Assignment 4: Graphical Model and Unsupervised Learning

AIGS/CSED515 Machine Learning

Instructor: Jungseul Ok

jungseul@postech.ac.kr

Due: 20:00pm Dec 16, 2020

Remarks

- Group study and open discussion via LMS board are encouraged, however, assignment that your hand-in must be **of your own work**, and **hand-written** unless you're asked a coding task.
- Submit a scanned copy of your answer on LMS online in **a single PDF file**, to which you also print and add your code if apply, i.e., no zip file, just a single PDF containing everything.
- Delayed submission may get some penalty in score: 5% off for delay of $0 \sim 4$ hours; 20% off for delay of $4 \sim 24$ hours; and delay longer than 24 hours will not be accepted.

1. [6 pt] (An application of belief propagation) Consider an integer programming (IP) of $x_1, ..., x_5$ with linear objective and constraints in the followings:

$$\begin{aligned} \underset{x_1,\dots,x_5\in\{0,1\}}{\text{maximize}} \quad & x_1 + 2x_2 + 3x_3 + 2x_4 + 2x_5 \\ \text{subject to} \quad & x_1 + x_2 + x_3 \le 1 \\ & x_3 + x_4 \le 1 \\ & x_4 + x_5 \le 1 \end{aligned}$$

In order to solve the IP, we can formulate a maximum a posterior (MAP) problem of the joint probability of $x_1, ..., x_5$ in the following factorized form:

$$p(x_1, ..., x_5) = \frac{1}{Z} f_a(x_1) f_b(x_2) f_c(x_3) f_d(x_4) f_e(x_5) f_A(x_1, x_2, x_3) f_B(x_3, x_4) f_C(x_4, x_5) \,,$$

where Z is the normalization constant, $f_a(x_1) = \exp(x_1)$, $f_b(x_2) = \exp(2x_2)$, $f_c(x_3) = \exp(3x_3)$, $f_d(x_4) = \exp(2x_4)$, $f_e(x_5) = \exp(2x_5)$,

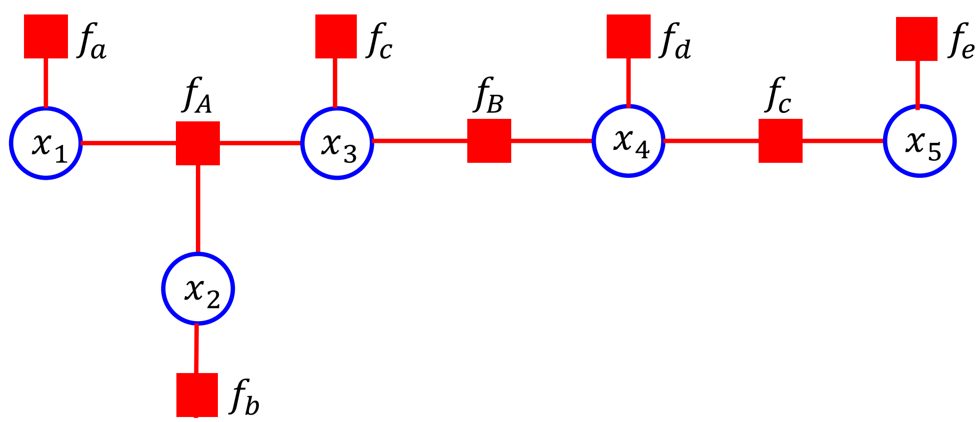
$$f_A(x_1, x_2, x_3) = \begin{cases} 1 & \text{if } x_1 + x_2 + x_3 \le 1 \\ 0 & \text{otherwise} \end{cases} ,$$

$$f_B(x_3, x_4) = \begin{cases} 1 & \text{if } x_3 + x_4 \le 1 \\ 0 & \text{otherwise} \end{cases} ,$$

$$f_C(x_4, x_5) = \begin{cases} 1 & \text{if } x_4 + x_5 \le 1 \\ 0 & \text{otherwise} \end{cases} .$$

Note that only configuration of $x_1, ..., x_5$ verifies all the constraints in the IP has non-zero probability, which is proportional to the exponential of the objective value of the IP. Hence, the MAP configuration is a solution to the integer programming.

