CS 274A Homework 1

Probabilistic Learning: Theory and Algorithms, CS 274A, Winter 2024

Due: 12 noon Wednesday January 17th, submit via Gradescope

Instructions and Guidelines for Homeworks

- Please answer all of the questions and submit your solutions to Gradescope (either hand-written or typed are fine as long as the writing is legible).
- All problems are worth equal points (10 points) unless otherwise stated. All homeworks will get equal weight in computation of the final grade for the class (with lowest-scoring homework being dropped).
- The homeworks are intended to help you better understand the concepts we discuss in class. It is important that you solve the problems yourself to help you learn and reinforce the material from class. If you don't do the homeworks you will likely have difficulty in the exams later in the quarter.
- In problems that ask you to derive or prove a result you should submit a complete mathematical proof (i.e., each line must follow logically from the preceding one, without "hand-waving"). Be as clear as possible in explaining your notation and in stating your reasoning as you go from line to line.
- If you can't solve a problem, you can discuss the high-level concepts *verbally* with another student (e.g., what concepts from the lectures or notes or text are relevant to a problem). However, you should not discuss any of the details of a solution with another student. In particular, do not look at (or show to any other student) *any written material* directly related to the homeworks, including other students' solutions or drafts of solutions, solutions from previous versions of this class, etc. The work you hand in should be your own original work.
- If you need to you can look up standard results/definition/identities from textbooks, class notes, textbooks, other reference material (e.g., from the Web). If you base any part of your solution on material that we did not discuss in class, or is not in the class notes, or is not a standard known result, then you may want to rovide a reference in terms of where the result is from, e.g., "based on material in Section 2.2 in ....." or a URL (e.g., Wikipedia).

Recommended Reading for Homework 1

- Note Sets 1 and 2 from the class Web page, for a review of basic concepts in probability, conditional independence, Gaussian models, etc.
- Chapter 6.1 to 6.5 in *Mathematics for Machine Learning* (MML) is recommended for additional details beyond what is covered in the Note Sets.

Problem 1: Properties of Geometric Distribution

Suppose that we repeatedly toss a coin (with no memory in the coin, so that tosses are independent) until we get the outcome of "heads." Let θ be the probability of heads on any toss. Let X be the number of such tosses until a heads occurs. This type of model can be used to describe "independent trials" and is frequently used in science and engineering to model various simple repetitive phenomena (e.g., how many times a user visits a Web site until they make a purchase, the number of consecutive days of rainfall at a given location, and so on).

In this situation X is a discrete random variable with a geometric distribution taking values $x \in \{1, 2, 3, \ldots, \}$, with a probability distribution defined as $P(x) = (1-\theta)^{x-1}\theta$. Here θ is the parameter of the geometric model (e.g., the probability of heads in coin-tossing) and $0 < \theta < 1$.

1. Prove that $\sum_{x=1}^{\infty} P(x) = 1$.
2. Derive an expression as a function of θ for the expected value of X, $\mu_x = E[X]$.
3. Derive an expression as a function of θ for the variance of X, where the variance is $\sigma_x^2 = E[(X - \mu_x)^2]$.

Problem 2: Expectations/Variance with Two Random Variables

The expected value of a real-valued random variable X, taking values x, is defined as $\mu_x = E[X] = \int p(x)\, x\, dx$ where $p(x)$ is the probability density function for X. The variance is defined as $\sigma_x^2 = var(X) = E[(X - \mu_x)^2] = \int p(x)(x - \mu_x)^2 dx$. In the questions below a and b are scalar constants (i.e., not random variables).

1. Prove that $var(X) = E[X^2] - \big(E[X]\big)^2$.

   In the next two questions let X and Y be two real-valued random variables, each one-dimensional (i.e., scalar-valued). In the equations below the expectation on the left is with respect to the joint density $p(x, y)$ and the expectations on the right are with respect to $p(x)$ and $p(y)$ respectively. Be sure to be clear in each line of your derivation and don't skip steps.

2. Prove that $E[aX + bY] = aE[X] + bE[Y]$.
3. Prove that if X and Y are independent that $var(aX + bY) = a^2 var(X) + b^2 var(Y)$.

Problem 3: Central Limit Theorem

Let $X_1, \ldots, X_n$ be a set of independent and identically distributed real-valued random variables each with the same density $p(x)$ where each X_i has mean μ and variance σ^2. (Note that the density $p(x)$ could be any probability density function, it need not be Gaussian).

