COMS 4721/4771 HW4 (Spring 2024)

Due: Sunday April 21, 2024 at 11:59pm

This homework is to be done **alone**. No late homeworks are allowed. To receive credit, a typesetted copy of the homework pdf must be uploaded to Gradescope by the due date. You must show your work to receive full credit. Discussing possible approaches for solutions for homework questions is encouraged on the course discussion board and with your peers, but you must write your own individual solutions and **not** share your written work/code. You **must cite** all resources (including online material, books, articles, ai generative bots, help taken from/given to specific individuals, etc.) you used to complete your work.

It is your responsibility to protect your work, and it is a violation of academic integrity policy to post any part of these questions or your answers to public websites such as: github, bitbucket, chegg, coursehero, etc. **Violators will be reported to the dean for disciplinary action**.

1 Low-Dimensional Embeddings from Dissimilarity Data

We often encounter problems in machine learning where we do not have access to the data directly, but instead to dissimilarity ratings or comparisons between our datapoints. For example, in a series of medical trials for drug development, we do not have access to a Euclidean representation of the drugs in question, but we may have access to differential trial data which compared the performance of different drugs across a population. For another more concrete example, consider distances between cities. Given interpoint distances between n cities, we'd like to be able to reproduce the 2-dimensional global positions of the cities in question. As a third example, at a wine tasting, you may know only the relative quality or character of each wine, represented as a set of ratings, but you may want to find an embedding of the wines for clustering or visualization purposes, according to these ratings. We will explore how this can be done.

Mathematically, we are given dissimilarity ratings in an $n \times n$ matrix $D \in \mathbb{R}^{n \times n}$ where $D_{ij} = d(\alpha_i, \alpha_j)^2$ for some data $\alpha_1, ...\alpha_n$ (which we do not have access to). We'd like to find a k-dimensional Euclidean representation $x_1, ..., x_n \in \mathbb{R}^k$ such that

$$\sum_{i \neq j}^{n} \left(D_{ij} - \|x_i - x_j\|^2\right)^2$$

is minimized, i.e. the learned (squared) Euclidean distance is as close as possible to the given distance D_{ij}.

(i) First, we will show that, if the underlying data $\alpha_1, ..., \alpha_n$ is Euclidean in $\mathbb{R}^n$ (i.e. there exists a perfect embedding such that $\|\alpha_i - \alpha_j\|^2 = D_{ij}$ for all i, j), we can recover this embedding exactly from the D matrix alone. First, we would like to transform the data matrix D_{ij} into a set of inner products of the form

$$= \langle \alpha_i - \overline{\alpha}, \alpha_j - \overline{\alpha} \rangle$$

where $\overline{x}$ represents the data average. Let $H = I - \frac{1}{n}\mathbb{1}\mathbb{1}^T$. Show that $-\frac{1}{2}H^T D H$ has the desired form. This is called a Gram Matrix.

Hint: $\|\alpha_i - \alpha_j\|^2 = \langle \alpha_i, \alpha_i \rangle + \langle \alpha_j, \alpha_j \rangle - 2\langle \alpha_i, \alpha_j \rangle$. Also try expanding both sides and matching up terms.

(ii) Assume the matrix B_{ij} is in this form. Let Q be the matrix whose columns are the eigenvectors of B, and $\Lambda^{1/2}$ the diagonal matrix whose diagonal entries are roots of the corresponding eigenvalues. Show that the rows of the matrix $Q\Lambda^{1/2} \in \mathbb{R}^{n \times n}$ are a perfect (isometric) embedding of the original data into $\mathbb{R}^n$.

Hint: First prove that B is positive semi-definite. What does this imply? It turns out that the data matrix is in fact isometrically embeddable in $\mathbb{R}^n$ `if and only` if the Gram matrix is positive semi-definite.