(a) [2pt] <u>Draw</u> the factor graph corresponding to the joint probability $p(x_1, ..., x_5)$.
sol)



(b) [4pt] <u>Solve</u> the IP using the max-product belief propagation algorithm. <u>What</u> is the optimal value?
sol) The answer is $(0, 0, 1, 0, 1)$.

2. [11pt] (Graph learning) Consider 4 binary random variables $X_1, X_2, X_3, X_4 \in \{0, 1\}$. Assume we have 100 observations and the following table shows the number of counts of observations. We want to learn the structure of random variables X_i's from our observations using the Chow-Liu tree algorithm.

X_1	X_2	X_3	X_4	Count	X_1	X_2	X_3	X_4	Count
0	0	0	0	2	1	0	0	0	7
0	0	0	1	5	1	0	0	1	5
0	0	1	0	2	1	0	1	0	7
0	0	1	1	5	1	0	1	1	12
0	1	0	0	2	1	1	0	0	10
0	1	0	1	4	1	1	0	1	10
0	1	1	0	6	1	1	1	0	10
0	1	1	1	8	1	1	1	1	5

(a) [4pt] Compute the marginal probability $p(X_i)$ for each $i \in \{1, 2, 3, 4\}$ and $p(X_i, X_j)$ for all $i \neq j \in \{1, 2, 3, 4\}$.

x	0	1
$p(X_1 = x)$		
$p(X_2 = x)$		
$p(X_3 = x)$		
$p(X_4 = x)$		

(x, y)	$(0, 0)$	$(0, 1)$	$(1, 0)$	$(1, 1)$
$p(X_1 = x, X_2 = y)$				
$p(X_1 = x, X_3 = y)$				
$p(X_1 = x, X_4 = y)$				
$p(X_2 = x, X_3 = y)$				
$p(X_2 = x, X_4 = y)$				
$p(X_3 = x, X_4 = y)$				

sol)

x	0	1
$p(X_1 = x)$	0.34	0.66
$p(X_2 = x)$	0.45	0.55
$p(X_3 = x)$	0.45	0.55
$p(X_4 = x)$	0.46	0.54

sol)

(x, y)	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
$p(X_1 = x, X_2 = y)$	0.14	0.20	0.31	0.35
$p(X_1 = x, X_3 = y)$	0.13	0.21	0.32	0.34
$p(X_1 = x, X_4 = y)$	0.12	0.22	0.34	0.32
$p(X_2 = x, X_3 = y)$	0.19	0.26	0.26	0.29
$p(X_2 = x, X_4 = y)$	0.18	0.27	0.28	0.27
$p(X_3 = x, X_4 = y)$	0.21	0.24	0.25	0.30

(b) [2pt] Compute the mutual information $I(X_i, X_j)$ for all $i \neq j \in \{1, 2, 3, 4\}$.

$I(X_1, X_2)$	
$I(X_1, X_3)$	
$I(X_1, X_4)$	
$I(X_2, X_3)$	
$I(X_2, X_4)$	
$I(X_3, X_4)$	

sol) When using $\log_{10}$,

$I(X_1, X_2)$	0.006629
$I(X_1, X_3)$	0.002082
$I(X_1, X_4)$	0.005222
$I(X_2, X_3)$	0.000554
$I(X_2, X_4)$	0.002583
$I(X_3, X_4)$	0.000031

When using ln,

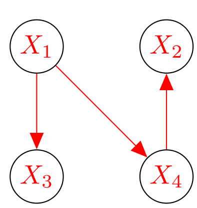
$I(X_1, X_2)$	0.001526
$I(X_1, X_3)$	0.004795
$I(X_1, X_4)$	0.012025
$I(X_2, X_3)$	0.001277
$I(X_2, X_4)$	0.005948
$I(X_3, X_4)$	0.000073

