CPSC 440/540 Machine Learning (January-April, 2022)
Assignment 3 (due Friday March 11th at midnight)

For this assignment you can work in groups of 1-2. However, please only hand in one assignment for the group.

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

1 Bayesian Learning

1.1 Conjugate Priors

Consider counts $y \in \{0, 1, 2, 3, \dots\}$ following a Poisson distribution with rate parameter $\lambda > 0$,

$$p(y \mid \lambda) = \frac{\lambda^y \exp(-\lambda)}{y!}.$$

We'll assume that λ follows a Gamma distribution (the conjugate prior to the Poisson) with shape parameter $\alpha > 0$ and rate parameter $\beta > 0$,

$$\lambda \sim \text{Gamma}(\alpha, \beta),$$

or equivalently that

$$p(\lambda \mid \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} \exp(-\beta\lambda),$$

where Γ is the gamma function.

Compute the following quantites:

1. The posterior distribution,
$$p(\lambda \mid y, \alpha, \beta).$$

2. The marginal likelihood of y given the hyper-parameters α and β,
$$p(y \mid \alpha, \beta) = \int p(y, \lambda \mid \alpha, \beta) d\lambda.$$

3. The posterior mean estimate for λ,

$$\mathbb{E}_{\lambda \mid y,\alpha,\beta}[\lambda] = \int \lambda p(\lambda \mid y, \alpha, \beta) d\lambda.$$

4. The posterior predictive distribution for a new independent observation $\tilde{y}$ given y,

$$p(\tilde{y} \mid y, \alpha, \beta) = \int p(\tilde{y}, \lambda \mid y, \alpha, \beta) d\lambda.$$

Hint: You should be able to use the form of the gamma distribution to solve all the integrals that show up in this question. You can use $Z(\alpha, \beta) = \frac{\Gamma(\alpha)}{\beta^\alpha}$ to represent the normalizing constant of the gamma distribution, and use α^+ and β^+ as the updated parameters of the gamma distribution in the posterior.

1.2 Empirical Bayes

If you run the script *example_groups.jl* it will load a dataset representing the number of people diagnosed with stomach cancer in a set of different regions, and the number of people at risk in the region. The format of the data is as follows:

- `n[j,1]` is number of positive examples in the training set for group j.
- `n[j,2]` is the total number of examples in the training set for group j.
- `nTest[j,1]` is number of positive examples in the testing set for group j.
- `nTest[j,1]` is the total number of examples in the testing set for group j.

The code first shows the negative log-likelihood (NLL) on the test set if we assume that the probability of getting stomach cancer among each group is 0.5. The NLL under this choice is very high, since the number of positives is fairly low (remember that we want to have a low NLL, corresponding to a high likelihood). The code then computes the MLE for each group, and evaluates the NLL with this choice.

1. Why do we get `NaN` for the NLL with the MLE values? What is the actual test likelihood supposed to be in this case?
2. Laplace smoothing corresponds to MAP estimation with a beta prior that has $\alpha = 2$ and $\beta = 2$. What is the test NLL if we use Laplace smoothing to estimate the parameters?
3. Intead of using MAP estimation and then computing the NLL, we could use the Bayesian approach of computing the negative logarithm of the posterior predictive probability of the test data. What is the negative log of the posterior predictive probability of the test data under a beta prior with $\alpha = 2$ and $\beta = 2$? Explain why you think this gives better/worse results than the NLL under MAP estimation.
4. Instead of modeling each group independently, consider fitting a single Bernoulli model by pooling the data across all groups. In other words, we ignore the groups and treat all the examples as if they came from a single source. Report the test NLL in this pooled setting when using MLE, MAP, and the posterior predictive. Why are these results more similar than when we use independent Bernoullis for each group?

5. In the pooled data model, you can compute the logarithm of the marginal likelihood of the hyper-parameters given the data as `logMargLik(a,b) = logbeta(a+n1,b+n0)-logbeta(a,b)` (which is the ratio of posterior and prior normalizing constants, converted to the log domain).[1] Here, `n1` is the number of 1s, `n0` is the number of 0s, `a` gives the value of α, and `b` gives the value of β. Can you find the value of α and β that optimize the marginal likelihood? Hint: it may be helpful to parameterize the beta distribution in terms of $m = \alpha/(\alpha+\beta)$ (the proportion of 1s) and $k = (\alpha+\beta)$ (the "strength" of our belief in this ratio). Try setting m to the MLE value of θ and varying k.

6. In the textbook where I got this data,[2] they suggest using an independent hyper-prior over m and k of the form
$$p(m) \propto m^{.01-1}(1-m)^{9.9-1}, \quad p(k) \propto 1/(1+k)^2.$$
The hyper-prior over m is biased towards low values (since we expect the cancer to be rare) but not particularly strong, while the prior over k is similar to what is called a "Jeffreys prior" over scale variables (which would satisfies a particularly definition of being an "uninformative prior"). In the pooled data model, find the value of α and β that optimize the marginal likelihood with this hyper-prior (or equivalently optimize the log of the marginal likelihood with the log of this prior as the regularizer). Up to one decimal place, what are the optimal values of α anbd β under this hyper-prior? What values of m and k does this correspond to choosing? Hand in your code for computing the objective function that is being optimized. Hint: for this question you can use a "brute force" approach to find α and β, by searching over all values of the from "x.y" for x in 0-9 and y in 0-9 (but where at least one of "x" and "y" must be bigger than 0.

