Assignment 5

Foundations of Machine Learning
IIT-Hyderabad
Aug-Dec 2021

Max Marks: 30
Due: 26th Nov 2021 11:59 pm

This homework is intended to cover programming exercises in the following topics:

- Clustering, Expectation Maximization, PCA, t-SNE

Instructions

- Please upload your submission on Google Classroom by the deadline mentioned above. Your submission should comprise of a single file (PDF/ZIP), named `<Your_Roll_No>_Assign5`, with all your solutions.
- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 7 grace days for late submission of assignments (of which atmost 4 can be used for a given submission). Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the FoML Marks and Grace Days document.
- Please use PYTHON for the programming questions.
- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

Questions: Theory

1. **Hierarchical Clustering *(4 marks)*:** Given below is the distance matrix for 6 data points

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0					
x_2	0.12	0				
x_3	0.51	0.25	0			
x_4	0.84	0.16	0.14	0		
x_5	0.28	0.77	0.70	0.45	0	
x_6	0.34	0.61	0.93	0.20	0.67	0

(a) Draw a dendrogram for the final result of hierarchical clustering with single link. [*1 mark*]

(b) Draw a dendrogram for the final result of hierarchical clustering with complete link. [*1 mark*]

(c) Change two values from the matrix so that the answer to the last two questions is same. [*2 marks*]

2. **PCA:** ***(7 marks)***

(a) ***(2 marks)*** Let $\mathbf{X}' = [\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_p}]$ have covariance matrix $\mathbf{\Sigma}$, with eigenvalue-eigen vector pairs $(\lambda_1, e_1), (\lambda_2, e_2), ...(\lambda_p, e_p)$, where $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$.
Let $\mathbf{Y_1} = \mathbf{e_1'X}, \mathbf{Y_2} = \mathbf{e_2'X}, ..., \mathbf{Y_p} = \mathbf{e_p'X}$ be the principal components. Prove that:

$$\sigma_{11} + \sigma_{22} + ... + \sigma_{pp} = \sum_{i=1}^{p} Var(\mathbf{X_i}) = \lambda_1 + \lambda_2 + ... + \lambda_p = \sum_{i=1}^{p} Var(\mathbf{Y_i})$$

(b) ***(5 marks)*** Suppose the random variables $\mathbf{X_1}, \mathbf{X_2}, \mathbf{X_3}$ have covariance matrix

$$\Sigma = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

It may be verified that the eigenvalue-eigenvector pairs are

$$\lambda_1 = 5.83, \mathbf{e_1'} = [0.383, -0.924, 0]$$
$$\lambda_2 = 2.00, \mathbf{e_2'} = [0, 0, 1]$$
$$\lambda_3 = 0.17, \mathbf{e_3'} = [0.924, 0.383, 0]$$

i. Find out the principal components $\mathbf{Y_1}, \mathbf{Y_2}, \mathbf{Y_3}$. ***(1 mark)***

ii. Do you think $\mathbf{X_3}$ is a principal component? If so, why? ***(0.5 mark)***

iii. Demonstrate $Var(\mathbf{Y_i}) = \lambda_i, i = 1, 2, 3$, and $Cov(\mathbf{Y_i}, \mathbf{Y_k}) = 0, i \neq k$. ***(2 marks)***

iv. Do you think any of the principal components could be ignored/eliminated? Give reasons. ***(1.5 marks)***

3. **EM application:** ***(9 marks)*** Consider the following problem. There are P papers submitted to a machine learning conference. Each of R reviewers reads each paper, and gives it a score indicating how good he/she thought that paper was. We let $x^{(pr)}$ denote the score that reviewer r gave to paper p. A high score means the reviewer liked the paper, and represents a recommendation from that reviewer that it be accepted for the conference. A low score means the reviewer did not like the paper.
We imagine that each paper has some "intrinsic" true value that we denote by μ_p, where a large value means it's a good paper. Each reviewer is trying to estimate, based on reading the paper, what μ_p is; the score reported $x^{(pr)}$ is then reviewer r's guess of μ_p.
However, some reviewers are just generally inclined to think all papers are good and tend to give all papers high scores; other reviewers may be particularly nasty and tend to give low scores to everything. (Similarly, different reviewers may have different amounts of variance in the way they review papers, making some reviewers more consistent/reliable than others.)

