6.790 Practice Material for Final Exam

Revision: 12/7/24 5:30 0PM

Note: these questions are gathered from several past years' exams, and they concentrate only on material from the second part of the course. They do not cover generalization bounds, missing data, PCA (but those topics may appear on our exam).

Question 6 uses graphical model notation we haven't really discussed but you might still find it useful to think about in the context of density estimation.

Finally, this is longer than a final exam – think of it just as a collection of problems.

Remember that the final exam will be cumulative and more directly cover material from the **second** half of the course.

Representation Learning

1. (a) Given an image x, in contrastive learning, positives are created by applying a set of transformations. After pre-training, suppose the downstream evaluation task is to predict image rotation in the set $\{0, 90, 180, 270\}$ degrees. Which of the following transformations should not be used during pre-training with contrastive learning: cropping, rotations, adding gaussian noise to image pixels, image translation.

   (b) Assume a pre-trained *transformer*-based LLM model, $p_\theta(x_t|x_{t-1:t-T})$, where $x_t \in \mathfrak{R}^N$ denotes the t^{th} word and θ are the model parameters. Also, assume a pre-trained *transformer* vision model that converts an image into a set of K feature vectors (or visual tokens): $v_{1:K}$, where $v_k \in \mathfrak{R}^M$. The goal is to use these two models and a *small* image-caption dataset to train a captioning system. Assuming f_ϕ is a learnable linear function, which of the following options is a reasonable way of training the captioning system keeping in mind the small training dataset and why:
   Option 1: $\max_{\theta,\phi} \mathbb{E}_{x,v} p_\theta\left(x_t|f_\phi(x_{t-1:t-T}), \{(v_k)\}_{k=1}^K\right)$
   Option 2: $\max_{\phi} \mathbb{E}_{x,v} p_\theta\left(x_t|x_{t-1:t-T}, \{f_\phi(v_k)\}_{k=1}^K\right)$
   Option 3: $\max_{\phi,\theta} \mathbb{E}_{x,v} p_\theta\left(x_t|x_{t-1:t-T}, \{f_\phi(v_k)\}_{k=1}^K\right)$
   Option 4: None of the above as linear f is not expressive enough.

   (c) Predicting the full input (x) from a masked version (x_M) is a state-of-the-art method for feature learning (e.g., Masked Auto-Encoders). Consider re-purposing the learned

features for solving an image classification problem. Since the features were trained using masked images, and the new task requires processing non-masked images – using the pre-trained features presents a distribution shift problem. Which among the two pre-training strategies would you prefer to mitigate distribution shift and why?
Option 1: Reduce the percentage of masked pixels in x_M to be $\leqslant 1\%$ pixels.
Option 2: With probability 0.2 do not mask any pixels in x_M. Otherwise mask 75% pixels.

Mixtures, ELBO

2. Consider a simple mixture model involving k spherical Gaussians in two dimensions. So $x \in \mathcal{R}^2$ and

$$P(x|\theta) = \sum_{z=1}^{k} P(z|\theta)P(x|z,\theta) = \sum_{z=1}^{k} p_z N(x; \mu_z, \sigma_z^2 I)$$

Our goal is to find parameters that maximize the log-likelihood of n points, i.e., $\ell(\theta) = \sum_{i=1}^{n} \log P(x^i|\theta)$. If we used stochastic gradient ascent, we would select a data point x^i at random, and update $\theta \leftarrow \theta + \eta \nabla_\theta \log P(x^i|\theta)$. We wish to express a stochastic algorithm using the ELBO criterion and its alternating optimization view. Recall the ELBO expression

$$\text{ELBO}(Q;\theta) = \sum_{i=1}^{n} \left\{ \sum_{z=1}^{k} Q(z|x^i) \log \left[p_z N(x^i; \mu_z, \sigma_z^2 I) \right] + H(Q_{z|x^i}) \right\} = \sum_{i=1}^{n} \text{ELBO}^i(Q_{\cdot|x^i}; \theta)$$

(a) What is the number of adjustable parameters (degrees of freedom) in this mixture model?

