Introduction to Machine Learning, Fall 2023
Homework 3
(Due Tuesday Nov. 30 at 11:59pm (CST))

January 11, 2024

1. [15 points] [Expectation Maximization Algorithm] Consider a probabilistic model in which we collectively denote the observed variables by $\boldsymbol{X}$ and all of the hidden variables by $\boldsymbol{Z}$. The joint distribution $p(\boldsymbol{X}, \boldsymbol{Z}|\theta)$ is parameterized by θ. Our goal is to maximize the likelihood function given by

$$p(\boldsymbol{X}|\theta). \tag{1}$$

   (a) Given an arbitrary distribution q, show that the log-likelihood of $\boldsymbol{X}$ is [5 points]

$$\log p(\boldsymbol{X}|\theta) = \mathbb{E}_{\boldsymbol{Z}\sim q}\left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\theta)}{q(\boldsymbol{Z})}\right] + KL\left(q(\boldsymbol{Z})\|p(\boldsymbol{Z}|\boldsymbol{X}, \theta)\right). \tag{2}$$

   (b) Next let's consider the expectation step. First show the evidence lower bound (ELBO) is a lower bound of the log-likelihood, namely [5 points]

$$\log p(\boldsymbol{X}|\theta) \geq \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\theta^{(t-1)}}\left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\theta)}{p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)})}\right], \tag{3}$$

   where $\theta^{(t-1)}$ is the parameter estimated in the previous iteration.

   (c) We want to maximize the ELBO, $\mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\theta^{(t-1)}}\left[\log \frac{p(\boldsymbol{X},\boldsymbol{Z}|\theta)}{p(\boldsymbol{Z}|\boldsymbol{X},\theta^{(t-1)})}\right]$ since maximizing $p(\boldsymbol{X}|\theta)$ is hard. EM algorithm defines $Q(\theta|\theta^{(t-1)}) := \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\theta^{(t-1)}}\left[\log p(\boldsymbol{X}, \boldsymbol{Z}|\theta)\right]$. The M-step is given by:

$$\theta^{(t)} \leftarrow \arg\max_{\theta} Q(\theta|\theta^{(t-1)}). \tag{4}$$

   Show that maximizing $Q(\theta|\theta^{(t-1)})$ and maximizing the ELBO is equivalent. [5 points] Formally,

$$\arg\max_{\theta} Q(\theta|\theta^{(t-1)}) = \arg\max_{\theta} \mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X},\theta^{(t-1)}}\left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\theta)}{p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)})}\right] \tag{5}$$

Solution:

(a)

$$\begin{aligned}
\log p(\boldsymbol{X} \mid \theta) &= \log \frac{p(\boldsymbol{X}, \boldsymbol{Z} \mid \theta)}{p(\boldsymbol{X}, \boldsymbol{Z} \mid \theta)/p(\boldsymbol{X} \mid \theta)} \\
&= \log \frac{p(\boldsymbol{X}, Z \mid \theta)}{q(\boldsymbol{Z})} \frac{q(\boldsymbol{Z})}{p(\boldsymbol{Z} \mid \boldsymbol{X}, \theta)} \\
&= \underbrace{\int q(\boldsymbol{Z}) d\boldsymbol{Z}}_{\text{this equals to 1}} \left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z} \mid \theta)}{q(\boldsymbol{Z})} + \log \frac{q(\boldsymbol{Z})}{p(\boldsymbol{Z} \mid \boldsymbol{X}, \theta)} \right] \\
&= \int q(\boldsymbol{Z}) \log \frac{p(\boldsymbol{X}, \boldsymbol{Z} \mid \theta)}{q(\boldsymbol{Z})} d\boldsymbol{Z} + \int q(\boldsymbol{Z}) \log \frac{q(\boldsymbol{Z})}{p(\boldsymbol{Z} \mid \boldsymbol{X}, \theta)} d\boldsymbol{Z} \\
&= \underbrace{\int q(\boldsymbol{Z}) \log \frac{p(\boldsymbol{X}, \boldsymbol{Z} \mid \theta)}{q(\boldsymbol{Z})} d\boldsymbol{Z}}_{\text{Evidence Lower Bound(ELBO)}} + KL(q(\boldsymbol{Z}) || p(\boldsymbol{Z} \mid \boldsymbol{X}, \theta))
\end{aligned}$$

(b) Since KL divergence is non-negative, so $\int q(Z \mid \theta) \log \frac{p(X,Z|\theta)}{q(Z)} dZ$ is a lower bound of $\log p(X \mid \theta)$. $q(\cdot) := p(\cdot \mid X, \theta^{(t-1)})$ yields the result.