(iii) What if we want a lower-dimensional embedding instead? Show that if we can take the top k eigenvectors Q_k and corresponding eigenvalues Λ_k of the centered matrix B, the rows of $Q_k\Lambda_k^{1/2}$ minimize the loss

$$\sum_{I\neq j}^{n}\left(D_{ij}-\|x_i-x_j\|^2\right)^2$$

over all possible k-dimensional embeddings $x_i \in \mathbb{R}^k$ (this is the same as PCA on the original matrix X).

Hint: You may wish to use SVD and the Eckart-Young theorem, or (requiring slightly more work) rewrite this equation as the PCA objective function. Rewriting the objective using the Frobenius norm as

$$\min_{\text{rank}\mathbf{Q}<k}\|D-\mathbf{Q}\|_F^2$$

think about whether applying H^T and H to the left and right sides of the normed term actually changes the minimizing value of Q? How does this change D and Q?

(iv) Download `city_distance_data.csv`. It contains an array of distances between major cities in the United States (BOS, NYC, DC, MIA, CHI, SEA, SF, LA, and DEN).

	BOS	NYC	DC	MIA	CHI	SEA	SF	LA	DEN
BOS	0	206	429	1504	963	2976	3095	2979	1949
NYC	206	0	233	1308	802	2815	2934	2786	1771
DC	429	233	0	1075	671	2684	2799	2631	1616
MIA	1504	1308	1075	0	1329	3273	3053	2687	2037
CHI	963	802	671	1329	0	2013	2142	2054	996
SEA	2976	2815	2684	3273	2013	0	808	1131	1307
SF	3095	2934	2799	3053	2142	808	0	379	1235
LA	2979	2786	2631	2687	2054	1131	379	0	1059
DEN	1949	1771	1616	2037	996	1307	1235	1059	0

Implement the above algorithm to learn an embedding of this data in 2-dimensions which approximately matches the given distances. Plot the output, and include it in your submission. Does it agree with your geographic intuition? Write a few works about why it might look the way it does.

2 Non-parametric Regression via Bayesian Modelling

Here we will study a generative modelling technique via Gaussians for non-parametric regression.

Before getting into regression we need to derive some facts about multivariate Gaussian distributions. Let $x \in \mathbb{R}^d$ be distributed normally as

$$x=\begin{bmatrix}x_1\\x_2\end{bmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\mu_1\\\mu_2\end{bmatrix},\begin{bmatrix}\Sigma_{11}&\Sigma_{12}\\\Sigma_{21}&\Sigma_{22}\end{bmatrix}\right)=\mathcal{N}(\mu,\Sigma)$$

(i) Derive the marginal distribution of x_1?

(ii) (*warning! tedious calculations*) Let $\Sigma^{-1}=\begin{bmatrix}\Sigma^{11}&\Sigma^{12}\\\Sigma^{21}&\Sigma^{22}\end{bmatrix}$. Using the facts that[1]

- $\Sigma^{11}=(\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^{\mathsf{T}})^{-1}=\Sigma_{11}^{-1}+\Sigma_{11}^{-1}\Sigma_{12}(\Sigma_{22}-\Sigma_{12}^{\mathsf{T}}\Sigma_{11}^{-1}\Sigma_{12})^{-1}\Sigma_{12}^{\mathsf{T}}\Sigma_{11}^{-1}$

[1]Feel free to prove these facts for yourself, if you are bored :).

- $\Sigma^{22} = (\Sigma_{22} - \Sigma_{12}^\mathsf{T}\Sigma_{11}^{-1}\Sigma_{12})^{-1} = \Sigma_{22}^{-1} + \Sigma_{22}^{-1}\Sigma_{12}^\mathsf{T}(\Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{12}^\mathsf{T})^{-1}\Sigma_{12}\Sigma_{22}^{-1}$
- $\Sigma^{12} = -\Sigma_{11}^{-1}\Sigma_{12}(\Sigma_{22} - \Sigma_{12}^\mathsf{T}\Sigma_{11}^{-1}\Sigma_{12})^{-1} = (\Sigma^{21})^\mathsf{T}$

Show that the joint distribution on x can be written as

$$\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\Big\{ -\frac{1}{2}\Big((x_1-\mu_1)^\mathsf{T}\Sigma_{11}^{-1}(x_1-\mu_1) + (x_2-b)^\mathsf{T}A^{-1}(x_2-b)\Big)\Big\},$$

where $b = \mu_2 + \Sigma_{12}^\mathsf{T}\Sigma_{11}^{-1}(x_1-\mu_1)$, and $A = \Sigma_{22} - \Sigma_{12}^\mathsf{T}\Sigma_{11}^{-1}\Sigma_{12}$.

