# CS534 — Written Homework Assignment 2 — Due Oct 20th 11:59pm, 2018

Please submit electronically via TEACH. Your submission should be a single PDF file.

1. (Maximum likelihood estimation.) In DNA, also known as the Code of Life, there exist four different possible bases: adenine (abbreviated A), cytosine (C), guanine (G) and thymine (T). We are given an organism of unknown DNA base frequencies. Let $p_a, p_c, p_g$, and $p_t$ be those unknown frequencies. Assume that we have obtained a strand of DNS sequences and we want to estimate the unknown frequencies. Let $n_a, n_c, n_g, n_t$ be the corresponding number of bases that you observe for A, C, T and G respectively. Please derive the maximum likelihood estimates for the unknown parameters $p_a, p_c, p_g$, and $p_t$.
   Hint: it it important to remember that $p_a + p_c + p_g + p_t = 1$. You can incorporate this constraint into the optimization using the lagrangian. If you are not familiar with this concept, here is a blog post that gives a good explanation
   `https://medium.com/@andrew.chamberlain/a-simple-explanation-of-why-lagrange-multipliers-works-253e2cdcbf74`.

   *First, let's write down the log-likelihood function:*

$$\log\left(p_a^{n_a} \times p_c^{n_c} \times p_g^{n_g} \times p_t^{n_t}\right)$$

$$= n_a \log p_a + n_c \log p_c + n_g \log p_g + n_t \log p_t$$

   *Given the constraint that $p_a + p_c + p_g + p_t = 1$, we introduce the following lagrangian:*

$$l(p_a, p_c, p_g, p_t) = n_a \log p_a + n_c \log p_c + n_g \log p_g + n_t \log p_t - \lambda(p_a + p_c + p_g + p_t - 1)$$

   *Taking the partial derivative w.r.t each parameter and set it to zero, we get:*

$$\frac{\partial l}{\partial p_a} = \frac{n_a}{\lambda}$$

$$\frac{\partial l}{\partial p_c} = \frac{n_c}{\lambda}$$

$$\frac{\partial l}{\partial p_g} = \frac{n_g}{\lambda}$$

$$\frac{\partial l}{\partial p_t} = \frac{n_t}{\lambda}$$

   *Plug these into $p_a + p_c + p_g + p_t = 1$, we have $\lambda = n_a + n_c + n_g + n_t$. The MLE estimates are:*

$$\hat{p_a} = \frac{n_a}{n_a + n_c + n_g + n_t}$$

$$\hat{p_c} = \frac{n_c}{n_a + n_c + n_g + n_t}$$

$$\hat{p_g} = \frac{n_g}{n_a + n_c + n_g + n_t}$$

$$\hat{p_t} = \frac{n_t}{n_a + n_c + n_g + n_t}$$

2. (Naive Bayes Classifier) Consider the following training set:

| A | B | C | Y |
|---|---|---|---|
| 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 |

(a) Learn a Naive Bayes classifier by estimating all necessary probabilities.
*Below are the list of probabilities that are learned from the training data.*
*Class prior:* $p(y=1)=1/2$
*Class conditional probability for* $y=1$*:*
$p(A=0|y=1)=1/3$; $p(B=0|y=1)=1/3$; $p(C=0|y=1)=2/3$
*Class conditional probability for* $y=0$*:*
$p(A=0|y=0)=2/3$; $p(B=0|y=0)=1/3$; $p(C=0|y=0)=1/3$

(b) Compute the probability $P(y=1|A=1,B=0,C=0)$. *Prediction for (1,0,0):*

$$p(y=1|X)=\frac{p(y=1)*P(A=1|y=1)P(B=0|y=1)P(C=0|y=1)}{P(A=1,B=0,C=0)}=\frac{0.5*\frac{2}{3}*\frac{1}{3}*\frac{2}{3}}{Z}=\frac{\frac{2}{27}}{Z}$$

$$p(y=0|X)=\frac{p(y=0)*P(A=1|y=0)P(B=0|y=0)P(C=0|y=0)}{P(A=1,B=0,C=0)}=\frac{0.5*\frac{1}{3}*\frac{1}{3}*\frac{1}{3}}{Z}=\frac{\frac{1}{54}}{Z}$$

