CPSC 440/540 Machine Learning (January-April, 2022)
Assignment 4 (due Friday April 1st at midnight)

For this assignment you can work in groups of 1-2. However, please only hand in one assignment for the group. It is possible that some questions on this assignment will be cancelled, depending on the pace of lectures.

1. Name(s):
2. Student ID(s):

1 Gaussians

1.1 Gaussian Self-Conjugacy

Consider n IID samples x^i distributed according to a Gaussian with mean μ and covariance $\sigma^2 I$,

$$x^i \sim \mathcal{N}(\mu, \sigma^2 I).$$

Assume that μ itself is distributed according to a Gaussian

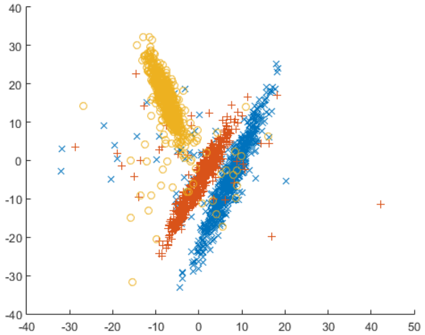
$$\mu \sim \mathcal{N}(\mu_0, \Sigma_0),$$

with mean μ_0 and (positive-definite) covariance Σ_0. In this setting, the posterior for μ also follows a Gaussian distribution.

Derive the form of the posterior distribution, $p(\mu \mid X, \sigma^2, \mu_0, \Sigma_0)$.
Hint: the posterior is a product of Gaussian densities.

1.2 Generative Classifiers with Gaussian Assumption

Consider the 3-class classification dataset in this image:



In this dataset, we have 2 features and each colour represents one of the classes. Note that the classes are highly-structured: the colours each roughly follow a Gausian distribution plus some noisy samples.

Since we have an idea of what the features look like for each class, we might consider classifying inputs x using a *generative classifier*. In particular, we are going to use Bayes rule to write

$$p(y^i = c \mid x^i, \Theta) = \frac{p(x^i \mid y^i = c, \Theta) \cdot p(y^i = c \mid \Theta)}{p(x^i \mid \Theta)},$$

where Θ represents the parameters of our model. To classify a new example $\tilde{x}^i$, generative classifiers would use

$$\hat{y}^i = \underset{y \in \{1,2,\ldots,k\}}{\arg\max}\; p(\tilde{x}^i \mid y^i = c, \Theta) p(y^i = c \mid \Theta),$$

where in our case the total number of classes k is 3.[1] Modeling $p(y^i = c \mid \Theta)$ is easy: we can just use a k-state categorical distribution,

$$p(y^i = c \mid \Theta) = \theta_c,$$

where θ_c is a single parameter for class c. The maximum likelihood estimate of θ_c is given by n_c/n, the number of times we have $y^i = c$ (which we've called n_c) divided by the total number of data points n.

Modeling $p(x^i \mid y^i = c, \Theta)$ is the hard part: we need to know the *probability of seeing the feature vector* x^i *given that we are in class* c. This corresponds to solving a density estimation problem for each of the k possible classes. To make the density estimation problem tractable, we'll assume that the distribution of x^i given that $y^i = c$ is given by a $\mathcal{N}(\mu_c, \Sigma_c)$ Gaussian distribution for a class-specific μ_c and Σ_c,

$$p(x^i \mid y^i = c, \Theta) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x^i - \mu_c)^T \Sigma_c^{-1} (x^i - \mu_c)\right).$$

Since we are distinguishing between the probability under k different Gaussians to make our classification, this is called *Gaussian discriminant analysis* (GDA). In the special case where we have a constant $\Sigma_c = \Sigma$ across all classes it is known as *linear discriminant analysis* (LDA) since it leads to a linear classifier between any two classes (while the region of space assigned to each class forms a convex polyhedron as in k-means clustering and softmax classification). Another common restriction on the Σ_c is that they are diagonal matrices, since this only requires $O(d)$ parameters instead of $O(d^2)$ (corresponding to assuming that the features are independent univariate Gaussians given the class label). Given a dataset $\mathcal{D} = \{(x^i, y^i)\}_{i=1}^n$,

[1] The denominator $p(\tilde{x}^i \mid \Theta)$ is irrelevant to the classification since it is the same for all y.