1. State precisely the central limit theorem as it applies to $X_1, \ldots, X_n$ (if you don't know or remember what the central limit theorem is you will need to look it up)
2. Let $Y = \frac{1}{n}\sum_{i=1}^{n} X_i$ where each X_i has a uniform distribution $U(a, b)$ with $a = 0, b = 1$. Simulate 1000 values of Y (using any language such as Python, R, Matlab, C, etc) for each of the following values of n: $n = 10^2, 10^3, 10^4, 10^5$. You should end up with 4 sets, each with 1000 simulated values for Y. Generate histogram plots of the 4 sets (one histogram for each value of n, producing 4 histograms). Please plot all 4 histograms on a single page (makes it easier for grading). Use $30 \approx \sqrt{1000}$ bins for each histogram. No need to submit your code.
3. From the definition of the properties of a uniform random variable (e.g., its mean and variance as a function of a and b—you can look up the definitions if you don't know them) and the definition of the central limit theorem, state what the mean and variance of Y should be as a function of n.
4. Evaluate how well your empirically simulated distributions from Part 2 match what the theory predicts from Part 3, e.g., show 1 or 2 tables, with different values of n for the rows, and where (in the columns) you compare the mean and variance of the simulated data with the values that theory predicts.

Problem 4: Logistic Function

Let X be a d-dimensional real-valued (vector) random variable taking values $\underline{x}$ and let Y be a binary random variable taking values 0 or 1. Say we would like to model the conditional probability $P(Y = 1|\underline{x})$ as a function of $\underline{x}$. One well-known approach is to assume that $P(Y = 1|\underline{x})$ is defined as a logistic function (this is the basis of the logistic regression classifier in machine learning and statistics):

$$P(Y = 1|\underline{x}) \;=\; \frac{1}{1 + exp(-\alpha_0 - \underline{\alpha}^T \underline{x})}$$

where α_0 is a real-valued scalar and $\underline{\alpha}^T$ is the transpose of a $d \times 1$ vector of real-valued coefficients $\alpha_1, \ldots, \alpha_d$. In this setup Y is typically referred to as the "class": its the variable we want to predict given $\underline{x}$.

1. Prove that the definition of the logistic function above is equivalent to stating that the log-odds $\log \frac{P(Y=1|\underline{x})}{P(Y=0|\underline{x})}$ is an affine function of $\underline{x}$, i.e., can be written as $\underline{a}^T \underline{x} + b$ for some vector $\underline{a}$ and scalar b.
2. Say we know that $P(\underline{x}|Y = 1) = N(\underline{\mu}_1, \Sigma)$ and $P(\underline{x}|Y = 0) = N(\underline{\mu}_0, \Sigma)$ (i.e., we know that the densities for each class are multivariate Gaussian), where $\underline{\mu}_1$ and $\underline{\mu}_0$ are the d-dimensional means for each class and Σ is a common covariance matrix. Prove that, under these assumptions, $P(Y = 1|\underline{x})$ is in the form of a logistic function. (Hint: one way to prove this is to make use of the result from part 1 of this problem).

Problem 5: Finite Mixture Models

Finite mixture models show up in a wide variety of contexts in machine learning and statistics (we will discuss them in more detail in lectures later in the quarter). In this problem consider a real-valued random variable X taking values x (in general we can define mixtures on vectors, but here we will just consider the 1-dimensional scalar case).

The basic idea is of a mixture model is to define a density (or distribution) $p(x)$ that is a weighted mixture of K component probability density functions $p_k(x|Z = k)$, where the weights are non-negative and sum to 1, i.e.,

$$p(x) \;=\; \sum_{k=1}^{K} p_k(x|Z = k)P(Z = k)$$

where

- Z is a discrete indicator random variable taking values from 1 to K, indicating which of the K mixture components generated data point x.
- The mixture weights $\alpha_k = P(Z = k)$ are the marginal probabilities of data point x being generated by component k, with $\sum_{k=1}^{K} \alpha_k = 1, \;\; 0 \leq \alpha_k \leq 1$.
- for each value of k, $p_k(x|Z = k)$ is itself a probability density function with its own parameters θ_k. For example, if a component k is Gaussian then $\theta_k = \{\mu_k, \sigma_k^2\}$.

The full set of parameters for a mixture model consists of both (a) the K weights, and (b) the K sets of component parameters θ_k for each of the K mixture components.

(Note that the "finite" aspect of finite mixture models comes from the fact that K is finite. There are also infinite mixture models where K is unbounded, but we will not consider those here).