(iii) Now using the fact that $|\Sigma| = |\Sigma_{11}||\Sigma_{22} - \Sigma_{12}^\mathsf{T}\Sigma_{11}^{-1}\Sigma_{12}|$, show that the joint on x can be decomposed as product

$$\mathcal{N}(x_1; \mu_1, \Sigma_{11})\mathcal{N}(x_2; b, A),$$

where b and A are as defined in previous part.

(iv) What is the conditional distribution of x_2 given x_1?

Now we are ready to do some regression via generative modelling. Like in any generative model, we first need a prior over our objects of interest (in this case, the regression functions), then given some data (also known as evidence), we shall compute the posterior (in this case, those regression functions that agree with the given data).

A prior over the regression functions. A function $f : \mathbb{R} \to \mathbb{R}$ can be thought of as an infinite length vector. Suppose we want to know the value of this function at positions $x_1, \ldots, x_n \in \mathbb{R}$, we can write it down the result as a vector $\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}$. A simple way to *generate* random functions then is to simply model it as the Gaussian distribution, specifically $\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\big(\mu_n, \Sigma_{n\times n}\big)$.

Throughout our discussion below, we will use 500 equal spaced points between the range of -10 and 10 as our locations $x_1, \ldots, x_n$, that is $x_1 = -10, x_2 = -9.959, \ldots, x_{500} = 10$.

(v) For $\mu_n = \vec{0}$ and $\Sigma_{n\times n} = I$, draw 4 random functions and show their plots[2]. What can you say about the smoothness of these functions? What happens if Σ is set to all ones matrix? Play with various values of μ and Σ, what effect does it have on the distribution of the random functions? Explain why these effects are occurring.

(vi) Usually $\mu_n = \vec{0}$ and $\Sigma_{n\times n} = K$, where $K_{ij} = k(x_i, x_j)$ for some kernel function k. A popular choice is $k : (x_i, x_j) \mapsto \exp\{-(x_i - x_j)^2/h\}$, for some fixed parameter h. Draw 4 random functions from this setting of μ and Σ ($h = 5$). and show their plots. What can you say about the smoothness of these functions?

(vii) If one is interested in random periodic functions, qualitatively explain what setting of μ and Σ would be appropriate? Pick a μ and Σ which can generate periodic functions of periodicity 3 units. Draw 4 random functions from that setting and plot them to verify.

The posterior over the regression functions. Of course, in the problem of regression, one is not interested in drawing random functions, but instead, understanding/predicting the trend in data given some observations. Suppose we are given a training data $(\bar{x}_1, \bar{y}_1), \ldots, (\bar{x}_m, \bar{y}_m) = (\overline{\mathbf{X}}, \overline{\mathbf{Y}})$. Then using the suggested model we can model the joint distribution as

[2] use x-axis range -10 to 10, y-axis range -3 to 3 for all your plots.