When using $\log_2$,

$I(X_1, X_2)$	0.002202
$I(X_1, X_3)$	0.006918
$I(X_1, X_4)$	0.017348
$I(X_2, X_3)$	0.001842
$I(X_2, X_4)$	0.008581
$I(X_3, X_4)$	0.000105

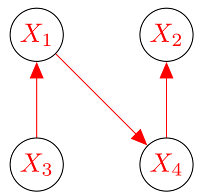
(c) [2pt] Performing Chow-Liu algorithm, draw a Bayesian network T_1 rooted from X_1.

sol)



(d) [2pt] Performing Chow-Liu algorithm, draw a Bayesian network T_3 rooted from X_3.

sol)



(e) [1pt] Let p_T denote the probability corresponding to Bayesian network T. Compute the difference $\mathrm{KL}(p\|p_{T_1}) - \mathrm{KL}(p\|p_{T_3})$ where Bayesian networks T_1 and T_3 are obtained in Problems 2c and 2d, resp.

sol) The difference is 0 (since both T_1 and T_3, which Chow-Liu algorithm output, minimize $\mathrm{KL}(p\|p_T)$ for all possible tree T.)

3. [12pt] (K-means) Given a dataset $\mathcal{D} = \{x^{(i)}\}_{i\in[N]}$ of N data points in $\mathbb{R}^2$, we want to partitioning them into K clusters using K-means algorithm. Let $\mu_k \in \mathbb{R}^2$ denote the center of cluster $k \in [K]$. Then, the K-means algorithm aims at optimizing:

$$\min_{\{r_{ik}\},\{\mu_k\}} \quad \sum_{i\in[N]} \sum_{k\in[K]} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2 \tag{1a}$$

$$\text{s.t.} \quad r_{ik} \in \{0,1\} \quad \forall i \in [N], \forall k \in [K] \quad \text{and;} \tag{1b}$$

$$\sum_{k\in[K]} r_{ik} = 1 \quad \forall i \in [N]\ . \tag{1c}$$

(a) [2pt] Given fixed cluster centers $\{\mu_k\}_{k\in[K]}$, <u>obtain</u> the optimal r_{ik} for (1). <u>Justify</u> your solution.

sol) Assigning $x^{(i)}$ to the closest cluster center μ_k in terms of L2 minimizes the part of loss (1a) from i, i.e., $\sum_{k\in[K]} \frac{1}{2} r_{ik} \|x^{(i)} - \mu_k\|_2^2$.

$$r_{ik} = \begin{cases} 1 & \text{if} \quad k = \arg\min_{k\in[K]} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$$

(b) [2pt] Given fixed $\{r_{ik}\}_{i\in[N],k\in[K]}$, verifying (1b) and (1c), <u>obtain</u> the optimal cluster center μ_k for (1). <u>Justify</u> your solution.

sol) Taking the derivative of (1a) w.r.t. μ_k and setting it to 0, we have

$$\sum_i r_{ik}(x^{(i)} - \mu_k) = 0\ ,$$

which concludes that the optimal μ_k is

$$\mu_k = \frac{\sum_i r_{ik} x^{(i)}}{\sum_i r_{ik}}\ .$$

(c) [4pt] We want to check the convergence of K-means algorithm which alternates Problems (3a) and (3b). <u>Describe</u> the algorithm. Let L_t be the loss (1a) after t-th iteration of K-means algorithm. <u>Check</u> if L_t is monotonically increasing in t. Using the following theorem (a part of monotone convergence theorem), <u>check</u> the convergence of K-means algorithm in terms of loss function. <u>Can</u> we guarantee that K-means algorithm converges to the global optimality?