(c)

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)}} \left[\log \frac{p(\boldsymbol{X}, \boldsymbol{Z}|\theta)}{p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)})} \right] &= \int p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)}) \log \frac{p(\boldsymbol{X}, \boldsymbol{Z} \mid \theta)}{p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)})} d\boldsymbol{Z} \\
&= Q(\theta|\theta^{(t-1)}) - \int p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)}) \log p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)}),
\end{aligned} \tag{6}$$

where $\int p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)}) \log p(\boldsymbol{Z}|\boldsymbol{X}, \theta^{(t-1)})$ is constant for the object. So the two optimization problems are equivalent.

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

2. [15 points] [Boosting] 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 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:

$$\epsilon_t = \sum_{i=1}^{n} D_t(i) \mathbb{1}(y_i \neq h_t(x_i)). \tag{7}$$

   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. [3 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 ϵ_1.)

   (1) What is the value of α_1 (the weight of this first classifier h_1)? [1 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 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. [3 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? [3 points]

(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? [3 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? [2 points]

Solution:

(a) 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) (1) 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)

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

Let

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

So

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

(3) 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:

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

(4) $h^{(4)}$ 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) The classifier after two rounds is:

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

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.

3. [10 points] [Perceptron Learning Algorithm] Consider a binary classification problem. The input space is $\mathbb{R}^d$. The output space is $\{+1, -1\}$. For simplicity, we modified the input to be $\mathbf{x} = [x_0, x_1, \cdots, x_d]^\intercal$ with $x_0 = 1$. The output is predicted using the hypothesis:

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\intercal \mathbf{x}), \tag{8}$$

where $\mathbf{w} = [w_0, w_1, \cdots, w_d]^\intercal$ and w_0 is the bias.

The *perceptron learning algorithm* determines $\mathbf{w}$ using a simple iterative method. Here is how it works. At iteration t, where $t = 0, 1, 2, \ldots$, there is a current value of the weight vector, call it $\mathbf{w}(t)$. The algorithm picks an example from $(\mathbf{x}_1, y_1) \cdots (\mathbf{x}_N, y_N)$ that is currently misclassified, call it $(\mathbf{x}(t), y(t))$, and uses it to update $\mathbf{w}(t)$. Since the example is misclassified, we have $y(t) \neq \text{sign}\left(\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)\right)$. The update rule is

$$\mathbf{w}(t+1) = \mathbf{w}(t) + y(t)\mathbf{x}(t). \tag{9}$$

(a) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t) < 0$. [Hint: $\mathbf{x}(t)$ is misclassified by $\mathbf{w}(t)$.] [3 points]

(b) Show that $y(t)\mathbf{w}^{\mathrm{T}}(t+1)\mathbf{x}(t) > y(t)\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}(t)$. [3 points]

(c) As far as classifying $\mathbf{x}(t)$ is concerned, argue that the move from $\mathbf{w}(t)$ to w$(t+1)$ is a move "in the right direction". [4 points]

Solution:

(a) If $x(t)$ is misclassified by $w(t)$, then $w^T(t)x(t)$ has different signs of $y(t)$, thus $y(t)w^T(t)x(t) < 0$.

(b)

$$\begin{aligned} y(t)w^T(t+1)x(t) &= y(t)\left(w(t) + y(t)x(t)\right)^T x(t) \\ &= y(t)\left(w^T(t) + y(t)x^T(t)\right)x(t) \\ &= y(t)w^T(t)x(t) + y(t)y(t)x^T(t)x(t) \\ &> y(t)w^T(t)x(t) \quad \text{because the last term is } \geq \text{ than } 0 \end{aligned}$$

(c) From previous problem, we see that $y(t)w^T(t)x(t)$ is increasing with each update.

If $y(t)$ is positive, but $w^T(t)x(t)$ is negative, we move $w^T(t)x(t)$ toward positive by increasing it.

If however $y(t)$ is negative, but $w^T(t)x(t)$ is positive, $y(t)w^T(t)x(t)$ increases means $w^T(t)x(t)$ is decreasing, i.e. moving toward negative region.

So the move from $w(t)$ to $w(t+1)$ is a move "in the right direction" as far as classifying $x(t)$ is concerned.