CS182, Spring 2022
Homework 3

(Due Thursday, Apr. 22 at 11:59pm (CST))

1. [15 points] Given a Bayesian network (Fig. 1) with five discrete variables $\{F, A, S, H, N\}$, where $\{F, A, S, H, N\}$ are boolean variables. Suppose that $\{F, A, H, N\}$ are observed variables and $\{S\}$ is a latent variable. Now we implement EM algorithm for this model.

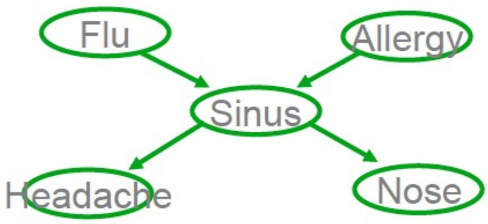

Figure 1: The Bayesian network with five discrete variables $\{F, A, S, H, N\}$.

(a) Derive the E-step. [5 points]

Solution: In E-step, calculate $P(S|F, A, H, N, \theta)$.

$$P(s_k = 0|f_k, a_k, h_k, n_k, \theta) = \frac{P(s_k = 0, f_k, a_k, h_k, n_k|\theta)}{\sum_{i=0}^{1} P(s_k = i, f_k, a_k, h_k, n_k|\theta)},$$

$$P(s_k = 1|f_k, a_k, h_k, n_k, \theta) = \frac{P(s_k = 1, f_k, a_k, h_k, n_k|\theta)}{\sum_{i=0}^{1} P(s_k = i, f_k, a_k, h_k, n_k|\theta)},$$

(b) Derive the M-step. [5 points]

Solution: In M-step, choose θ' which maximize $E_{P(S|F,A,H,N,\theta)} \log P(S, H, F, A, N|\theta')$, where

$E_{P(S|F,A,H,N,\theta)} \log P(S, H, F, A, N|\theta')$

$$= \sum_{k=1}^{K} \sum_{i=0}^{1} P(s_k = i|f_k, a_k, h_k, n_k, \theta)[\log P(f_k) + \log P(a_k) + \log P(s_k|f_k, a_k) + \log P(h_k|s_k) + \log P(n_k|s_k)].$$

(c) Guess the solution of parameter estimation in the M-step, according to the MLE solution with all variables being observed. [5 points]

Solution: The solutions of MLE are:

$$\theta_f = \frac{\sum_{k=1}^{K} \delta(f_k = 1)}{K},$$

$$\theta_a = \frac{\sum_{k=1}^{K} \delta(a_k = 1)}{K},$$

$$\theta_{s|f,a} = \frac{\sum_{k=1}^{K} \delta(s_k = s, f_k = f, a_k = a)}{\sum_{k=1}^{K} \delta(f_k = f, a_k = a)},$$

$$\theta_{h|s} = \frac{\sum_{k=1}^{K} \delta(h_k = 1, s_k = s)}{\sum_{k=1}^{K} \delta(s_k = s)},$$

$$\theta_{n|s} = \frac{\sum_{k=1}^{K} \delta(n_k = 1, s_k = s)}{\sum_{k=1}^{K} \delta(s_k = s)}.$$

By replacing $\delta(\cdot)$ by $P(\cdot)$ for the unobserved variables $\{S\}$, we have the solutions of the M-step in the EM algorithm:

$$
\begin{aligned}
\theta_f &= \frac{\sum_{k=1}^{K} \delta(f_k = 1)}{K}, \\
\theta_a &= \frac{\sum_{k=1}^{K} \delta(a_k = 1)}{K}, \\
\theta_{s|f,a} &= \frac{\sum_{k=1}^{K} P(s_k = s)\delta(f_k = f, a_k = a)}{\sum_{k=1}^{K} \delta(f_k = f, a_k = a)}, \\
\theta_{h|s} &= \frac{\sum_{k=1}^{K} \delta(h_k = 1)P(s_k = s)}{\sum_{k=1}^{K} P(s_k = s)}, \\
\theta_{n|s} &= \frac{\sum_{k=1}^{K} \delta(n_k = 1)P(s_k = s)}{\sum_{k=1}^{K} P(s_k = s)}.
\end{aligned}
$$

2. [20 points] Suppose two data points($x_1 = 0,\ x_2 = 1$) are generated from two Gaussian mixture model(A and B).The parameter of the two Gaussian model are unknown. We want to use EM to guess parameters of the two Gaussian models. For simplicity, the priors are set to equal, which means $P(a) = P(b) = \frac{1}{2}$. EM can be divided into following steps:

   (a) Randomly choose (μ_a, σ_a^2) and (μ_b, σ_b^2).