Theorem 1. *If $(a_t)_{t\in\mathbb{N}}$ is a <u>monotone</u> sequence of real numbers, i.e., if $a_t \le a_{t+1}$ for every $t \ge 1$, or $a_t \ge a_{t+1}$ for every $t \ge 1$, then this sequence has a finite limit if and only if the sequence is <u>bounded</u>.*

sol)

- The algorithm begins with arbitrary choice of μ_k's, and iterates:

Step1. $r_{ik} = \begin{cases} 1 & \text{if} \;\; k = \arg\min_{k \in [K]} \|x^{(i)} - \mu_k\|_2^2 \\ 0 & \text{otherwise} \end{cases}$

Step2. $\mu_k = \frac{\sum_i r_{ik} x^{(i)}}{\sum_i r_{ik}}$.

- The L_t is monotonically "decreasing" as each step of K-means algorithm reduces the loss.
- In addition to monotonically decreasing L_t, it is straightforward to check that L_t is lower bounded by 0. Hence, from Theorem 1, L_t converges.
- However, we cannot guarantee the convergence to the global optimum since the algorithm alternates the optimization of non-smooth cost function (in particular, step1).

(d) [4pt] Complete `Kmeans.py` which performing K-means algorithm aforementioned. For the given dataset, after how many updates does the algorithm converge? What cost function value does it converge to? What are the obtained centers?

sol)

- See `sol_Kmeans.py`
- Convergence after 2 updates
- Convergence to a cost function value of 4.56
- Cluster centers are $(1.9, -1.9)$ and $(-2.1, 2.1)$

(The value may differ than the above depending on the version of Python)

4. [20pt] (Generative Adversarial Newtorks) Consider the following max-min problem for a dataset $\mathcal{D}$ consisting of x's:

$$\max_{\theta} \min_{w} -\sum_{x\in\mathcal{D}} \log p_w(y=1 \mid x) - \sum_{z\in\mathcal{Z}} \log(1-p_w(y=1 \mid G_\theta(z))) + \frac{C}{2}\|w\|_2^2 \,. \qquad (2)$$

Here the generator $G_\theta(z)$ parameterized by θ transforms noise $z \in \mathcal{Z}$ into artificial data. The discriminator $p_w(y \mid x)$ parameterized by w checks if x is artificial or not, where $y=1$ indicates that x is real, and $y=0$ indicates that x is artificial. The hyper-parameter $C \geq 0$ controls impact of regularization. Note that solving (2) is challenging mainly due to the objective is neither convex in w nor concave in θ in general. We will check if the cost function is is convex in w for specific choice of the discriminator model. To do so, we use several facts:

Fact1. A function $f(w)$ is convex in w if Hessian[1] of $f(w)$ is positive semi-definite[2].

Fact2. A sum of convex functions is also convex.

(a) [2pt] Suppose that we model the discriminator as follows:

$$p_w(y=1 \mid x) = \frac{1}{1+\exp(w^\top x)} \,.$$

Using this, write down the resulting cost function for (2).

sol)

$$\sum_{x\in\mathcal{D}} \log(1+\exp(w^\top x)) + \sum_{z\in\mathcal{Z}} \left(\log(1+\exp(w^\top G_\theta(z))) - w^\top G_\theta(z)\right) + \frac{C}{2}\|w\|_2^2$$

(b) [2pt] Obtain Hessian of (A) $= \frac{C}{2}\|w\|_2^2 - w^\top b$ in w. Check if (A) is convex, and justify your answer.

sol)

- Hessian of (A) is CI
- (A) is convex as $C \geq 0$ implies Hessian of (A) is positive semi-definite (if $C \geq 0$, $x^\top CIx = Cx^\top Ix = C\|x\|_2^2 \geq 0$ for all x)

(c) [2pt] Obtain Hessian of (B) $= \log(1+\exp(w^\top b))$ in w. Check if (B) is convex, and justify your answer.

sol)

(d) Hessian of (B) is $\frac{\exp(w^\top b)}{(1+\exp(w^\top b))}bb^\top$

(e) (B) is convex as Hessian of (B) is always positive semi-definite ($\frac{\exp(w^\top b)}{(1+\exp(w^\top b))} \geq 0$ and $x^\top bb^\top x = (xb^\top)^2 \geq 0$ for all x)

[1] https://en.wikipedia.org/wiki/Hessian_matrix
[2] https://en.wikipedia.org/wiki/Definite_symmetric_matrix

(f) [2pt] Check if the cost function obtained in Problem 4a is convex, and justify your answer.

sol) It is convex as sum of convex functions is convex.