*Because* $p(y=1|X)+p(Y=0|X)=1$*, we must have:*

$$Z=\frac{2}{27}+\frac{1}{54}=\frac{5}{54}$$

*Note that this is essentially calculating*

$$P(A=1,B=0,C=0)=P(A=1,B=0,c=0|y=1)*P(y=1)+P(A=1,B=0,C=0|y=0)*P(y=0)$$

*This gives us:*

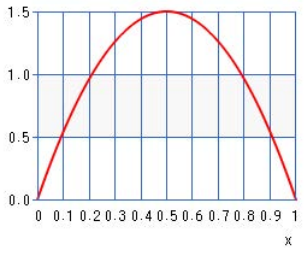
$$p(y=1|X)=\frac{4}{5};p(y=0|X)=\frac{1}{5}$$

*Note that you may notice that one can also directly make prediction* $y=0$ *without calculating the normalization factor* $Z$ *by simply noticing that no matter what* $Z$ *value is,* $p(y=1|X)$ *is greater than* $p(y=0|X)$*.*

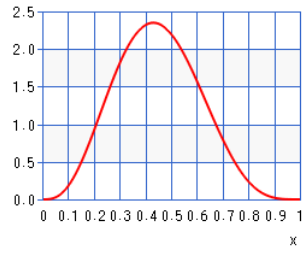
(c) Suppose we know that A, B and C are independent random variables, can we say that the Naive Bayes assumption is valid? (Note that the particular data set is irrelevant for this question). If your answer is yes, please explain why; if you answer is no please give an counter example.
*No.* $p(A,B,C)=p(A)p(B)p(C)$ *does not imply* $p(A,B,C|y)=p(A|y)p(B|y)p(C|y)$*.*

3. (Maximum A Posterior Estimation.) As discussed in class, consider using a beta prior $Beta(2,2)$ for estimating $p$, the probability of head for a weighted coin. What is the posterior distribution of $p$ after we observe 5 coin tosses and 2 of them are head? What is the posterior distribution of $p$ after we observe 50 coin tosses and 20 of them are head? Plot the pdf function of these two posterior distributions. Assume that $p=0.4$ is the true probability, as we observe more and more coin tosses from this coin, what do you expect to happen to the posterior?

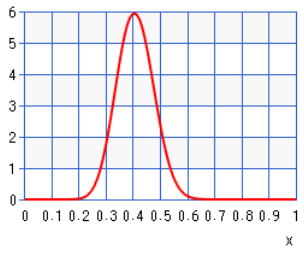
*With prior* $Beta(2,2)$ *and an observation of* $n_1$ *heads and* $n_0$ *tails, the posterior for* $p$ *is* $Beta(2+n_1,2+n_0)$*. So after observing 5 coin tosses with 2 heads, the posterior of* $p$ *becomes* $Beta(4,5)$*. With 50 tosses and 20 heads, the posterior becomes* $Beta(22,32)$*. You can see the pdf functions of the prior, and the two posteriors as follows.*




*(a)* $Beta(2,2)$




*(b)* $Beta(4,5)$




*(c)* $Beta(22,32)$

*As can be seen from the figure, the posterior becomes more and more peaked as we increase the observations. Eventually, all of the probability will concentrate at the true $p$ value. This is one nice property about the Bayesian approach: when we have very little data, we can fall back onto the prior to avoid catastrophic choices, and as we have more and more data they start to take over and the influence of the prior becomes increasingly neglectable.*

4. (Perceptron) The perceptron algorithm will only converge if the data is linearly separable. It is possible to *force* your data to be linearly separable as follows. If you have $N$ data points in $D$ dimensions, map data point $\vec{x}_n$ to the $(D+N)$-dimensional point $\langle \vec{x}_n, e_n \rangle$, where $e_n$ is a $N$-dimensional vector of all zeros but one 1 at the $n$th position. (Eg., $e_4 = \langle 0, 0, 0, 1, 0, \ldots \rangle$.)

   (a) Show that if you apply this mapping the data becomes linearly separable (you may wish to do so by providing a weight vector $\vec{w}$ in $(D+N)$-dimensional space that successfully separates the data).
   *You can construct a weight vector that are all zeros on the first D dimensional space. For the remaining N weights, make it positive one if it corresponds to a positive instance, otherwise, make it negative one.*

   (b) How does this mapping affect generalization?
   *This mapping essentially allows us to memorize the label for each instance, and does not give us any generalization ability.*

5. (Kernels) Cubic Kernels. In class, we showed that the quadratic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^2$ was equivalent to mapping each $\mathbf{x} = (x_1, x_2) \in R^2$ into a higher dimensional space where

$$\Phi(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

   Now consider the cubic kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^3$. What is the corresponding $\Phi$ function?
   *Let $\mathbf{x}_i = (x_{i1}, x_{i2})$ and $\mathbf{x}_j = (x_{j1}, x_{j2})$. Then the kernel is*