   (b) For each point x_i, calculate $P(a|x_i)$ and $P(b|x_i)$.

   (c) Adjust (μ_a, σ_a^2) and (μ_b, σ_b^2).

   (d) Repeat 2 and 3 until convergence.

   Suppose we randomly choose parameters as: $(\mu_a, \sigma_a^2) = (0, 1)$ and $(\mu_b, \sigma_b^2) = (1, 1)$

   (1) E-step: calculate $P(a|x_i)$ and $P(b|x_i)$, $i = 1, 2$.[10 points]
   Solution:

$$
\begin{aligned}
P(x_1|a) &= \frac{1}{\sqrt{2\pi}}e^0 = \frac{1}{\sqrt{2\pi}} \\
P(x_1|b) &= \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}} \\
a_1 = P(a|x_1) &= \frac{P(x_1|a)P(a)}{P(x_1|a)P(a) + P(x_1|b)P(b)} = \frac{\sqrt{e}}{1+\sqrt{e}} \\
b_1 = 1 - a_1 &= \frac{1}{1+\sqrt{e}} \\
P(x_2|a) &= \frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}} \\
P(x_2|b) &= \frac{1}{\sqrt{2\pi}}e^0 = \frac{1}{\sqrt{2\pi}} \\
a_2 = P(a|x_2) &= \frac{P(x_2|a)P(a)}{P(x_2|a)P(a) + P(x_2|b)P(b)} = \frac{1}{1+\sqrt{e}} \\
b_2 = 1 - a_2 &= \frac{\sqrt{e}}{1+\sqrt{e}}
\end{aligned}
$$

   (2) M-step: Adjust (μ_a, σ_a^2) and (μ_b, σ_b^2) with following formula.[10 points]

$$
\mu_a = \frac{a_1x_1 + a_2x_2}{a_1 + a_2}, \qquad \sigma_a^2 = \frac{a_1(x_1-\mu_a)^2 + a_2(x_2-\mu_a)^2}{a_1+a_2}
$$

   where $a_1 = P(a|x_1)$ and $a_2 = P(a|x_2)$
   Solution:

$$
\begin{aligned}
\mu_a &= \frac{a_1x_1 + a_2x_2}{a_1 + a_2} = \frac{1}{1+\sqrt{e}} \\
\sigma_a^2 &= \frac{a_1(x_1-\mu_a)^2 + a_2(x_2-\mu_a)^2}{a_1+a_2} = \frac{\sqrt{e}}{(1+\sqrt{e})^2} \\
\mu_b &= \frac{b_1x_1 + b_2x_2}{b_1 + b_2} = \frac{\sqrt{e}}{1+\sqrt{e}} \\
\sigma_b^2 &= \frac{b_1(x_1-\mu_b)^2 + b_2(x_2-\mu_b)^2}{b_1+b_2} = \frac{\sqrt{e}}{(1+\sqrt{e})^2}
\end{aligned}
$$

Table 1: The training data in (a).

i	x_{i1}	x_{i2}	y_i
1	1.5	0.5	1
2	2.5	1.5	1
3	3.5	3.5	1
4	6.5	5.5	1
5	7.5	10.5	1
6	1.5	2.5	-1
7	3.5	1.5	-1
8	5.5	5.5	-1
9	7.5	8.5	-1
10	1.5	10.5	-1

3. [30 points] Suppose that we are interested in learning a classifier, such that at any turn of a game we can pose a question, like "should I attack this ant hill now?", and get an answer.That is, we want to build a classifier which we can feed some features on the current game state, and get the output "attack" or "don't attack". There are many possible ways to define what the action "attack" means, but for now let's define it as sending all friendly ants that can see the ant hill under consideration towards it.