[1] This uses the "SpecialFunctions" package to implement `logbeta`, the logarithm of the beta function (the beta function is the normalizing constant of the beta distribution).

[2] "Ordinal Data Modeling" by Johnson and Albert.

7. We can also use the (regularized) marginal log-likelihood to learn the hyper-parameters in our original setting where we had a separate parameter for each group. In this setting the marginal likelihood is given by

$$p(X \mid \alpha, \beta) \propto \prod_{j=1}^{k} \frac{Z(\alpha + n_{1j}, \beta + n_{0j})}{Z(\alpha, \beta)},$$

where n_{1j} is the number of 1s in group j, n_{0j} is the number of 0s in group j, and the normalizing constant Z is the beta function. Find the values of α and β that optimize the marginal likelihood multiplied by the hyper-prior from the previous question, up to 1 decimal place (the value "x" in the brute-force search will need to be increased). Hand in your code to compute the objective function being optimized, report the values of α and β that you find, and report the negative logarithm of the posterior predictive probability of the test data under this choice of α and β. If this is too slow, you can restrict the search to positive integer values for α and β.

8. In the model from the last question, what is the posterior predictive probability of seeing a 1 for a new group?

9. Under the choice of α and β from the previous question, what is the NLL of the test data if we use MAP estimation for the parameters?

10. In this question, many of the best-performing methods achieve similar performance. Give an explanation for why these different model choices do not make a big difference for this dataset.

2 More Deep Learning

The script *example_flux.jl* shows how the compute the loss function and gradient of a neural network with one hidden layer (with 3 units) using a variety of strategies, and how to update the parameters based on an SGD update using a variety of strategies. This includes the backpropagation code from the previous assignment, as well automatic differentiation and other functionality provided by the *Flux* package. The function does not produce any output, but you can verify that all methods return the same results up to numerical precision.

1. On the previous assignment we concatenated all parameters into one big vector of parameters w. What is an advantage of the other approaches in the script, where we a matrix W and a vector v?
2. If we use use `Dense(W)` for the first layer in the network, how many parameters does the first layer have and what do the extra parameters represent? Hint: you can use `params(layer)` to get the parameters of a layer.
3. Give code for implementing a neural network with 2 hidden layers, using the `Dense` function within the `Chain` function.
4. In one of the Flux tutorials, they use the `softmax` function as the last argument to the `Chain` function (for multi-class classification). What is a reason we might not want to include the softmax function within the `Chain` function?
5. It we use *Conv((5,5),3=>6,relu)* for a layer, why do we get a layer with 456 parameters?

6. Re-write the model and training procedure you developed for Assignment 2 Question 3.3 to use one of these alternate ways to compute the gradient and the update of the parameters (so you should not be calling *NeuralNet.jl* anymore). Re-state the changes you made, hand in your modified code, and report whether you noticed any performance differences (in terms of runtime, memory, or accuracy).

3 Very-Short Answer Questions

1. When we use categorical variables, we assume that the categories do not have any special ordering. But we when sample from a categorical distribution, we use the CDF $p(x \leq c)$. Why does it make sense to use the CDF for sampling if the categories are unordered?
2. Monte Carlo methods approximate the expectation of a function of a random variable, but we said that they can be used to approximate probabilities. What random function would you use to approximate the probability of an event?
3. For the categorical likelihood, what is the relationship between the usual parameters θ_c and the parameters $\tilde{\theta}_c$ we used to derive the MLE.
4. How do MAP and Bayesian methods differ in how make predictions?
5. Why do we say that Bayesian methods incorporate regularization, even though there is no regularization term in the posterior predictive distribution.

6. What is the difference the standard MLE and type II MLE?
7. How does using a conjugate prior simplify the marginal likelihood calculation?
8. What is a hyper-hyper-parameter?
9. Suppose we use the tabular parameterization for multi-class classification with d features, where we have k classes and each feature can take k values. What is the exact number of parameters we need if we remove redundant parameters by exploiting that probabilities sum to 1.
10. In the softmax function we have k weight vectors, one for each class. But in the special case of the sigmoid function we fix one of these weight vectors to zero, so we only have one weight vector for two classes. Explain why we would or would not get a more-general model if we used the softmax function with two weight vectors for binary classification (by "more-general", we mean "can model a larger set of functions").

11. If we use a neural network with one hidden layer for multi-class classification, what are the sizes of the parameter matrices W and V and what is the cost of backpropagation? You can use k as the number of classes and m as the number of hidden units.
12. Why can RNNs label sequence of different lengths?
13. When predicting with sequence-to-sequence RNNs we do not know the length of the output sequence. How is the length of the output sequence determined during decoding?
14. In LSTMs, what is the range (set of possible values) for a_t, o_t, c_t, f_t, i_t, and g_t under the usual choices of activation functions (h: sigmoid, h_0: tanh)?
15. Why do we use "context vectors" instead of including all encoding states in attention models?

