CS446: Machine Learning **Fall 2014**

Problem Set 5 Solutions

Handed Out: October 28^{th}, 2014 *Handed In: November 06^{th}, 2014*

1. **[SVM - 50 points]**

   (a) (1) An easy solution will be $\mathbf{w} = (-1, 0)$ and $\theta = 0$.

   (2) $\mathbf{w} = (-0.5, 0.25)$ and $\theta = 0$.

   (3) Assume $\mathbf{w}^* = (w_1, w_2), \theta^*$ is the solution. We can directly find it as the largest margin hyperplane by geometry. The closest positive and negative examples are examples 1 and 6. The largest margin plane should be the perpendicular bisector. The midpoint of the two is (0.4, 0.8), and the slope of the segment connecting the two is $(1.6 - 0)/(-1.2 - 2) = -0.5$, so the slope of the perpendicular bisector is 2, i.e. $w_1 = -2w_2$. (0.4, 0.8) is on it then we know θ^* is 0.

   By the constraints $y^{(i)}(\mathbf{w} \cdot \mathbf{x}^{(i)} + \theta) \geq 1$, $i \in \{1, 6\}$, we can get $\mathbf{w}^* = (-0.5, 0.25)$.

   (b) (1) $I = \{1, 6\}$

   (2) $\alpha = \{0.15625, 0.15625\}$.
   We can get this easily from $\mathbf{w}^* = \alpha_1 y^{(1)}\mathbf{x}^{(1)} + \alpha_2 y^{(6)}\mathbf{x}^{(6)}$

   (3) Objective function value: $\frac{1}{2}||\mathbf{w}^*||^2 = 0.15625$.

   (c) C determines the tradeoff between the model complexity (here, the norm of $\mathbf{w}$) and the training loss (here, the hinge loss from all examples). When $C = \infty$, we minimize the training loss, and obtain the max-margin hyperplane. In this case, typically, a small number of examples will be support vectors. When C is very small, we minimize the model complexity, and obtain an hyperplane that has a very small norm. In this case, most examples will be support vectors because their hinge loss will be positive. When $C = 0$, we will get $\mathbf{w} = \mathbf{0}$. When C has a moderate value ($C = 1$), we penalize both the model complexity and the training loss. In this case, typically, the number of support vectors will be more than the number of support vectors in the max-margin hyperplane. This is desirable in practice because we want to avoid overfitting.

2. **[Kernels - 15 points]**

   (a) In the dual representation, the weight vector can be presented as a collection of "important" examples M on which the Perceptron makes mistakes. We predict the label for a new example $\vec{\mathbf{x}}$ using the following equation:

   DualPrediction(M,$\vec{\mathbf{x}}$)

   $$\text{return } Th_\theta\left(\sum_{(\vec{\mathbf{x}}_m, y_m)\in M} y_m \vec{\mathbf{x}}^T \vec{\mathbf{x}}_m\right)$$

The training algorithm is as follows:

$M \leftarrow \emptyset$
for $(\vec{\mathbf{x}}, y) \in S$ **do**
  if $y \not\equiv$ DUALPREDICTION(M,$\vec{\mathbf{x}}$) **then**
    $M \leftarrow M \cup (\vec{\mathbf{x}}, y)$
  end if
end for

(b) From the lecture notes, we know that a function $K(\mathbf{x}, \mathbf{z})$ is a valid kernel if it corresponds to an inner product in some (perhaps infinite dimensional) feature space.

Given that we know that K_1 and K_2 are both valid kernel functions, there exists two functions ϕ_1 and ϕ_2 such that

$$\begin{aligned} K_1(\vec{\mathbf{x}}, \vec{\mathbf{z}}) &= \phi_1(\vec{\mathbf{x}})^T \phi_1(\vec{\mathbf{z}}) \\ \text{and } K_2(\vec{\mathbf{x}}, \vec{\mathbf{z}}) &= \phi_2(\vec{\mathbf{x}})^T \phi_2(\vec{\mathbf{z}}) \\ \text{Define} \quad \phi(\vec{\mathbf{x}}) &= \left[\sqrt{\alpha}\phi_1(\vec{\mathbf{x}}) \quad \sqrt{\beta}\phi_2(\vec{\mathbf{x}})\right] \\ \text{It follows that} \quad K(\vec{\mathbf{x}}, \vec{\mathbf{z}}) &= \phi(\vec{\mathbf{x}})^T \phi(\vec{\mathbf{z}}) = \alpha K_1(\vec{\mathbf{x}}, \vec{\mathbf{z}}) + \beta K_2(\vec{\mathbf{x}}, \vec{\mathbf{z}}) \end{aligned}$$

Therefore, K is a valid kernel function.