Let's recall the AdaBoost algorithm described in class. Its input is a dataset $\{(x_i, y_i)\}_{i=1}^n$, with x_i being the i-th sample, and $y_i \in \{-1, 1\}$ denoting the i-th label, $i = 1, 2, ..., n$. The features might be composed of a count of the number of friendly ants that can see the ant hill under consideration, and a count of the number of enemy ants these friendly ants can see. For example, if there were 10 friendly ants that could see a particular ant hill, and 5 enemy ants that the friendly ants could see, we would have:

$$x_1 = \begin{bmatrix} 10 \\ 5 \end{bmatrix}.$$

The label of the example x_1 is $y_1 = 1$, once the friendly ants were successful in razing the enemy ant hill, and $y_1 = 0$ otherwise. We could generate such examples by running a greedy bot (or any other opponent bot) against a bot that we make periodically try to attack an enemy ant hill. Each time this bot tries the attack, we record (say, after 20 turns or some other significant amount of time) whether the attack was successful or not.

(a) Let ϵ_t denote the error of a weak classifier h_t:

$$\epsilon_t = \sum_{i=1}^{n} D_t(i)\mathbb{1}(y_i \neq h_t(x_i)).$$

In the simple "attack" / "don't attack" scenario, suppose that we have implemented the following six weak classifiers:

$$h^{(1)}(x_i) = 2 * \mathbb{1}(x_{i1} \geq 2) - 1, \qquad h^{(4)}(x_i) = 2 * \mathbb{1}(x_{i2} \leq 2) - 1,$$
$$h^{(2)}(x_i) = 2 * \mathbb{1}(x_{i1} \geq 6) - 1, \qquad h^{(5)}(x_i) = 2 * \mathbb{1}(x_{i2} \leq 6) - 1,$$
$$h^{(3)}(x_i) = 2 * \mathbb{1}(x_{i1} \geq 10) - 1, \qquad h^{(6)}(x_i) = 2 * \mathbb{1}(x_{i2} \leq 10) - 1.$$

Given ten training data points ($n = 10$) as shown in Table 1, please show that what is the minimum value of ϵ_1 and which of $h^{(1)}, ..., h^{(6)}$ achieve this value? Note that there may be multiple classifiers that all have the same ϵ_1. You should list all classifiers that achieve the minimum ϵ_1 value. [6 points]

Solution:
The value of ϵ_1 for each of the classifiers is: $\frac{4}{10}, \frac{4}{10}, \frac{5}{10}, \frac{4}{10}, \frac{4}{10}$, and $\frac{5}{10}$. So, the minimum value is $\frac{4}{10}$ and classifiers 1, 2, 4, and 5 achieve this value.

(b) For all the questions in the remainder of this section, let h_1 denote $h^{(1)}$ chosen in the first round of boosting. (That is, $h^{(1)}$ was the classifier that achieved the minimum ϵ_1.)

(1) What is the value of α_1 (the weight of this first classifier h_1)? [2 points]
Solution:
Plugging into the formula for α we get: $\alpha_1 = \frac{1}{2} \ln \left(\frac{1 - \epsilon_1}{\epsilon_1} \right) = \frac{1}{2} \ln \frac{3}{2} = 0.2027$

(2) What should Z_t be in order to make sure the distribution D_{t+1} is normalized correctly? That is, derive the formula of Z_t in terms of ϵ_t that will ensure $\sum_{i=1}^{n} D_{t+1}(i) = 1$. Please aslo derive the formula of α_t in terms of ϵ_t. [6 points]
Solution:

$$
\begin{aligned}
Z_t &= \sum_{i=1}^{n} D_t(i) \exp(-\alpha_t y_i h_t(x_i)) \\
&= \sum_{i: y_i \neq h_t(x_i)} D_t(i) \exp(\alpha_t) + \sum_{i: y_i = h_t(x_i)} D_t(i) \exp(-\alpha_t) \\
&= \epsilon_t \exp(\alpha_t) + (1 - \epsilon_t) \exp(-\alpha_t)
\end{aligned}
$$

Let

$$
\begin{aligned}
\frac{\partial Z_t}{\partial \alpha_t} &= 0 \\
\epsilon_t \exp(\alpha_t) &= (1 - \epsilon_t) \exp(-\alpha_t) \\
\alpha_t + \ln(\epsilon_t) &= -\alpha_t + \ln(1 - \epsilon_t) \\
\alpha_t &= \frac{1}{2} \ln\left(\frac{1 - \epsilon_t}{\epsilon_t}\right)
\end{aligned}
$$

