be noted that features need to equal, that is both must be equal to either 0 or 1.

2.3

Let w_i be weight vector after i examples, According to Perceptron algorithm, an update is performed when the training example is mis-classified as follows:

$$w_{i+1} = w_i + y_i x_i$$

and the final weight vector is written as $\mathbf{w} = \sum_{(x_i,y_i)\in M} y_i x_i$ where $M = \{(x_i, y_i)\}$ set of examples on which the perceptron makes mistakes
when a transformation $\phi(x)$ is applied the new update becomes

$$w_{i+1} = w_i + y_i \phi(x_i)$$

and the final weight vector thus would be

$$\mathbf{w} = \sum_{(x_i,y_i)\in M} y_i \phi(x_i)$$

where $M = \{(x_i, y_i)\}$ is the set of examples on which the perceptron makes mistakes

2.4

It is known from the previous question that

$$\mathbf{w} = \sum_{(x_i,y_i)\in M} y_i \phi(x_i)$$

where $M = \{(x_i, y_i)\}$ is the set of examples on which the perceptron makes mistakes

$$y = \mathbf{sgn}(\mathbf{w}^T \phi(\mathbf{x}))$$

$$y = \mathbf{sgn}(\sum_{(x_i,y_i)\in M} y_i \phi(x_i)^T \phi(\mathbf{x}))$$

$$y = \mathbf{sgn}(\sum_{(x_i,y_i)\in M} y_i K(x, x_i))$$

2.5

Pseudo Code for kernel perceptron algorithm :
Initialize M = {} and Let S denote the set of all examples

for each example x_i, y_iin S(x,y)

if($y \neq sgn(\sum_{(x_i,y_i)\in M} y_i K(x, x_i)))$

M = M $\cup$ (x_i, y_i)

return $\sum_{(x_i,y_i)\in M} y_i K(x, x_i)$