where $x^i \in \mathbb{R}^d$ and $y^i \in \{1, \dots, k\}$, the maximum likelihood estimate (MLE) for the μ_c and Σ_c in the GDA model is the solution to

$$\operatorname*{arg\,max}_{\mu_1,\mu_2,\dots,\mu_k,\Sigma_1,\Sigma_2,\dots,\Sigma_k} \prod_{i=1}^{n} p(x^i \mid y^i, \mu_{y^i}, \Sigma_{y^i}).$$

This means that the negative log-likelihood will be equal to

$$\begin{aligned} -\log p(X \mid y, \Theta) &= -\sum_{i=1}^{n} \log p(x^i \mid y^i, \mu_{y^i}, \Sigma_{y^i}) \\ &= \sum_{i=1}^{n} \frac{1}{2}(x^i - \mu_{y^i})^T \Sigma_{y^i}^{-1}(x^i - \mu_{y^i}) + \frac{1}{2}\sum_{i=1}^{n} \log|\Sigma_{y^i}| + \text{const.} \end{aligned}$$

In class we stated the MLE for this model under the assumption that we use full covariance matrices and that each class has its own covariance.

1. Derive the MLE for the GDA model under the assumption of *common diagonal covariance* matrices, $\Sigma_c = D$ (d parameters). (Each class will have its own mean μ_c.)

2. Derive the MLE for the GDA model under the assumption of *individual scaled-identity* matrices, $\Sigma_c = \sigma_c^2 I$ (k parameters).

3. When you run *example_generative* it loads a variant of the dataset in the figure that has 12 features and 10 classes. This data has been split up into a training and test set, and the code fits a k-nearest neighbour classifier to the training set then reports the accuracy on the test data (around $\sim 63\%$ test error). The k-nearest neighbour model does poorly here since it doesn't take into account the Gaussian-like structure in feature space for each class label. Write a function *gda* that fits a GDA model to this dataset (using individual full covariance matrices). Hand in the function and report the test set accuracy.

4. In this question we would like to replace the Gaussian distribution of the previous problem with the more robust multivariate-t distribution so that it isn't influenced as much by the noisy data. Unlike the previous case, we don't have a closed-form solution for the parameters. However, if you run *example_student* it generates random noisy data and fits a multivariate-t model. By using the *studentT* model, write a new function *tda* that implements a generative model that is based on the multivariate-t distribution instead of the Gaussian distribution. Report the test accuracy with this model.

Hints: you may be able to substantially simplify the notation in the MLE derivations if you use the notation $\sum_{i \in y_c}$ to mean the sum over all values i where $y^i = c$. Similarly, you can use n_c to denote the number of cases where $y_i = c$, so that we have $\sum_{i \in y_c} 1 = n_c$. Note that the determinant of a diagonal matrix is the product of the diagonal entries, and the inverse of a diagonal matrix is a diagonal matrix with the reciprocals of the original matrix along the diagonal.

For the implementation you can use the result from class regarding the MLE of a general multivariate Gaussian. At test time for GDA, you may find it more numerically reasonable to compare log probabilities rather than probabilities of different classes, and you may find it helpful to use the *logdet* function to compute the log-determinant in a more numerically-stable way than taking the log of the determinant. (Also, don't forget to center at training and test time.)

For the last question, you may find it helpful to define an empty array that can be filled with k *DensityModel* objects using:

```
subModel = Array{DensityModel}(undef,k)
```

1.3 Empirical Bayes

Consider the model

$$y^i \sim \mathcal{N}(w^T z^i, \sigma^2), \quad w_j \sim \mathcal{N}(0, \lambda^{-1}),$$

where z^i is a length-k non-linear transformation of the features x^i (like a polynomial basis or RBFs). The posterior distribution in this model has the form

$$w \sim \mathcal{N}(w^+, \Theta^{-1}),$$

where the posterior precision and mean are given by

$$\begin{aligned} \Theta &= \frac{1}{\sigma^2} Z^T Z + \lambda I, \\ w^+ &= \frac{1}{\sigma^2} \Theta^{-1} Z^T y, \end{aligned}$$

and Z contains the z^i vectors in the rows. The marginal likelihood in this model is given by

$$p(y \mid X, \sigma^2, \lambda) = \frac{(\lambda)^{k/2}}{(\sigma\sqrt{2\pi})^n |\Theta|^{1/2}} \exp\left(-\frac{1}{2\sigma^2}\|Zw^+ - y\|^2 - \frac{\lambda}{2}\|w^+\|^2\right).$$

As discussed in class, the marginal likelihood can be used to optimize hyper-parameters like σ, λ, and even the basis Z.