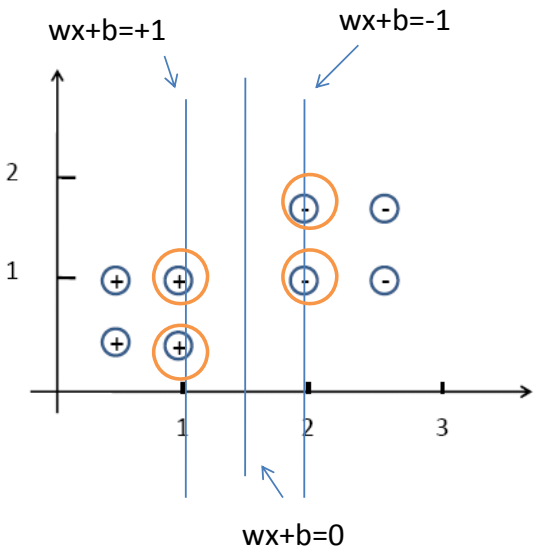
$$\begin{aligned}
K(\mathbf{x}_i, \mathbf{x}_j) &= (x_{i1}x_{j1} + x_{i2}x_{j2} + 1)^3 \\
&= (x_{i1}x_{j1} + x_{i2}x_{j2} + 1)^2 \cdot (x_{i1}x_{j1} + x_{i2}x_{j2} + 1) \\
&= (x_{i1}^2x_{j1}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2} + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2} + 1) \cdot (x_{i1}x_{j1} + x_{i2}x_{j2} + 1) \\
&= x_{i1}^3x_{j1}^3 + 3x_{i1}^2x_{j1}^2 + 3x_{i1}x_{j1} + \\
&\quad 3x_{i1}^2x_{j1}^2x_{i2}x_{j2} + 6x_{i1}x_{j1}x_{i2}x_{j2} + 3x_{i1}x_{j1}x_{i2}^2x_{j2}^2 + \\
&\quad 3x_{i2}x_{j2} + 3x_{i2}^2x_{j2}^2 + x_{i2}^3x_{j2}^3 + 1 \\
&= (x_{i1}^3, \sqrt{3}x_{i1}^2, \sqrt{3}x_{i1}, \sqrt{3}x_{i1}^2x_{i2}, \sqrt{6}x_{i1}x_{i2}, \sqrt{3}x_{i1}x_{i2}^2, \sqrt{3}x_{i2}, \sqrt{3}x_{i2}^2, x_{i2}^3, 1) \cdot \\
&\quad (x_{j1}^3, \sqrt{3}x_{j1}^2, \sqrt{3}x_{j1}, \sqrt{3}x_{j1}^2x_{j2}, \sqrt{6}x_{j1}x_{j2}, \sqrt{3}x_{j1}x_{j2}^2, \sqrt{3}x_{j2}, \sqrt{3}x_{j2}^2, x_{j2}^3, 1)
\end{aligned}$$

   *Hence, the function $\Phi(\mathbf{x}) = (x_1^3, \sqrt{3}x_1^2, \sqrt{3}x_1, \sqrt{3}x_1^2x_2, \sqrt{6}x_1x_2, \sqrt{3}x_1x_2^2, \sqrt{3}x_2, \sqrt{3}x_2^2, x_2^3, 1)$*

6. (Linear SVM) Apply linear SVM without soft margin to the following problem. Note that the two right most positive points are $(1, 0.5)$ and $(1, 1)$. The two left most negative points are $(2, 1)$ and $(2, 1.5)$.

   a. Please mark out the support vectors, the decision boundary ($\mathbf{w}^T\mathbf{x} + b = 0$) and $\mathbf{w}^T\mathbf{x} + b = 1$ and $\mathbf{w}^T\mathbf{x} + b = -1$. Note that you don't need to solve the optimization problem for this, just eyeball the solution.
   *See the figure.*

   b. Please solve for $\mathbf{w}$ and $b$ based on the support vectors you identified in (a).
   *Note that we can tell the equation for the line is in the form of $wx_1 + b = 0$, assuming the $x$ axis represents feature $x_1$. We can see that the point $(1.5, 0)$ lie on this line. Thus we can plug in $(1.5, 0)$ into the $wx_1 + b = 0$ equation, obtaining $b = -1.5w$. Also note that point $(1, 1)$ lies on line $wx_1 + b = 1$. Plug in $(1, 1)$ and $b = -1.5w$, we have $w - 1.5w = 1$, thus $w = -2$. The final solution is thus: $\mathbf{w} = [-2, 0]^T$, which is perpendicular to the decision boundary and $b = 3$.*


wx+b=+1
wx+b=-1
2
1
1
2
3
wx+b=0