(g) [2pt] Introducing auxiliary variables $\xi_x = w^\top x$ and $\xi_z = w^\top G_\theta(z)$, consider the following optimization (for the discriminator):

$$\begin{aligned} \min_w \quad & \sum_{x\in\mathcal{D}} \log(1+\exp\xi_x) + \sum_{z\in\mathcal{Z}} \log(1+\exp(\xi_z)) - \sum_{z\in\mathcal{Z}} w^\top G_\theta(z) + \frac{C}{2}\|w\|_2^2 && \text{(3a)} \\ \text{s.t.} \quad & \xi_x = w^\top x \quad \forall x \in \mathcal{D} && \text{(3b)} \\ & \xi_z = w^\top G_\theta(z) \quad \forall z \in \mathcal{D} && \text{(3c)} \end{aligned}$$

Write the Lagrangian for this optimization, where λ_x and λ_z are Lagrange multipliers corresponding to (3b) and (3c), resp.

sol)

$$\sum_{x\in\mathcal{D}} (\lambda_x\xi_x + \log(1+\exp\xi_x)) + \sum_{z\in\mathcal{Z}} (\lambda_z\xi_z + \log(1+\exp(\xi_z)))$$
$$- w^\top\left(\sum_{x\in\mathcal{D}} \lambda_x x + \sum_{z\in\mathcal{Z}} (1+\lambda_z)G_\theta(z)\right) + \frac{C}{2}\|w\|_2^2$$

(h) [2pt] Obtain the value of

$$\min_w \frac{C}{2}\|w\|_2^2 - w^\top b$$

in terms of b and $C \geq 0$.

sol) Taking the derivative and setting it to 0, we get the optimal $w = b/C$ and thus the minimal value is

$$-\frac{1}{2C}\|b\|_2^2 .$$

(i) [2pt] Obtain the value of

$$\min_\xi \quad \lambda\xi + \log(1+\exp\xi)$$

in terms of λ assuming $-1 \leq \lambda \leq 0$.

sol) Taking the derivative and setting it to 0, we get the optimal $\xi = \log\left(\frac{-\lambda}{1+\lambda}\right)$, and the minimal value is $\lambda\log(-\lambda) - (1+\lambda)\log(1+\lambda)$.

(j) [4pt] Combining Problems 4g, 4h, and 4i and using $H(a) = a\log(-a) - (1 + a)\log(1 + a)$, obtain dual function $g(\lambda)$ for (3). For training the discriminator, we can replace the original minimization over w described in (2) with the dual maximization over valid values of λ. Using this, write down an alternative of GAN training in (2), in which we have a max-max problem instead of the max-min problem. Note that such an alternative training in max-max form can help to bypass challenges from fining a saddle-point, i.e., solving the max-min problem.

sol)

- Dual function:

$$g(\lambda) = -\frac{1}{2C}\left\|\sum_{x\in\mathcal{D}}\lambda_x x + \sum_{z\in\mathcal{Z}}(1+\lambda_z)G_\theta(z)\right\|_2^2 + \sum_{x\in\mathcal{D}}H(\lambda_x) + \sum_{z\in\mathcal{Z}}H(\lambda_z)$$

- Alternative GAN training:

$$\max_{\theta} \max_{-1\leq\lambda_x\leq 0, -1\leq\lambda_z\leq 0} g(\lambda)$$

(k) [2pt] Complete `GAN.py`, which is an implementation of the alternative training of GAN obtained in Problem 4j with the $\log D$ trick in the lecture. (Hint: use `target1` and `target2`)

sol) See `sol_GAN.py`