The demo *example_basis* loads a dataset and fits a degree-2 polynomial to it. Normally we would use a test set to choose the degree of the polynomial but here we'll use the marginal likelihood of the training set. Write a function, *leastSquaresEmpiricalBaysis*, that uses the marginal likelihood to choose the degree of the polynomial as well as the parameters λ and σ (you can restrict your search for λ and σ to powers of 2). Hand in your code and report the marginally most likely values of the degree, σ, and λ.

2 Markov Models

2.1 Inference with Discrete States

The function *example_markov.jl* loads the initial state probabilities and transition probabilities for a Markov chain model,

$$p(x_1, x_2, \dots, x_d) = p(x_1) \prod_{j=2}^{d} p(x_j \mid x_{j-1}),$$

corresponding to the "grad student Markov chain" from class.

1. Write a function, *sampleAncestral*, that uses ancestral sampling to sample a sequence x from this Markov chain of length d. Hand in this code and report the univariate marginal probabilities for time 50 using a Monte Carlo estimate based on 10000 samples.
   Hint: you can use *sampleDiscrete* in *misc.jl* to sample from a discrete probability mass function using the inverse transform method.

2. Write a function, *marginalCK*, that uses the CK equations to compute the exact univariate marginals up to a given time d. Hand in this code, report all exact univariate marginals at time 50, and report how this differs from the marginals in the previous question.
3. What is the state c with highest marginal probability, $p(x_j = c)$, for each time j?
4. Write a function, *viterbiDecode*, that uses the Viterbi decoding algorithm for Markov chains to find the optimal decoding up to a time d. Hand in this code and report the optimal decoding of the Markov chain up to time 50 and up to 100.

5. Report all the univariate conditional probabilities at time 50 if the student starts in grad school, $p(x_{50} = c \mid x_1 = 3)$ for all c. Hint: you should be able to do this by changing the input to the CK equations.

6. Report for all c the univariate conditional probabilities $p(x_5 = c \mid x_{10} = 6)$ ("where you were likely to be 5 years after graduation if you ended up in academia after 10 years") obtained using a Monte Carlo estimate based on 10000 samples and rejection sampling. Also report the number of samples accepted among the 10000 samples.

7. Give code implementing a dynamic programming approach to exactly compute $p(x_5 = c \mid x_{10} = 6)$, and report the exact values for all c.

8. Why is $p(x_j = 7 \mid x_{10} = 6)$ equal to zero for all j less than 10?

Hint: for some of the quesitons you may find it helpful to use a k by d matrix M to represent a dynamic programming table

2.2 Markov Chain Monte Carlo

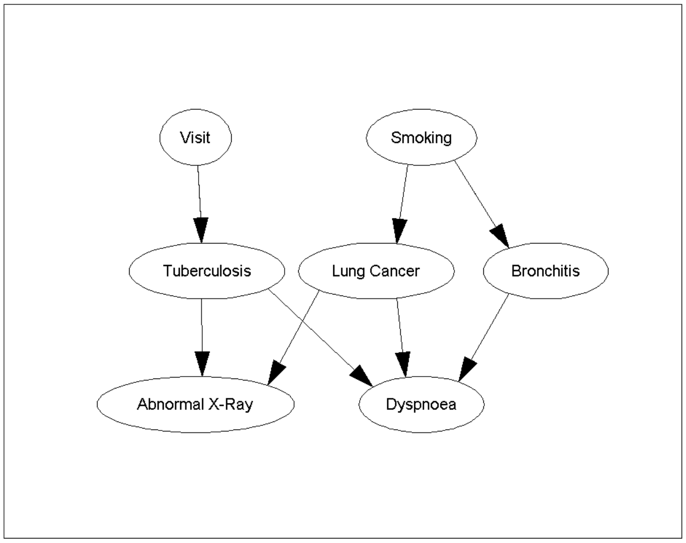
If you run *example_MH.jl*, it loads a set of images of '2' and '3' digits. It then runs the Metropolis MCMC algorithm to try to generate samples from the posterior over w, in a logistic regression model with a Gaussian prior. Once the samples are generated, it makes a histogram of the samples for several of the variables.[2]