We let ν_r denote the "bias" of reviewer r. A reviewer with bias ν_r is one whose scores generally tend to be ν_r higher than they should be.
All sorts of different random factors influence the reviewing process, and hence we will use a model that incorporates several sources of noise. Specifically, we assume that reviewers' scores are generated by a random process given as follows:

$$y^{(pr)} \sim \mathcal{N}(\mu_p, \sigma_p^2) \quad (1)$$

$$z^{(pr)} \sim \mathcal{N}(\nu_p, \tau_r^2) \quad (2)$$

$$x^{(pr)}|y^{(pr)}, z^{(pr)} \sim \mathcal{N}(y^{(pr)} + z^{(pr)}, \sigma^2) \quad (3)$$

The variables $y^{(pr)}$ and $z^{(pr)}$ are independent; the variables (x, y, z) for different paper reviewer pairs are also jointly independent. Also, we only ever observe the $x^{(pr)}$'s; thus, the $y^{(pr)}$'s and $z^{(pr)}$'s are all latent random variables.

We would like to estimate the parameters μ_p, σ_p^2, ν_r, τ_r^2 If we obtain good estimates of the papers' "intrinsic values" μ_p these can then be used to make acceptance/rejection decisions for the conference.

We will estimate the parameters by maximizing the marginal likelihood of the data $\{x^{(pr)}; p = 1, ..., P, r = 1, ..., R\}$ This problem has latent variables $y^{(pr)}$ and $z^{(pr)}$, and the maximum likelihood problem cannot be solved in closed form. So, we will use EM. Your task is to derive the EM update equations. Your final E and M step updates should consist only of addition/subtraction/multiplication/division/log/exp/sqrt of scalars; and addition/subtraction/multiplication/inverse/determinant of matrices. For simplicity, you need to treat only $\{\mu_p, \sigma_p^2; p = 1, ..., P\}$ and $\{\nu_r, \tau_r^2; r = 1, ..., R\}$ as parameters. I.e. treat σ^2 (the conditional variance of $x^{(pr)}$ given $y^{(pr)}$ and $z^{(pr)}$) as a fixed, known constant.

(a) we will derive the E-step:

i. The joint distribution $p(y^{(pr)}, z^{(pr)}, x^{(pr)})$ has the form of a multivariate Gaussian density. Find its associated mean vector and co-variance matrix in terms of the parameters μ_p, σ_p^2, ν_r, τ_r^2 and σ^2.

ii. Derive an expression for $Q_{pr}(y^{(pr)}, z^{(pr)}) = p(y^{(pr)}, z^{(pr)}|x^{(pr)})$ (E-step), using the rules for conditioning on subsets of jointly Gaussian random variables

(b) Derive the M-step updates to the parameters { μ_p, σ_p^2, ν_r, τ_r^2 }. [Hint: It may help to express the lower bound on the likelihood in terms of an expectation with respect to $(y^{(pr)}, z^{(pr)})$ drawn from a distribution with density $Q_{pr}(y^{(pr)}, z^{(pr)})$]

Questions: Programming

1. **Clustering *(7 marks)*:** DBSCAN, as we discussed in class, is a density-based clustering algorithm. In this problem, you need to implement your own DBSCAN algorithm. You can read more about it from paper that proposed this method [link].

(a) Use the Kmeans clustering algorithm from sklearn and find the number of clusters in dataset1 shared with you. Plot the data points with different colors for different clusters. [*1 mark*]

(b) Implement your own DBSCAN algorithm on the same dataset and plot the data points. [*3 marks*]

(c) What differences do you see between the DBSCAN and k-means methods, and why? [*1 mark*]

(d) Consider the dataset2 (also shared with you) with three clusters. Use (a) and (b) for dataset2, and compare the performance. List your observations clearly, and make conclusions on pros and cons of DBSCAN and k-means. [*2 marks*]

2. **t-SNE:** ***(3 marks)*** In this question, use the handwritten digits dataset which has images of size 8×8 (64 dimensional). Use sklearn's t-SNE algorithm and reduce this 64-dimensional data to 2 dimensions (2-D).

   - For dataset, you can use 'load_digits' from sklearn.datasets
   - You can use numpy, scipy for any mathematical operations and seaborn for scatterplot
   - Implement the algorithm taking perplexity = 30 and number of iterations = 1000
   - It is recommended to take degrees of freedom as (dimensions of reduced space - 1) which is 2-1 = 1 here.
   - Visualize the reduced 2-D data as a scatter plot

   (a) Fix perplexity to 30, repeat the algorithm with number of iterations = 100 and 2000. Observe the scatter plots. Write your observation on how scatter plots varies with no. of iterations (100, 1000 and 2000) and give reasons. (For example, if you say that experiments with no. of iterations = 1000 and 2000 produces the same results, give reasons on why you think that happens and so on)

   (b) It is observed that, for some datasets, different runs of t-SNE algorithm with the same hyperparameters produce different results. Why do you think it happens?

   Deliverables:

   - Code (Preferably an ipynb file)
   - Brief report (PDF) with your answers to the above questions