(c) The easiest way prove that K is a valid kernel function is to use the result from the previous question. This result easily generalizes from a sum of two kernels to a sum of any number of kernels. Now, it remains to be shown that $(\vec{\mathbf{x}}^T\vec{\mathbf{z}})^3$, $(\vec{\mathbf{x}}^T\vec{\mathbf{z}})^2$, and $\vec{\mathbf{x}}^T\vec{\mathbf{z}}$ are valid kernel functions. The last one is trivial.

To show that $(\vec{\mathbf{x}}^T\vec{\mathbf{z}})^3$ is a valid kernel, we can show that it is a dot product.

$$\begin{aligned} (\vec{\mathbf{x}}^T\vec{\mathbf{z}})^3 &= x_1^3 z_1^3 + 3x_1^2 x_2 z_1^2 z_2 + 3x_1 x_2^2 z_1 z_2^2 + x_1^3 z_1^3 \\ &= \phi(\vec{\mathbf{x}})^T \phi(\vec{\mathbf{z}}) \end{aligned}$$

where $\phi(\vec{\mathbf{x}})$ is defined as

$$\phi(\vec{\mathbf{x}}) = \begin{bmatrix} x_1^3 \\ \sqrt{3}x_1^2 x_2 \\ \sqrt{3}x_1 x_2^2 \\ x_2^3 \end{bmatrix}$$

The proof that $(\vec{\mathbf{x}}^T\vec{\mathbf{z}})^2$ is a kernel is very similar and is left as an exercise.

3. **[Boosting - 30 points]**

(a) We note that $D_0(i) = 0.1$ for all ten examples. Looking at the given data, we see that the weak learners (*rules of thumb*) with lowest errors are : $x_1 \equiv [x > 5]$ and

i	Label	D_0	$x_1 \equiv [x > 5]$	$x_2 \equiv [y > 6]$	$h_1 \equiv [x > 5]$	D_1	$x_1 \equiv [x > 8]$	$x_2 \equiv [y > 8]$	$h_2 \equiv [y > 8]$
		Hypothesis 1				Hypothesis 2			
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
1	-1	0.1	-1	+1	-1	0.0625	-1	+1	+1
2	-1	0.1	-1	-1	-1	0.0625	-1	-1	-1
3	+1	0.1	+1	+1	+1	0.0625	-1	-1	-1
4	-1	0.1	-1	-1	-1	0.0625	-1	-1	-1
5	-1	0.1	-1	+1	-1	0.0625	-1	+1	+1
6	+1	0.1	+1	+1	+1	0.0625	-1	-1	-1
7	+1	0.1	+1	+1	+1	0.0625	+1	+1	+1
8	-1	0.1	-1	-1	-1	0.0625	-1	-1	-1
9	+1	0.1	-1	+1	-1	0.25	-1	+1	+1
10	-1	0.1	+1	+1	+1	0.25	-1	-1	-1

Table 1: Table for Boosting results

$x_2 \equiv [y > 6]$.

$$\begin{aligned}
\epsilon_{x_1} &= [\text{weighted sum of mistakes if } h = x_1] = \frac{2}{10} = 0.2 \\
\epsilon_{x_2} &= [\text{weighted sum of mistakes if } h = x_2] = \frac{3}{10} = 0.3 \\
\therefore \alpha_0 &= \frac{1}{2}\log_2 \frac{1-\epsilon}{\epsilon} = \frac{1}{2}\log_2 \frac{0.8}{0.2} = 1
\end{aligned}$$

Hence, the first weak learner is $h_0 = x_1$. Also see Table 1.