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \\ f(\bar{x}_1) = \bar{y}_i \\ \vdots \\ f(\bar{x}_m) = \bar{y}_m \end{bmatrix} = \begin{bmatrix} f(\mathbf{X}) \\ f(\overline{\mathbf{X}}) = \overline{\mathbf{Y}} \end{bmatrix} \sim \mathcal{N}\big(\mu_{n+m}, \Sigma_{(n+m)\times(n+m)}\big) = \mathcal{N}\left(\begin{bmatrix}\mu_n \\ \mu_m\end{bmatrix}, \begin{bmatrix}\Sigma_{nn} & \Sigma_{nm} \\ \Sigma_{mn} & \Sigma_{mm}\end{bmatrix}\right)$$

$$= \mathcal{N}\left(\begin{bmatrix}\mu_n \\ \mu_m\end{bmatrix}, \begin{bmatrix}K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\overline{\mathbf{X}}) \\ K(\overline{\mathbf{X}},\mathbf{X}) & K(\overline{\mathbf{X}},\overline{\mathbf{X}})\end{bmatrix}\right).$$

Let $y_1, \ldots, y_n = \mathbf{Y}$ be the regression values we are interested in knowing for a set of (test) locations $x_1, \ldots, x_n = \mathbf{X}$, then we can write the above joint model more compactly as

$$\begin{bmatrix}\mathbf{Y} \\ \overline{\mathbf{Y}}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mu_n \\ \mu_m\end{bmatrix}, \begin{bmatrix}K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\overline{\mathbf{X}}) \\ K(\overline{\mathbf{X}},\mathbf{X}) & K(\overline{\mathbf{X}},\overline{\mathbf{X}})\end{bmatrix}\right).$$

Hence, given the training data $(\overline{\mathbf{X}}, \overline{\mathbf{Y}})$ we are interested in knowing the posterior $\mathbf{Y}\big|\overline{\mathbf{Y}}$

(viii) Using the result from part (iv), what is the posterior $\mathbf{Y}\big|\overline{\mathbf{Y}}$?

(ix) For training data $\{(-6,3),(0,-2),(7,2)\}$ and K induced by kernel function $k : (x_i, x_j) \mapsto \exp\{-(x_i - x_j)^2/5\}$, draw 4 random functions from the posterior and plot the resulting functions. Make sure to depict the three training datapoints on the same plot. What do you notice?

(x) For the training data in part (ix) and the periodic Σ used in part (vii), draw 4 random functions from the posterior and plot the resulting functions (along with the training data). What do you notice in this case?

Notice that this Bayesian modelling technique provides a (posterior) *distribution* over regression values for the test locations. One can use the mean value as the final prediction over the test locations.

(xi) What is the mean of the posterior $\mathbf{Y}\big|\overline{\mathbf{Y}}$?

(xii) Plot the mean "function" for parts (ix) and (x) as well.

3 Non-linear Dimensionality Reduction

Here is a simple way to accomplish non-linear dimensionality reduction:

Input: High-dimensional dataset $X = x_1, \ldots, x_n \in \mathbb{R}^D$, target dimension d
Output: $y_1, \ldots, y_n \in \mathbb{R}^d$ as the low-dimensional mapping of the given dataset

- Construct a k-nearest neighbor graph[3] G on X
- Let π_{ij} denote the shortest path between datapoints x_i and x_j according to G.
- Select $y_1, \ldots, y_n \in \mathbb{R}^d$ according to the following minimization problem

$$\text{minimize}_{y_1,\ldots,y_n} \sum_{i,j} \big(\|y_i - y_j\| - \pi_{ij}\big)^2$$

(i) What is the derivative of the optimization function above with respect to a fixed y_i?

[3] A k-nearest neighbor graph is simply a graph where the nodes correspond to the datapoints, and edges correspond to the Euclidean distance between the corresponding datapoints. Important: For each node/datapoint, only the k closest nodes are connected, with edge weight being the Euclidean distance between the nodes.

(ii) Is the optimization above convex with respect a fixed y_i? Why or why not.

(iii) Write a program in your preferred language to find a low-dimension embedding of any given input dataset. You must submit your code to receive full credit.

(iv) For the two datasets provided, compute your two dimensional embedding. Plot the original 3D data, its 2D PCA projection and the results obtained from your 2D embedding.

Analyze the quality of results you obtain. Under what circumstances would

- this non-linear embedding fail/succeed?
- PCA will perform better/worse than this non-linear embedding?