(b) Given a selected x^i, how do we choose $Q(z|x^i)$? Write your answer explicitly using only expressions already provided above.

(c) Based on the selected x^i and your choice of $Q(z|x^i)$ above, we would like to make a stochastic gradient update of μ_1 using the ELBO^i criterion. Please select all the correct statements about our ELBO^i update of μ_1:

( ) For the purposes of updating μ_1, $Q(z|x^i)$ is treated as fixed
( ) The update of μ_1 will depend only on $Q(1|x^i)$, μ_1, σ_1^2 and x^i
( ) After updating μ_1 and hence θ, $\text{ELBO}^i(Q_{\cdot|x^i}; \theta) > \log P(x^i|\theta)$
( ) After updating μ_1 and hence θ, $\text{ELBO}^i(Q_{\cdot|x^i}; \theta) = \log P(x^i|\theta)$

(d) Does the stochastic ELBO^i update of μ_1 keep it equivalent to the stochastic gradient update of the original log-likelihood $\log P(x^i|\theta)$ with respect to μ_1? (Y/N)

Diffusion models

3. Diffusion models operate by adding noise to clean examples at varying levels and learning to predict what the added noise was. This enables the models to later sample new images by iteratively de-noising noisy starting points towards clean examples. Our dataset of clean examples is given by $\{x_0^i\}$, which can be written as a distribution $q(x_0)$.

   Let x_t be a noisy image resulting from a clean example x_0. This transformation can can be described as

   $$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim N(0, I)$$

   where $\bar{\alpha}_t$ is a function of the specific noise level at time t. A simple estimation criterion for de-noising diffusion models minimizes

   $$E\{ \|\epsilon_\theta(x_t, t) - \epsilon\|^2 \}$$

   with respect to the parameters θ of the de-noising model $\epsilon_\theta(x_t, t)$.

   (a) Let's define what the expectation is over in the estimation criterion. Specify all the variables that are random, and the distributions that their values are sampled from.

   (b) Suppose we only had a single data point x_0 yet used the estimation criterion to estimate a diffusion model. What is the solution to $\epsilon_\theta(x_t, t)$ that we would get?

Domain Adaptation

4. Consider an unsupervised domain adaptation task (co-variate shift assumption). We have a large number of labeled source domain examples therefore effectively having access to $P_S(x, y) = P_S(x)P_S(y|x)$. We also have a large number of unlabeled examples from the target domain so we can construct $P_T(x)$. Assume also that the support of $P_T(x)$ is contained within the support of $P_S(x)$.

   We are interested in evaluating the target risk of any *class-label classifier* h. To this end, a friend of ours was kind enough to train a probabilistic *domain classifier*, i.e., a classifier that takes in x and outputs the probability that x came from domain S or T. This is different from the class-label classifier that outputs our target label y.

Unfortunately for us, in training the domain classifier, the friend assumed that target examples were twice as likely to occur overall than the source examples. The domain classifier was trained to maximize the log-probability of the correct domain for each x when x was sampled from $P_T(x)$ twice as often as from $P_S(x)$.

(a) Write down an expression for the friend's probabilistic domain classifier $Q(d = T|x)$ as a function of $P_S(x)$ and $P_T(x)$.

(b) We wish to use the domain classifier to evaluate the target risk of any class-label classifier $h(x)$. You can assume that $\text{Loss}(y, h(x))$ is given. Provide an expression for the target risk of h as a function of $P_S(x)$, $P_S(y|x)$, h, $\text{Loss}(y, h(x))$, and the friend's domain classifier $Q(d = T|x)$.

Probable cause

5. You have a classification problem, but your training examples are only labeled with probabilities, so your data set consists of pairs $(x^{(i)}, p^{(i)})$, where $p^{(i)}$ is the probability that $x^{(i)}$ belongs to class 1.

