DD2434/FDD3434 Machine Learning, Advanced Course
Assignment 3, 2021 (V2.0)

Jens Lagergren and Aristides Gionis

Deadline, see Canvas

Read this before starting

There are some commonalities between the problems and they cover different aspects of the course and very in difficulty, consequently, it may be useful to read all of them before starting. Also think about the formulation and try to visualize the model. You are allowed to discuss the formulations, but have to make a note of the people you have discussed with. You will present the assignment by a written report, submitted before the deadline using Canvas. You must solve the assignment individually and it will automatically be checked for similarities to other students' solutions as well as documents on the web in general. Although you are allowed to discuss the problem formulations with others, you are not allowed to discuss solutions, and any discussions concerning the problem formulations must be described in the solutions you hand in. From the report it should be clear what you have done and you need to support your claims with results. You are supposed to write down the answers to the specific questions detailed for each task. This report should clearly show how you have drawn your conclusions and explain your derivations.

Being able to communicate results and conclusions is a key aspect of scientific as well as corporate activities. It is up to you as a author to make sure that the report clearly shows what you have done. Based on this, and only this, we will decide if you pass the task. No detective work should be required on our side. In particular, neat and tidy reports please!

The grading of the assignment will be as follows,

C Two correctly solved problems of 3.1–3.6

B Four correctly solved problems of 3.1–3.6

A Five correctly solved problems of 3.1–3.6

These grades are valid for assignments submitted before the deadline, late assignments can at most receive the grade E, which makes it meaningless to hand in late solutions for this assignment.

Good Luck!

3.1 Success probability in the Johnson-Lindenstrauss lemma

In the proof of Johnson-Lindenstrauss lemma we first bounded the probability that a single projection maintains all pairwise distances with distortion that is between $(1-\epsilon)$ and $(1+\epsilon)$. In particular, we showed that the probability of achieving such a distortion for all pairs of points is at least $1/n$. Assume now, that we want to boost the probability of success to be at least 95%.

Question 3.1.1: *Show that $\mathcal{O}(n)$ independent trials are sufficient for the probability of success to be at least 95%.*

An independent trial here refers to generating a new projection of the data points with a newly-generated projection matrix.

3.2 Node similarity for representation learning

Let $G=(V,E)$ be an undirected and connected graph and let $\mathbf{A}$ be the adjacency matrix of G, that is, $\mathbf{A}_{ij}=1$ if $(i,j)\in E$ and $\mathbf{A}_{ij}=0$ otherwise.
Let $\mathbf{D}$ be a diagonal matrix with $\mathbf{D}_{ii}=\sum_j \mathbf{A}_{ij}$, and let $\mathbf{P}=\mathbf{D}^{-1}\mathbf{A}$.
In graph representation learning, our goal is to learn vector representations (embeddings) for the nodes of the graph. The main idea is to define an appropriate similarity measure between the graph nodes, and then learn vector representations for the graph nodes, so that the similarity between pairs of learned vectors approximates the similarity between the corresponding graph nodes.
Assume now that for a similarity measure between graph nodes, we define

$$\mathbf{S}_{ij}=\sum_{k=1}^{\infty}\alpha^k\,[\mathbf{P}^k]_{ij},$$

for each pair of nodes $i,j\in V$, and for some real $0<\alpha<1$. Here, by $[\mathbf{P}^k]_{ij}$ we refer to the (i,j) entry of the matrix $\mathbf{P}^k$.

Question 3.2.2: *Explain the intuition for the definition of the similarity measure* $\mathbf{S}$.

Question 3.2.3: *Show that* $\mathbf{S}$ *can be computed efficiently using matrix addition, and a single matrix inversion operation, while avoiding computing an infinite series.*

3.3 Complicated likelihood for leaky units on a tree

Consider the following model. A binary tree T has random variables associated with its vertices. A vertex u has an observable variable X_u and a latent class variable Z_u. Each class $c\in[C]$ has a Normal distribution $\mathcal{N}(\mu_c,\sigma^2)$. If the three neighbors of u are v_1, v_2, and v_3, then

$$p(X_u|Z_u=c,Z_{v_1}=c_1,Z_{v_2}=c_2,Z_{v_3}=c_3)\sim\mathcal{N}(X_u\mid(1-\alpha)\mu_c+\sum_{i=1}^{3}\frac{\alpha}{3}\mu_{c_i},\ \sigma^2)$$

The class variables are i.i.d., each follows the categorical distribution π.

Question 3.3.4: *Provide a linear time algorithm that computes $p(X|T,M,\sigma,\alpha,\pi)$ when given a tree T (with vertices $V(T)$), observable variables for its vertices $X=\{X_v:v\in V(T)\}$, and parameters $M=\{\mu_c:c\in[C]\},\sigma,\alpha$.*

Note: For root vertex you can assign the weights as $(1-\alpha)$ & $(\alpha/2)$ and for a leaf vertex $(1-\alpha)$ & α.

3.4 Super Epicentra - Variational Inference

As in Task 2.4 from Assignment 2, we have seismographic from an area with frequent earthquakes emanating from K super epicentra. In fact, the core of the present model is the model described in Task 2.4 from Assignment 2, but now the parameters also have conjugate prior distributions. As shown in Figure 1, the present model has the following prior distributions.

1. π has a $\mathrm{Dir}(\alpha)$ prior.
2. $\tau_{k,i}$ has a $\mathrm{Ga}(\alpha', \beta')$ prior.
3. $\mu_{k,i}$ has a $\mathcal{N}(\mu, (C\tau_{k,i})^{-1})$ prior.
4. λ_k has a $\mathrm{Ga}(\alpha_0, \beta_0)$ prior.