So

$$
Z_t = 2\sqrt{\epsilon_t(1 - \epsilon_t)}
$$

(3) Which points will increase in significance in the second round of boosting? That is, for which points will we have $D_1(i) < D_2(i)$? What are the values of D_2 for these points? [5 points]
Solution:
The points that $h^{(1)}$ misclassifies will increase in weight. These are the points $i = 1, 7, 8, 9$ from the data table. Their new weight under D_2 will be:

$$
\begin{aligned}
D_2(i) &= \frac{D_1(i) \exp(-\alpha_1 y_i h_1(x_i))}{Z_1} \\
&= \frac{\exp\{0.2027\}}{4 * \exp\{0.2027\} + 6 * \exp\{-0.2027\}} \\
&= \frac{1}{8}
\end{aligned}
$$

(4) In the second round of boosting, the weights on the points will be different, and thus the error ϵ_2 will also be different. Which of $h^{(1)}, ..., h^{(6)}$ will minimize ϵ_2? (Which classifier will be selected as the second weak classifier h_2?) What is its value of ϵ_2? [6 points]
Solution:
$h^{(4)}$ will be chosen.

Classifier	ϵ_2
$h^{(1)}$	$1/2$
$h^{(2)}$	$2/8 + 2/12 = 5/12$
$h^{(3)}$	$1/8 + 4/12 = 11/24$
$h^{(4)}$	$1/8 + 3/12 = 3/8$
$h^{(5)}$	$2/8 + 2/12 = 5/12$
$h^{(6)}$	$3/8 + 2/12 = 13/24$

(5) What will the average error of the final classifier H be, if we stop after these two rounds of boosting? That is, if $H(x) = \text{sign}(\alpha_1 h_1(x) + \alpha_2 h_2(x))$, what will the training error $\epsilon = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(y_i \neq h(x_i))$ be? Is this more, less, or the same as the error we would get, if we just used one of the weak classifiers instead of this final classifier H? [5 points]
Solution:
The classifier after two rounds is:

$$
H(x) = sign\left(\frac{1}{2} \ln\left(\frac{3}{2}\right) h_1(x) + \frac{1}{2} \ln\left(\frac{5}{3}\right) h_2(x)\right)
$$

Since $\ln\left(\frac{5}{3}\right) > \ln\left(\frac{3}{2}\right)$ the classifier H will always go with the guess made by $h^{(4)}$. So, it is the same as the error we could get using a single weak classifier, $\epsilon = \frac{4}{10}$. More rounds of boosting are necessary before the interplay of specific settings of the α becomes relevant and allows us to do better than a single weak classifier.

4. [20 points] Given valid kernels $k_1(\mathbf{x}, \mathbf{x}')$ and $k_2(\mathbf{x}, \mathbf{x}')$, please verify the following new kernels will also be valid:

   (a) $k(\mathbf{x}, \mathbf{x}') = f(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') f(\mathbf{x}')$, where $f(\cdot)$ is any function. [6 points]

   (b) $k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}'))$, where $q(\cdot)$ is a polynomial with nonnegative coefficients. [6 points]

   (c) $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A} \mathbf{x}'$, where $\mathbf{A}$ is a symmetric positive semi-definite matrix. [8 points]

   Solution:

   (a) Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, there must exist a feature vector $\phi(\mathbf{x})$ such that

$$k_1(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}').$$

   Then we can rewrite the given kernel as

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= f(\mathbf{x}) \phi(\mathbf{x})^\top \phi(\mathbf{x}') f(\mathbf{x}') \\ &= \mathbf{v}(\mathbf{x})^\top \mathbf{v}(\mathbf{x}'), \end{aligned}$$

   where $\mathbf{v}(\mathbf{x}) \triangleq f(\mathbf{x}) \phi(\mathbf{x})$. We can see that the kernel can be rewritten as the scalar product of feature vectors, and hence is a valid kernel.

   (b) Suppose $q(x) = \sum_{i=1}^{n} a_n x^n, \forall a_n \geq 0$, then the kernel can be expressed as

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{n} a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n.$$

   We focus on the i-th term of the kernel, which is $a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n$. Since $k_1(\mathbf{x}, \mathbf{x}')$ is a valid kernel, the product of kernels is also a valid kernel. Hence, $a_n \left(k_1(\mathbf{x}, \mathbf{x}')\right)^n$ is a valid kernel. With the fact that the sum of kernels is a valid kernel, the original kernel is valid.