1. Given the definition above for a finite mixture model, prove that a finite mixture $p(x)$ is a valid probability density function, i.e., it obeys all the necessary properties needed to be a density function.
2. Derive general expressions for the (a) mean μ of $p(x)$, and (b) the variance σ^2 of $p(x)$, as a function of the component weights, means and variances $\alpha_k, \mu_k, \sigma_k^2, 1 \leq k \leq K$.
   For each of μ and σ^2, also provide an intuitive interpretation in words of your interpretation of the equations you derived for each of the mean and the variance.
3. Now assume that $K = 2$ and that both components are Gaussian densities with $\mu_1 = 0$ and $\mu_2 = 5$. Plot $p(x)$ as a function of x for each of the following cases:
   (a) $\alpha_1 = 0.5, \sigma_1 = 3, \sigma_2 = 3$
   (b) $\alpha_1 = 0.5, \sigma_1 = 2, \sigma_2 = 2$
   (c) $\alpha_1 = 0.5, \sigma_1 = 2, \sigma_2 = 1$

(d) $\alpha_1 = 0.1, \sigma_1 = 2, \sigma_2 = 2$

Let x range from -5 to 10 in your plots. Its fine to write some code to generate the plots (in fact this is preferred since generating these plots accurately by hand would be tricky to do). No need to submit your code.

Problem 6: Computation with Markov Chains

Say we have a random variable W_t that can take values from the finite set $\{a, b, c, d\}$ where t indicates which position in a sequence we are referring to, $t = 1, 2, 3, \ldots$. For example, we could have a sequence like $a\ a\ b\ a\ c\ a\ d\ \ldots$, corresponding to $W_1 = a$, $W_2 = a$, $W_3 = b$, and so on.

In this problem we will look a very simple model for sequential dependence, namely the first-order homogeneous Markov model. This model is defined by two set of parameters:(1) a K-ary initial probability distribution for $P(W_1)$, and (2) a $K \times K$ transition matrix $\mathbf{A}$, with entries $a_{ij} = P(W_{t+1} = j | W_t = i)$ with $1 \leq i, j \leq K$ and $t \geq 1$. The rows of $\mathbf{A}$ sum to 1 (this is known as a stochastic matrix). The Markov chain is said to be *homogeneous* because the transition probabilities are the same for all values of t.

For our problem, we have $K = 4$ possible values for any position in the sequence (these values are also sometimes referred to as "states" of the Markov chain). Lets assume we have the following parameters for our Markov chain:

$$\mathbf{A} = \begin{pmatrix} 0.7 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.7 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.7 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.7 \end{pmatrix}$$

i.e., each self-transition probability $a_{ii} = 0.7$ and the other transition probabilities are all 0.1. For the initial state-distribution lets assume that each state is equally likely, i.e, $P(W_1 = i) = 1/4, i = 1, \ldots, 4$.

Sidenote: there are many real-world applications of sequential models, including (perhaps most famously at the moment), large language models (LLMs). In LLMs, the conditional probabilities $P(W_{t+1}| \ldots)$ (for the next word or token) can depend on the entire history of the sequence up and including W_t, and not just on W_t. Also, the number of possible values for W_{t+1} can be on the order of 10^5 or 10^6. In this homework problem, however, we will look at a much simpler Markov chain model.

1. Given the information above, compute the numerical value of each of the following probabilities and show precisely the equations that you used in computing these numbers:
   - $P(W_2 = b)$
   - $P(W_{t+2} = d | W_t = a)$ for any value $t \geq 1$
   - $P(W_{t+3} = a | W_t = a)$ for any value $t \geq 1$
2. Using the law of total probability, show how to write a general equation for $P(W_{t+4} = j | W_t = i)$, for any pair of states i and j, where the terms in your equation only involve elements of the matrix $\mathbf{A}$,

i.e., the transition probabilities. It may be helpful for this problem to think about a Markov chain as a special case of a graphical model.

3. Consider the more general situation with K possible states and a $K \times K$ transition matrix, $K \geq 2$.

   (a) define (in pseudocode, or with equations, or clearly in words) the most efficient computation method (i.e., an algorithm) that you can think of for computing the K-ary distribution $P(W_{t+m}|W_t = i)$, for any $m \geq 2$, any $t \geq 1$, any $i \in \{1, \ldots, K\}$. (if you have answered the earlier part of this question then you in effect already have figured out how to compute these probabilities efficiently).

   (b) Define the time complexity ("big O" notation) of your efficient algorithm as a function of K and m.

Problem 7: Markov Chains of Finite Length

In our second problem, lets change the Markov model we used earlier to allow an additional state e with the special property that e indicates "end of sequence" (or "end of sentence/document" in language modeling). Technically e is known as an *absorbing state*: once the Markov chain enters this state it stops and cannot transition to any other state. As a result of adding the state e, the number of states K is now 5. Lets assume that our 5×5 transition matrix is defined as

$$\mathbf{A} = \begin{pmatrix} 0.5 & 0.1 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.5 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.5 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0.5 & 0.2 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1 \end{pmatrix} \tag{1}$$

where the last row and last column correspond to the state e. This chain is similar to the Markov chain we looked at in the earlier problem except that the self-transition probabilities are different and there is a probability of 0.2 that the sequence ends every time the chain transitions to a new state. Lets also assume that the initial state probabilities are still 1/4 for each of a, b, c, d, i.e., the chain always produces at least 1 state value before it ends.