   You want to train a neural network **with a single unit** to predict these probabilities.

   (a) What is a good choice for the activation function of your final output unit?

   (b) You can think of the training label $p^{(i)}$ as specifying a true probability and the current output of your neural network as specifying an approximate probability $q^{(i)}$. You think a reasonable objective would be to minimize the KL divergence $\text{KL}(p \parallel q)$ between the distributions implicitly represented by the predicted and target outputs.

   Find a simple expression for the error on training example i. You can write q for the actual output of the network (no need to make the dependence on weight and activation function explicit in this answer.)

(c) If $q^{(i)} = f(w \cdot x^{(i)})$ where f is your activation function, what is the SGD (stochastic gradient descent) weight update rule when using the KL objective function above? For simplicity, assume that $x^{(i)}$ and w are both scalars and write f' for the derivative of f.

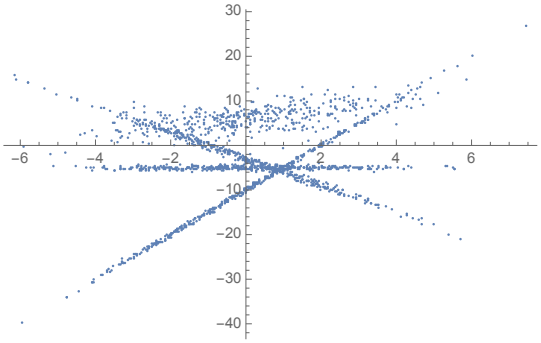
Latent variables

6. Dana has a data set in which each element is a pair of floating-point numbers and thinks, from domain knowledge, it is reasonable to model it as a mixture of Gaussians with 4 clusters. Match each of the graphical models to the data set it is most suitable for. In the graphical models, let X_1 be a random variable modeling the first component of each data element and X_2 be a random variable modeling the second component. We will describe the four clusters using two binary random variables C_1 and C_2, so that a setting of both variables determines the cluster. When we write two random variable names in a single node, we mean that the node models the joint distribution on those two variables directly.

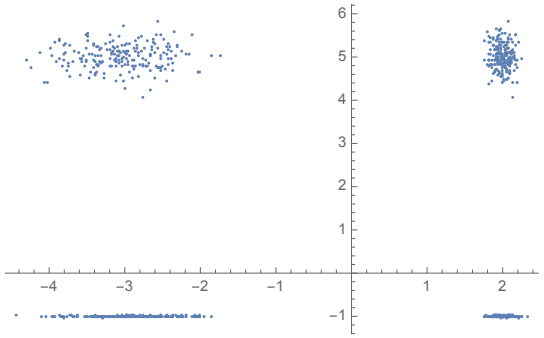
   Any continuous-valued random variable that is conditioned on nothing or only on discrete parents can be assumed to have a normal distribution conditioned on the parents. Any continuous-valued random variable A conditioned on continuous random variables B is assumed to be a linear regression model in which $A \sim \mathcal{N}(BW, \Sigma)$ where W is a weight matrix and Σ a fixed covariance matrix.

   Here are the data sets:

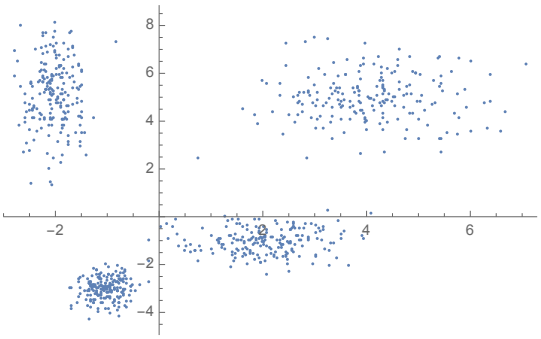
A.



