CIS 4190/5190: Applied Machine Learning — Fall 2022

Homework 5

Handed Out: November 2 — *Due: November 16, 8:00 p.m.*

- You are encouraged to format your solutions using LaTeX. You'll find some pointers to resources for learning LaTeXamong the Canvas primers. Handwritten solutions are permitted, but remember that you bear the risk that we may not be able to read your work and grade it properly — do not count on providing post hoc explanations for illegible work. You will submit your solution manuscript for written HW5 as a single PDF file.

- The homework is **due at 8:00 PM** on the due date. We will be using Gradescope for collecting the homework assignments. Please submit your solution manuscript as a PDF file via Gradescope. Post on Piazza and contact the TAs if you are having technical difficulties in submitting the assignment.

1 Multiple Choice & Written Questions

Note: You do not need to show work for multiple choice questions. If formatting your answer in LaTeX, use our LaTeX template `hw5_template.tex` (This is a read-only link. You'll need to make a copy before you can edit. Make sure you make only private copies.).

1. [Text Generation/Language Modeling] (10 pts) Text generation is a popular application and area of research in NLP. In this problem, we will look at a specific yet common scenario of text generation, where you want to generate a sentence by sampling words from an autoregressive language model (such as GPT). Given the first n prompt words $\{w_1, w_2, ..., w_n\}$ from left-to-right order in a sentence, an autoregressive language model outputs the probability distribution of the next word conditioned on the prompt words: $P(w_{n+1} \mid w_1, w_2, ..., w_n)$. A complete sentence can be generated by iteratively sampling words from the next word probability distributions until an end-of-sentence indicator (such as period ".") is reached. But how should we sample the words from $P(w_{n+1} \mid w_1, w_2, ..., w_n)$?

   In this question, we will compare two different sampling strategies and learn the intuition behind them with a toy example. Suppose you are interested in generating a sentence that starts with the word "Bob". You are given an autoregressive language model with only 5 words in vocabulary – {Bob, loves, hates, cherry, cookie}. You tried the following three prompts, and here are the three conditional probability distributions of the next word you get. For all subquestions, assume that you only want to generate the next two words after "Bob".

w_2	$P(w_2 \mid w_1 = \text{Bob})$
loves	0.6
hates	0.3
cookie	0.05
cherry	0.03
Bob	0.02

w_3	$P(w_3 \mid w_1 = \text{Bob}, w_2 = \text{loves})$
cookie	0.35
Bob	0.3
cherry	0.25
hates	0.08
loves	0.02

w_3	$P(w_3 \mid w_1 = \text{Bob}, w_2 = \text{hates})$
cookie	0.8
cherry	0.1
Bob	0.08
loves	0.01
hates	0.01

(a) (2 pts) A natural goal of text generation is to generate the most likely sentence out of your vocabulary. In other words, you want to sample the sentence w_1, w_2, w_3 that maximizes the joint probability $P(w_2, w_3 \mid w_1 = \text{Bob})$. Optimizing the joint probability can be NP hard in general, so a commonly used approximation is to use the following estimate:

$$\ln P(w_2, w_3 \mid w_1 = \text{Bob}) \approx \ln P(w_2 \mid w_1 = \text{Bob}) + \ln P(w_3 \mid w_1 = \text{Bob}, w_2). \quad (1)$$

Use this formula to compute estimates of the log-likelihood for the following two sentences: "Bob loves cookie" and "Bob hates cookie".

(b) (2 pts) One heuristic for computing w_2, w_3 that optimizes Eq. 1 is greedy sampling—i.e., iteratively sample the next word with highest conditional probability. In our setting, this strategy is to sample w_2^* that maximizes $P(w_2 \mid w_1 = \text{Bob})$, and then sample w_3^* that maximizes $P(w_3 \mid w_1 = \text{Bob}, w_2 = w_2^*)$. What is the sentence generated by this strategy (i.e. "Bob" plus the next two words)?

(c) (2 pts) Based on your answer to the previous question, does the greedy sampling strategy always give you the sentence with the highest estimated log-likelihood? Explain your answer.

(d) (4 pts) Consider an alternative sampling strategy called beam search. Instead of always taking the highest likelihood word, this strategy takes the top k words; for this question, we let $k = 2$. In our case, for w_2, this strategy would give us a beam with two hypotheses: "Bob loves" and "Bob hates".