To be clear, in terms of sequences that this model will produce, the sequence ends the first time that the state e occurs, i.e., sequences will look like $b\ a\ d\ b\ a\ e$, or $c\ c\ e$, and so on. In the problems below we will define the length L of a sequence as the length not including e, i.e., lengths 5 and 2 for the two example sequences in the previous sentence.

1. Write code to simulate $N = 1000$ sequences from the model and plot a histogram showing empirical estimates of $P(L)$ for this chain. By "empirical" we mean frequency counts in the form n_L/N where n_L is the number of simulated sequences of length L. Let the histogram have 20 bins, $L = 1, 2, \ldots, 20$ in your plot (i.e., don't plot anything above $L = 20$). You can use any programming language you wish for your simulation (e.g., Python, R, C, Matlab, etc). You do not need to submit your code.

2. It turns out that we can derive an exact equation for $P(L)$ in the form of a well-known distribution. Show how to derive an equation for $P(L)$: i.e., derive its form in a step-by-step logical manner. In your derivation assume that the transition matrix has the same general structure as in Equation 1 above, as follows:

   - For the each row $i = 1, \ldots, K-1$ in the transition matrix: self-transition probabilities all take value α, transitions to e all take value β, and transitions to states $i \neq j, 1 \leq i, j \leq K-1$ are equally likely; and α and β are constrained so that the probabilities in each row to sum to 1. In the definition of $\mathbf{A}$ for your simulations, $\alpha = 0.5$ and $\beta = 0.2$.
   - the transition probabilities in the last row are all 0 except for the self-transition probability.
   - the initial state distribution is uniform over the first $K-1$ states.

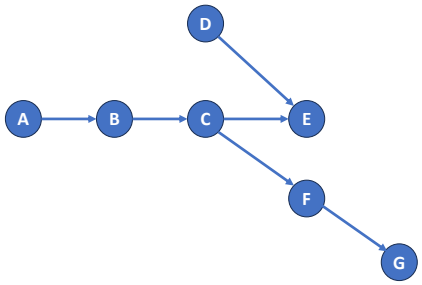
   Your equation for $P(L)$ should be a function of at least some of L, α, β, K (not necessarily all of them). Make sure your proof/derivation explains every step clearly.

3. To verify your derivation from Part 2, plot $P(L), L = 1, \ldots, 20$ overlaid on your histogram from Part 1 (making sure $P(L)$ is clearly different in color or style from the histogram itself): the empirical histogram and the theory should line up. Also calculate the mean and variance of $P(L)$, both from your simulations and from your theoretical $P(L)$, and show the results (4 numbers).

For the problem above, Part 1 should be straightforward. Part 2 will require a bit more thought: if you can't figure it out then just submit your solution for Part 1.

Problem 8: Inference in Graphical Models

Consider the directed graphical model in the figure below. All variables are discrete and all take $K \geq 2$ values.



Answer the following questions:

1. Write an equation for $P(a, b, c, d, e, f, g)$ that factorizes the joint probability in a manner that reflects the conditional independence assumptions in this graphical model.

2. List all of the different conditional and marginal probability distributions in this graphical model and define precisely how many parameters in total are required to specify the model. Take into account the fact that all distributions must sum to 1. "Parameter" here means a marginal or conditional probability in a probability table. Express your final answer in the form of a polynomial in K.

3. Consider computing the probability $P(G = g^*|A = a^*)$, where g^* and a^* are some specific values of G and A respectively. Describe (step by step, for all steps) the most efficient way to compute this conditional probability, using only the marginal and conditional probability tables that are specified in the graphical model. You can interpret "most efficient" to mean a method that requires the least number of summations as a function of K, e.g., $\sum_{x,y} P(x, y, z^*)$ would involve a sum over the values x and y with the number of summations being of order $O(K^2)$.

4. Now consider computing the probability $P(E = e^*|G = g^*)$, where e^* and g^* are some specific values of E and G. As in the last question, describe the most efficient way to compute this conditional probability. The most straightforward way to do this is to first compute the joint probability

$$P(E = e, G = g^*)$$

for each possible value of E; and to then compute the conditional probability of interest, $P(E = e^*|G = g^*)$ via Bayes rule using the joint probabilities.

In the last 2 problems above you will want to start by using the law of total probability (LTP) and introducing variables that lie on the path between the nodes in the expression you are trying to compute. For example, in part 3 in may be helpful to first write $P(G = g^*|A = a^*)$ as a sum over F's values using LTP, and then proceed by seeing what needs to be computed next, and so on.