   (c) Since $\mathbf{A}$ is a symmetric positive semi-definite matrix, we can decompose $\mathbf{A}$ as $\mathbf{A} = \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top$ where $\mathbf{Q}$ is an orthogonal matrix and $\boldsymbol{\Lambda}$ is a diagonal matrix. When $\mathbf{A}$ is positive semi-definite, the entries of $\boldsymbol{\Lambda}$ are nonnegative. Hence, we can rewrite the kernel as

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^\top \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top \mathbf{x}' \\ &= (\boldsymbol{\Lambda}^{1/2} \mathbf{Q}^\top \mathbf{x})^\top (\boldsymbol{\Lambda}^{1/2} \mathbf{Q}^\top \mathbf{x}') \\ &= \boldsymbol{\Phi}(\mathbf{x})^\top \boldsymbol{\Phi}(\mathbf{x}'), \end{aligned}$$

   where $\boldsymbol{\Phi}(\mathbf{x}) \triangleq \boldsymbol{\Lambda}^{1/2} \mathbf{Q}^\top \mathbf{x}$. We can see that the kernel can be rewritten as the scalar product of feature vectors, and hence is a valid kernel.

5. [20 points] We have learned that when solving a SVM problem, we need to first construct Lagrangian function $L(w, b, \alpha)$ and set partial derivative to zero. By using KKT conditions, we can get the dual problem of SVM :

$$\min_{\alpha} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j < x_i, x_j > -\sum_{i=1}^{n}\alpha_i,$$

$$s.t.\ \alpha_i \geq 0, \quad \sum_{i=1}^{n}\alpha_i y_i = 0$$

Now we use a simple example to better understand how SVM works. We consider the separating hyperplane being $wx + b = 0$. Suppose we have three data points: $x_1 = (2, -1)^T, x_2 = (2, -3)^T, x_3 = (4, -1)^T$, the corresponding labels are: $y_1 = -1, y_2 = -1, y_3 = 1$. Use SVM to find the the values of $w^* = (w_1^*, w_2^*)$, b^* and give the separating hyperplane. Please show your calculation process.

Solution:

Lagrangian function $L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{3}\alpha_i(y_i(wx_i + b) - 1)$. Set $\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial b} = 0$, we can get $w = \sum_{i=1}^{3}\alpha_i y_i x_i$ and $\sum_{i=1}^{3}\alpha_i y_i = 0$. Construct the dual problem:

$$\begin{aligned}
&\min_{\alpha} \frac{1}{2}\sum_{i=1}^{3}\sum_{j=1}^{3}\alpha_i\alpha_j y_i y_j < x_i, x_j > -\sum_{i=1}^{3}\alpha_i \\
&= \frac{1}{2}(5\alpha_1^2 + 13\alpha_2^2 + 17\alpha_3^2 + 14\alpha_1\alpha_2 - 18\alpha_1\alpha_3 - 22\alpha_2\alpha_3) - \alpha_1 - \alpha_2 - \alpha_3 \\
&s.t.\ \alpha_i \geq 0, \quad -\alpha_1 - \alpha_2 + \alpha_3 = 0
\end{aligned}$$

Represent α_3 by α_1 and α_2, we can get $f(\alpha_1, \alpha_2) = 2\alpha_1^2 + 4\alpha_2^2 + 4\alpha_1\alpha_2 - 2\alpha_1 - 2\alpha_2$. To solve α_1, α_2, we set $\frac{\partial f}{\partial \alpha_1} = 0, \frac{\partial f}{\partial \alpha_2} = 0$, we can get $\alpha_1 = \frac{1}{2}, \alpha_2 = 0$. Then $\alpha_3 = \alpha_1 + \alpha_2 = \frac{1}{2}$.

So $w^* = \sum_{i=1}^{3}\alpha_i y_i x_i = (1, 0)$. Since we have $\alpha_i(y_i(wx_i + b) - 1) = 0$ in KKT conditions, then we can use either x_1 or x_3 to get $b^* = -3$. And the separating hyperplane is $x_1 - 3 = 0$.