1. Why would the samples coming from the Metropolis algorithm not give a good approximation to the posterior?
2. Modify the proposal used by the demo to $\hat{w} \sim \mathcal{N}(w, (1/100)I)$ instead of $\hat{w} \sim \mathcal{N}(w, I)$. Hand in your code and the update histogram plot.
3. Modify the proposal to use $\hat{w} \sim \mathcal{N}(w, (1/10000)I)$. Do you think this performs better or worse than the previous choice? (Briefly explain.)

[2] The "positive" variables are some of the positive weights when you fit an L2-regularized logistic regression model to the this data. The "negative" variables are some of the negative regression weights in that model, and the "neutral" ones are set to 0 in that model.

2.3 D-Separation

Consider the DAG model below, for distinguishing between different causes of shortness-of-breath (dyspnoea) and the causes of an abnormal lung x-ray, while modelling potential causes of these diseases too (whether the person is a smoker or had a 'visit' to a country with a high degree to tuberculosis).



We'll assume that the distribution over the variables is "faithful" to the graph, meaning that variables are conditionally independent if and only if the graph structure implies their independence. Under this assumption, say why each of the following conditional independence statements is true or false (provide a very-brief justification):

1. (Smoking) $\perp$ (Bronchitis).
2. (Smoking) $\perp$ (Dyspnoea).
3. (Tuberculosis) $\perp$ (Lung Cancer).
4. (Abnormal X-Ray) $\perp$ (Dyspnoea).
5. (Abnormal X-Ray) $\perp$ (Visit) | (Tuberculosis).
6. (Bronchitis) $\perp$ (Lung Cancer) | (Smoking).
7. (Tuberculosis) $\perp$ (Lung Cancer) | (Dyspnoea).
8. (Visit,Smoking) $\perp$ (Abnormal X-Ray, Dyspnoea) | (Tuberculosis, Lung Cancer, Bronchitis)
9. (Smoking) $\perp$ (Visit) | (Dyspnoea).
10. (Tuberculosis) $\perp$ (Bronchitis) | (Abnormal X-Ray).

3 Mixture Models

Cancelled.

4 Very-Short Answer Questions

1. What is the relationship between using a product of Gaussians and using a multivariate Gaussian?
2. How does the affine property allow us to sample from multivariate Gaussians?
3. With a Gaussian likelihood, a Gaussian prior for its mean, and a fixed covariance, how do we know that the posterior predictive will also be a Gaussian?
4. How do the sparsity patterns of the covariance and precision matrix in a multivariate Gaussian relate to independence and conditional independence in the models?
5. In linear discriminant analysis, why might we not assign a point to its closest mean?

6. The maximum of the posterior predictive in Bayesian linear regression gives the same prediction as if we used the MAP estimate. What is a reason we might use Bayesian linear regression anyways?
7. What is the key difference between Monte Carlo approximations and variational approximations?
8. What is a key advantage of "end-to-end" training of computer vision system?
9. What are your 3 favourite properties of the exponential family?
10. What is the difference between computing marginals and computing the stationary distribution of a Markov chain.

11. What is the cost of generating a sample from a Markov chain of length d with k possible states for each time? What is the cost of decoding?

12. In what setting is it unnecessary to include the q function in the Metropolis-Hastings acceptance probability?

13. What is "explaining away"?

14. If two variables are not d-separated, are they necessarily dependent? If two variables are d-separated, are they necessarily independent?

15. Describe how we could use ancestral sampling to sample from the joint density over x and y defined by a Gaussian discriminant analysis model.

GPU Access

We applied for UBC to fund some GPUs (RTX 3070 8GB) and a lot of RAM (192 GB) for use by students for their coursework. This is the first semester that these resources will be used, so there may be a learning curve to use these (from your side and our side). If you would like to use these for your course project, please make a private post on Piazza with the following information:

1. Names of people on the project.
2. Estimated number of GPU/CPU hours.
3. Roughly how much memory your jobs will need.
4. Roughly how much disk usage you will need.
5. Are there datasets you plan to use that other groups might also be using?