(b) Using α_0 to compute the new distribution, we get:

$$\begin{aligned}
D_{t+1}(i) &= \begin{cases} \frac{1}{Z_0} D_0(i) 2^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ \frac{1}{Z_0} D_0(i) 2^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\
\therefore D_1(i) &= \begin{cases} \frac{1}{20 Z_0} & \text{if } h_0(x_i) = y_i \\ \frac{1}{5 Z_0} & \text{if } h_0(x_i) \neq y_i \end{cases}
\end{aligned}$$

To calculate Z_0,

$$\frac{8}{20 Z_0} + \frac{2}{5 Z_0} = 1 \implies Z_0 = \frac{4}{5}$$

$$\therefore D_1(i) = \begin{cases} \frac{1}{16} = 0.0625 & \text{if } h_0(x_i) = y_i \\ \frac{1}{4} = 0.25 & \text{if } h_0(x_i) \neq y_i \end{cases}$$

The new weak learners (*rules of thumb*) for this new distribution D_1 are

$x_1 \equiv [x > 8]$ and $x_2 \equiv [y > 8]$.

$$\begin{aligned}
\epsilon_{x_1} &= [\text{weighted sum of mistakes if } h_1 = x_1] = \frac{1}{4} + \frac{2}{16} = \frac{3}{8} \\
\epsilon_{x_2} &= [\text{weighted sum of mistakes if } h_1 = x_2] = \frac{4}{16} = \frac{1}{4} \\
\alpha_1 &= \frac{1}{2}\log_2 \frac{1-\epsilon}{\epsilon} = \frac{1}{2}\log_2 \frac{3/4}{1/4} = \frac{1}{2}\log_2(3) = 0.79
\end{aligned}$$

(c) The final hypothesis produced by AdaBoost is

$$H(x) = sgn\big(1[x > 5] + 0.79[y > 8]\big)$$

If we use natural logarithms, the final hypothesis is just a scaled equivalent:

$$H(x) = sgn\big(0.695[x > 5] + 0.549[y > 8]\big)$$

In this case, the $D_1(i)$s should be computed with base e rather than base 2, and you can check the final values and other calculations don't change.

4. **[Probability - 5 points]**

(a) i. Let X be a random variable to denote number of children in a family.

- Town A: Since every family has exactly one child (a uniform distribution), the expected value of number of children, $E[X] = 1$.
- Town B: The expected value of number of children is given as

$$\begin{aligned}
E[X] &= \sum_i i \cdot \Pr(X = i) \\
&= 1 \times \frac{1}{2} + 2 \times \left(\frac{1}{2}\right)^2 + \dots
\end{aligned}$$

Notice that this is the same as finding the expected value of a geometric series with ratio $\lambda = 0.5$. We can show that the expected value of geometric series with parameter λ is $\frac{1}{\lambda}$. Hence, the expected number of children in a family in town B, $E[X] = \frac{1}{0.5} = 2$.

This is, in fact, also easy to compute, if you don't know the formula by heart. There are multiple ways to prove it, and is left as an exercise.

ii. Let X be number of boy children and Y be number of girl children in a town.

- Town A: Let there be m families in town A. Since it is equally likely to have a boy child or a girl child, and each family has only one child, $E[X] = E[Y] = \frac{m}{2}$, and the boy to girl ratio is $E[X] : E[Y] = 1 : 1$.

- Town B: Let there be n families in town B. Since each family stops having children when a boy child is born and not earlier, there is a boy child in every family. So, $E[X] = n$.
Let us compute $E[Y]$, the expected number of girl children in town B:

$$
\begin{aligned}
E[Y] &= n\left[\sum_i i \cdot \Pr(Y = i)\right] \\
&= n\left[0 \times \frac{1}{2} + 1 \times \left(\frac{1}{2}\right)^2 + \ldots\right] \\
&= n \qquad \text{(proof is easy and is left as an exercise)}
\end{aligned}
$$

So, we see that for town B, $E[X] = E[Y] = n$. Hence the boy to girl ratio in town B is also $E[X] : E[Y] = 1 : 1$.