For w_3, we sample the top two words for the two beam hypotheses respectively, which gives us the following four hypotheses:

- "Bob loves cookie"
- "Bob loves Bob"
- "Bob hates cookie"
- "Bob hates cherry"

The next step would be estimating the log-likelihood of the four hypotheses; then, we keep the two hypotheses with the highest estimated log-likelihoods. (In general, we would iteratively generate the next word using this same strategy, but for our example, we are done since we are only looking for two additional words.) Which two hypotheses among the above four should we keep when using this strategy? In other words, which two have the top two highest estimated log-likelihood among the above four? Show your work.

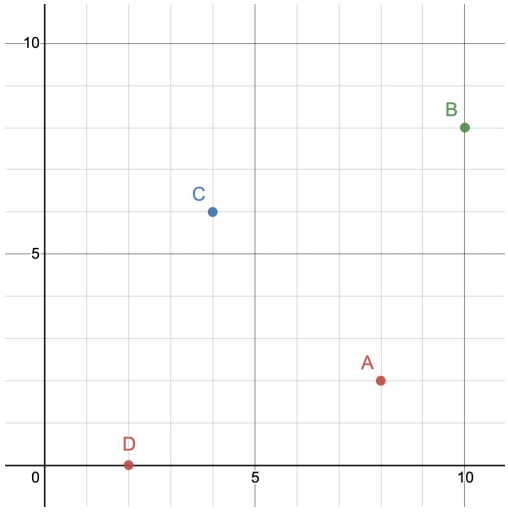
2. [PCA] (14 pts) Consider the following set of four 2D points:

$$X = \begin{bmatrix} 8 & 2 \\ 4 & 6 \\ 10 & 8 \\ 2 & 0 \end{bmatrix}.$$

Our goal is to compress it into four 1D points using PCA.

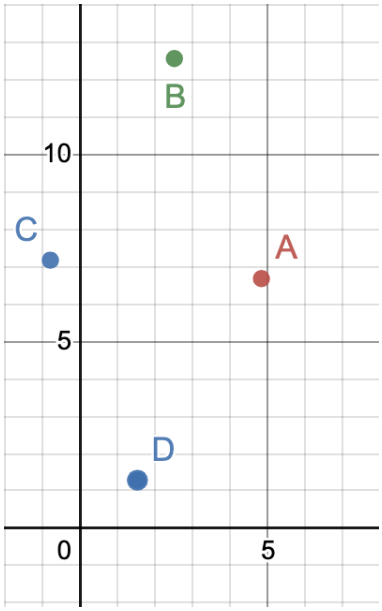
(a) (6pts) Find the unit-vector principal components of X. Given that the goal is to compress to four 1D points, which principal component should we choose, and why?

(b) (4pts) Suppose we plot the points in X as in the following figure:



Building on this plot, sketch the direction of the principal component as well as the projection of the four 2D points onto this principal component (you should plot the principal component through the center of the data). In addition, label each of the projected points with the value of its projection onto the principal component (i.e., the signed distance from the center of the data to the projected point).

(c) (4pts) Suppose that all the points in X are rotated by an angle of 40° anti-clockwise around the origin, as shown in the following figure:



Furthermore, suppose that PCA is applied to the rotated points. Sketch the new direction of the principal component, project the four 2D coordinates on this principal component, and label the values of the new principal coordinates.

3. [K-Means] (5pts) Work through the K-Means clustering algorithm for a dataset with 4 samples, with $K = 2$, and using the L_2 distance. The samples in the dataset are: $A = (1, 5)$, $B = (5, 5)$, $C = (9, 8)$, and $D = (12, 5)$. The initial centroids are chosen as: $(9, 5)$ for cluster 1 and $(11, 5)$ for cluster 2. Recall that in each iteration of K-Means, two things happen: first, cluster assignments are updated, and second, cluster centroids are updated. Work through two such iterations. Report results for each iteration as:

   - cluster 1 members: A, B, etc.
   - cluster 1 centroid: (x, y)
   - cluster 2 members: A, B, etc.
   - cluster 2 centroid: (x, y)

2 Python Programming Questions

A Google Colab notebook is linked in the "HW5 Coding" assignment on Canvas. This will tell you everything you need to do, and provide starter code. Remember to include the plots and answer the questions in your submission.