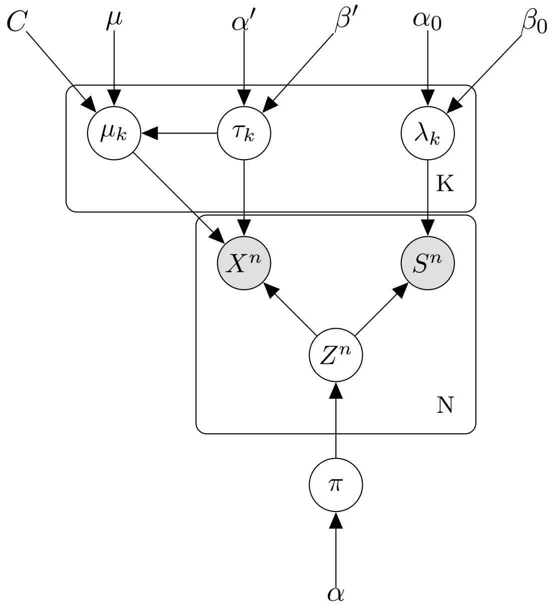

Figure 1: The K super epicentra model with priors.

Question 3.4.5: *Derive a VI algorithm that estimates the posterior distribution for this model.*

3.5 The Casino Model and Sampling Tables Given Dice Sums

Consider the following generative model. There are $2K$ tables in a casino, $t_1, ..., t_K, t'_1, ..., t'_K$ of which each is equipped with a single dice (which may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$) and N players $P_1, ..., P_N$ of which each is equipped with a single dice (which also may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$). Each player P_i visits K tables. In the k:th step, if the previous table visited was t_{k-1}, the player visits t_k with probability $1/4$ and t'_k with probability $3/4$, and if the previous table visited was t'_{k-1}, the player visits t'_k with probability $1/4$ and t_k with probability $3/4$. So, in each step the probability of staying among the primed or unprimed tables is $1/4$. At table k player i throws her own dice as well as the table's dice. We then observe the sum S^i_k of the two dice, while the outcome of the table's dice X_k and the player's dice Z_k are hidden variables. So for player i, we observe $S^i = S^i_1, ..., S^i_K$, and the overall observation for N players is $S^1, ..., S^N$.

Question 3.5.6: *Provide a drawing of the Casino model as a graphical model. It should have a variable indicating the table visited in the k:th step, variables for all the dice outcomes, variables for the sums, and plate notation should be used to clarify that N players are involved.*

Question 3.5.7: *Implement the casino model.*

Question 3.5.8: *Provide data generated using at least three different sets of categorical dice distributions – what does it look like for all unbiased dice, i.e., uniform distributions, for example, or if some are biased in the same way, or if some are unbiased and there are two different groups of biased dice*

You will now design an algorithm that does inference on the casino model that you designed.

Question 3.5.9: *Describe an algorithm that, given (1) the parameters Θ of the full casino model of Task 2.2 (so, Θ is all the categorical distributions corresponding to all the dice), (2) a sequence of tables $r_1 \ldots, r_K$ (that is, r_i is t_i or t'_i) , and (3) an observation of dice sums $s_1, \ldots, s_K$, outputs $p(r_1, ..., r_K | s_1, \ldots, s_K, \Theta)$.*

Notice, in the dynamic programming algorithm for the above problem you have to keep track of the last table visited.

Question 3.5.10: *You should also show how to sample $r_1, \ldots, r_K$ from $p(R_1, ..., R_K | s_1, \ldots, s_K, \Theta)$ as well as implement and show test runs of this algorithm. In order to design this algorithm show first how to sample r_K from*

$$p(R_K | s_1, \ldots, s_K, \Theta) = p(R_K, s_1, \ldots, s_K | \Theta) / p(s_1, \ldots, s_K | \Theta)$$

and then r_{K-1} from

$$p(R_{K-1} | r_K, s_1, \ldots, s_K, \Theta) = p(R_{K-1}, r_K, s_1, \ldots, s_K | \Theta) / p(r_K, s_1, \ldots, s_K | \Theta).$$

3.6 The Casino Model - Expectation-Maximization

Consider the following simplification of the casino model from Problem 3.5 . There are K tables in the casino $t_1, ..., t_K$ of which each is equipped with a single dice (which may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$) and N players $P_1, ..., P_N$ of which each is equipped with a single dice (which also may be biased, i.e., any categorical distribution on $\{1, ..., 6\}$). Let Θ be the parameters of all these categorical distributions.

Each player P_i visits the K tables in the order $1, ..., K$. At table k the player i throws her own dice as well as the table's dice. We then observe the sum S_k^i of the dice, while the outcome of the table's dice X_k and the player's dice Z_k are hidden variables. So for player i, we observe $s^i = s_1^i, ..., s_K^i$, and the overall observation for N players is $s^1, ..., s^N$.

Design and describe an EM algorithm for this model. That is, an EM algorithm that given $s^1, ..., s^N$ finds locally optimal parameters for the categorical distributions (i.e., the dice), that is, the Θ maximising $P(s_1^i, ..., s_K^i|\Theta)$.

Question 3.6.11: *Present the algorithm written down in a formal manner (using both text and mathematical notation, but not pseudo code).*

Question 3.6.12: *Implement it and test the implementation with data generated in Task 3.5 , and provide graphs or tables of the results of testing it with the data.*