

CS182, Spring 2022  
Homework 3  
(Due Tuesday, May. 24 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.

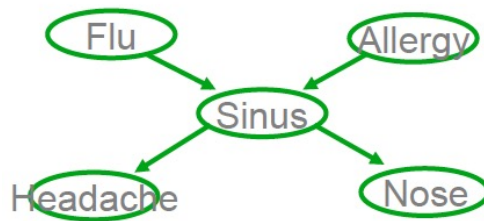


Figure 1: The Bayesian network with five discrete variables  $\{F, A, S, H, N\}$ .

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

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  $(\mu_a, \sigma_a^2)$  and  $(\mu_b, \sigma_b^2)$ .
- (b) For each point  $x_i$ , calculate  $P(a|x_i)$  and  $P(b|x_i)$ .
- (c) Adjust  $(\mu_a, \sigma_a^2)$  and  $(\mu_b, \sigma_b^2)$ .
- (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]
- (2) M-step: Adjust  $(\mu_a, \sigma_a^2)$  and  $(\mu_b, \sigma_b^2)$  with following formula. [10 points]

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

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

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,  $i = 1, 2, \dots, n$ . 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  $\epsilon_t$  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:

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

Given ten training data points ( $n = 10$ ) as shown in Table 1, please show that what is the minimum value of  $\epsilon_1$  and which of  $h^{(1)}, \dots, h^{(6)}$  achieve this value? Note that there may be multiple classifiers that all have the same  $\epsilon_1$ . You should list all classifiers that achieve the minimum  $\epsilon_1$  value. [6 points]

- (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  $\epsilon_1$ .)

- (1) What is the value of  $\alpha_1$  (the weight of this first classifier  $h_1$ )? [2 points]

- (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  $\epsilon_t$  that will ensure  $\sum_{i=1}^n D_{t+1}(i) = 1$ . Please also derive the formula of  $\alpha_t$  in terms of  $\epsilon_t$ . [6 points]

- (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]
- (4) In the second round of boosting, the weights on the points will be different, and thus the error  $\epsilon_2$  will also be different. Which of  $h^{(1)}, \dots, h^{(6)}$  will minimize  $\epsilon_2$ ? (Which classifier will be selected as the second weak classifier  $h_2$ ?) What is its value of  $\epsilon_2$ ? [6 points]
- (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]

4. [15 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. [5 points]
  - (b)  $k(\mathbf{x}, \mathbf{x}') = q(k_1(\mathbf{x}, \mathbf{x}'))$ , where  $q(\cdot)$  is a polynomial with nonnegative coefficients. [5 points]
  - (c)  $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{A} \mathbf{x}'$ , where  $\mathbf{A}$  is a symmetric positive semi-definite matrix. [5 points]

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 values of  $w^* = (w_1^*, w_2^*)$ ,  $b^*$  and give the separating hyperplane. Please show your calculation process.