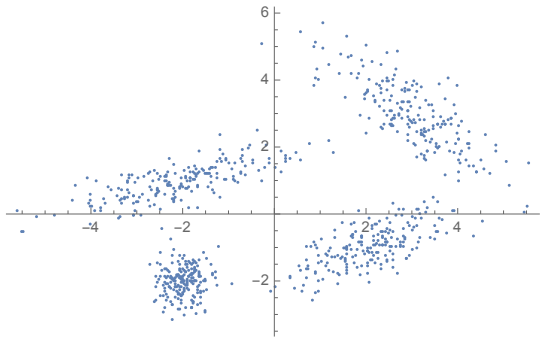
B.



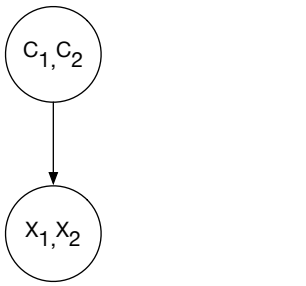
C.



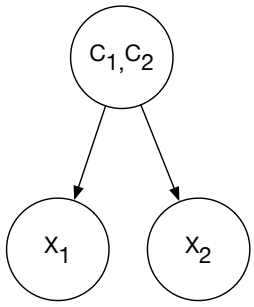
D.



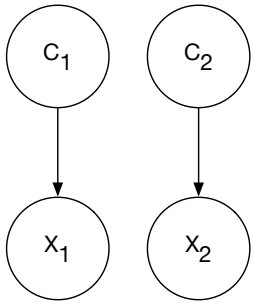
For each of the graphical models below, **indicate all of the data sets** that could have been generated for some setting of the parameters of that model.



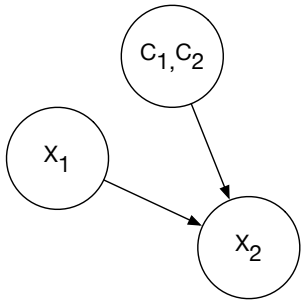
◯ A ◯ B ◯ C ◯ D



◯ A ◯ B ◯ C ◯ D



◯ A ◯ B ◯ C ◯ D



◯ A ◯ B ◯ C ◯ D

Clustering bit vectors

7. We have a data set made up of N vectors, each containing d binary feature values. We want to cluster it into K clusters. We believe the individual bits x_j in a vector x are independent given the cluster. That is:

$$C \sim \text{Multinoulli}(\pi_1, \ldots, \pi_K)$$
$$X_j \mid C = k \sim \text{Bernoulli}(\theta_{jk})$$

Our goal is to find parameter set π, θ that maximizes the likelihood of a data set $x^{(1)}, \ldots, x^{(n)}$, each of which is a vector of d binary values.

Assume you start with a set of parameters π^0 and θ^0 and decide to apply EM.

(a) The E step computes variables $\gamma_k^{(i)} = P(C = k \mid x^{(i)}, \theta^0, \pi^0)$. Provide the formula for computing their values.

The first M step is as follows:

$$\pi_k^1 := \frac{\sum_i \gamma_k^{(i)}}{N} \qquad \theta_{jk}^1 := \frac{\sum_i \gamma_k^{(i)} x_j^{(i)}}{\sum_i \gamma_k^{(i)}}$$

Recall that, for the general purposes of EM, we have defined

$$Q(\theta, \theta^{\text{old}}) = \sum_Z p(Z \mid X, \theta^{\text{old}}) \log p(X, Z \mid \theta)$$

where X are observable variables, Z are latent variables, and θ are parameters.

(b) What maximization problem is solved in the M step to get this result? Check all correct entries below.

- ◯ $\pi^1, \theta^1 = \arg\max_{\pi,\theta} E_C P(\mathcal{D} \mid C, \pi, \theta)$
- ◯ $\pi^1, \theta^1 = \arg\max_{\pi,\theta} E_C P(\mathcal{D}, C \mid \pi, \theta)$
- ◯ $\pi^1, \theta^1 = \arg\max_{\pi,\theta} Q(\langle\theta, \pi\rangle, \langle\theta^0, \pi^0\rangle)$
- ◯ $\pi^1, \theta^1 = \arg\max_{\pi,\theta} \sum_C p(C \mid \mathcal{D}, \theta^0, \pi^0) \log p(\mathcal{D}, C \mid \theta, \pi)$
- ◯ $\pi^1, \theta^1 = \arg\max_{\pi,\theta} \sum_C p(C \mid \mathcal{D}, \theta, \pi) \log p(\mathcal{D}, C \mid \theta^0, \pi^0)$

(c) What happens if you begin EM with $\pi^0 = [.5, .5]$ and $\theta_{jk}^0 = 0.5$ for all j and k?

(d) Here is a concrete example in which we are trying to cluster 4 2-dimensional vectors into 2 clusters using EM. Our data $\mathcal{D} = \{10, 11, 11, 00\}$. In the first E step, we found that $\gamma_1^{(1)} = \gamma_1^{(2)} = \gamma_1^{(3)} = 0.99999$ and $\gamma_1^{(4)} = 0.000001$. (Note that $\gamma_2^{(i)} = 1 - \gamma_1^{(i)}$.)
As a result of the M step, approximately what are the values of the θ^1 and π^1 parameters?

(e) Approximately what is the likelihood $p(\mathcal{D} \mid \theta^1, \pi^1)$? Please write an expression that would evaluate to a number, but you don't need to calculate the value.

(f) Compare the effects of a single-point cluster in this model to the effects of a single-point cluster in a Gaussian mixture.

(g) Given the class of models described at the beginning of this question and a data set, are there multiple maximum-likelihood solutions? If so, are they all desirable? Explain what they are, or why it does not have them.

(h) Quinn is really attached to the gradient descent code from 6.867 and wants to use it instead of EM to find the maximum likelihood values for π and θ. Is that a reasonable plan? Why or why not? (Please ignore computation time.)

(i) Jessie thinks this model's independence assumptions are too strong, and instead wants to

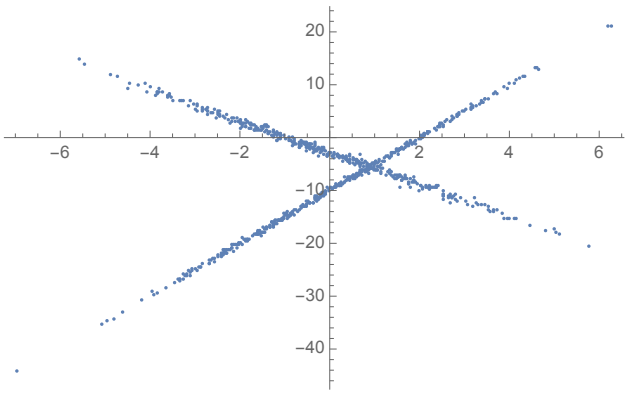
interpret each bit vector as an integer in $[1, \ldots, 2^d]$, and use the following model:

$$C \sim \text{Multinoulli}(\pi_1, \ldots, \pi_K)$$
$$X \mid C = k \sim \text{Multinoulli}(\theta_{1k}, \ldots, \theta_{2^d,k})$$

Is this a reasonable model for clustering? If so, would EM be a good solution? If not, explain why not.

Blurred lines

8. Chris has some data that looks like this.



We'd like to model a conditional distribution $P(Y \mid X)$. It seems like a bad idea to do a single linear regression, so Chris has the idea of making a mixture of linear regressions. We will consider a simple case, where the input $x^{(i)}$ is one-dimensional, there are only two components in the mixture (modeled with random variable C with values in $\{0, 1\}$), and the variance σ^2 is known. Here is the distributional model.

$$C \sim \text{Bernoulli}(\pi)$$
$$Y \mid X, C \sim \text{Normal}(w_{0c} + w_{1c}x, \sigma^2)$$

So, this model has 5 parameters: $\pi, w_{00}, w_{01}, w_{10}, w_{11}$.

(a) Assume we are given data $D = \{(x^{(i)}, y^{(i)})\}$ and parameter guesses $\widehat{\theta} = (\widehat{\pi}, \widehat{w_{00}}, \widehat{w_{01}}, \widehat{w_{10}}, \widehat{w_{11}})$ Let $\gamma_c^{(i)}$ be the probability that training example i is assigned to component c, that is, for all i,

$$\gamma_c^{(i)} = P(C^{(i)} = c \mid x^{(i)}, y^{(i)}; \widehat{\theta})$$

Write the expression used in the E step to compute $\gamma_c^{(i)}$ ($c \in \{0, 1\}$) from the data and $\widehat{\theta}$.

(b) Describe an optimization problem that must be solved to compute new values for $\widehat{w}$ from the γ values and D.

It should be a detailed expression in terms of π, w, $\gamma^{(i)}$, $x^{(i)}$ and $y^{(i)}$, but you don't have to solve for the new w.

Great responsibility

9. You may remember that non-parametric models are actually models with lots of parameters, whose capacity can adjust to fit the amount of data and its apparent complexity. With such great power comes great responsibility: to avoid overfitting.

   For each of the following non-parametric models, describe a parameter in the set-up that can be varied to control complexity and whether it should be increased or decreased to obtain a simpler model **or** state that there is not a problem with overfitting and explain briefly why.

   (a) Gaussian processes

   (b) Bayesian clustering with uncertainty about the number of clusters

   (c) Decision trees

Processing Gaussians

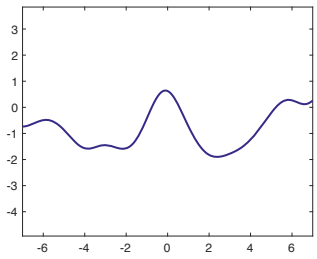
10. Recall that a Gaussian process is parameterized by a covariance function $K(x, x')$ and a mean function f that we will assume to be $f(x) = 0$. Suppose we are using a covariance function of the form

$$K(x, x') = \theta \ \exp(-\frac{1}{2\ell^2}(x - x')^2)$$

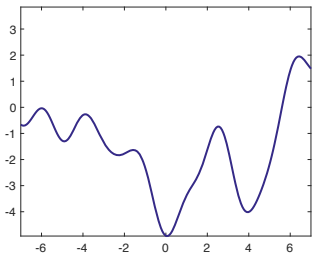
    where x, and x' are real valued scalars, ℓ denotes the length scale and θ denotes a variance multiplier. Each of the plots below shows a random function generated from a GP with hyper-parameters (θ, ℓ).

    We sampled functions from 4 different GP models.

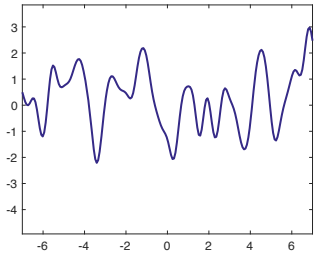
A.



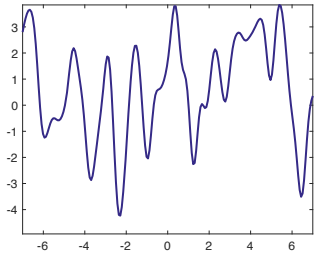
B.



C.



D.



For each parameter pair, select the plot that represents a random draw from the GP defined by those parameters.

i. $(\theta, \ell) = (2.2, 0.3)$ ◯ A ◯ B ◯ C ◯ D

ii. $(\theta, \ell) = (2.2, 1.0)$ ◯ A ◯ B ◯ C ◯ D

iii. $(\theta, \ell) = (1.0, 1.0)$ ◯ A ◯ B ◯ C ◯ D

iv. $(\theta, \ell) = (1.0, 0.3)$ ◯ A ◯ B ◯ C ◯ D