Project Proposal(s)

As discussed in the syllabus, we are using the following marking scheme for the course:

1. CPSC 440: 50% assignments and 50% for the best among {final exam, research project, sample lecture/assignment}.
2. CPSC 540: 50% assignments, 25% for the best among {final exam, research project, sample lecture/assignment}, and 25% for the second best among those three categories.

If you are doing the research project and/or the sample lecture/assignment, the final part of this assignment is to submit a short proposal about what you will focus on. Both the project and sample lecture/assignment must be done in groups of 2-3, but you should only submit one proposal for the group. The next two subsections describe what the proposal should look like for the two types of projects. This part of the assignment will not be marked: we are doing this so that you (a) start forming project groups now, (b) you start thinking about what you want to do for the project(s), and (c) we check that the scope and topic of the project is suitable.

If you plan to do the final exam but not the research project or sample lecutre/assignment, then you do not need to submit anything for this part of the assignment. If you are doing the resaerch project and sample lecture/assignment, note that the groups do not have to be the same for both. Also note that you can do these projects on similar topics. For example, the research project might be focus on applying a particular ML method to a particular application while the lecture goes over how the ML method works. Similarly, if you have projects in other courses this term, it is ok to do them on the same topic provided that you check with the other instructor (although the submission formats and grading schemes may differ between courses).

There is flexibility in the choice of project topics even after the proposal. If you want to explore different topics you can ultimately choose to do a project that is unrelated to the one in your proposal. However, in this case you may want to check with the instructor/TAs that the scope and topic are appropriate..

Research Project

The proposal should be a maximum of 2 pages (and 1 page or half of a page is ok if you can describe your plan concisely). The proposal should be written for the instructor and the TAs, so you do not need to introduce any ML background but you will need to introduce non-ML topics. The projects must be done in groups of 2-3, and the proposal be submitted separately from the assignment on Gradescope.

There is quite a bit of flexibility in terms of the type of project you can do, as I believe there are many ways that people can make valuable contributions to research. However, note that the final deliverable will be a report containing at most 6 pages of text (the actual document can be longer due to figures, tables, references, and proofs) that emphasizes a particular "contribution" (which is "what doing the project has added to the world").

The three mains questions your project proposal needs to answer are:

1. What problem you are focusing on?
2. What do you plan to do?
3. What will be the "contribution"?

Also, note that for the course project that negative results (like "we tried something that we thought we would work in a particular setting but it didn't work") are acceptable (and often unavoidable).

Here are some standard project "templates" that you might want to follow:

- **Application bake-off**: you pick a specific application (from your research, personal interests, or maybe from Kaggle) or a small number of related applications, and try out a bunch of techniques (e.g., random forests vs. logistic regression vs. generative models). In this case, the contribution would be showing that some methods work better than others for this specific application (or your contribution could be that everything works equally well/badly).
- **New application**: you pick an application where where people aren't using ML, and you test out whether ML methods are effective for the task. In this case, the contribution would be knowing whether ML is suitable for the task (and perhaps how to prepared the data and constructed features).
- **Scaling up**: you pick a specific machine learning technique, and you try to figure out how to make it run faster or on larger datasets (for example, how do we apply kernel methods when n is very large). In this case, the contribution would be the new technique and an evaluation of its performance, or could be a comparison of different ways to address the problem.
- **Improving performance**: you pick a specific machine learning technique, and try to extend it in some way to improve its performance (for example, how can we efficiently use use non-linearities within graphical models). In this case, the contribution would be the new technique and an evaluation of its performance.
- **Generalization to new setting**: you pick a specific machine learning technique, and try to extend it to a new setting (for example, making a graphical-model version of random forests). In this case, the contribution would be the new technique and an evaluation of its performance, or could be a comparison of different ways to address the problem.
- **Perspective paper**: you pick a specific topic in ML, read a larger number of papers on the topic, then write a report summarizing what has been done on the topic and what are the most promising directions of future work. In this case, the contribution would be your summary of the relationships between the existing works, and your insights about where the field is going.
- **Coding project**: you pick a specific method or set of methods (like independent component analysis), and build an implementation of them. In this case, the contribution could be the implementation itself or a comparison of different ways to solve the problem.
- **Theory**: you pick a theoretical topic (like the variance of cross-validation or the convergence of stochastic gradient in non-smooth and non-convex setting), read what has been done about it, and try to prove a new result (usually by relaxing existing assumptions or adding new assumptions). The contribution could be a new analysis of an existing method, or why some approaches to analyzing the method will not work.

The above are just suggestions, and many projects will mix several of these templates together, but if you are having trouble getting going then it's best to stick with one of the above templates. Also note that the above includes topics not covered in the course (like random forests), so there is flexibility in the topic, but the topic should be closely-related to ML.

Sample Lecture and Assignment

Throught CPSC 340-440, we try to introduce you to a variety of fundamental concepts that we think will stand the test of time and we also try to give you an idea of what methods people are currently finding useful in a variety of settings. But unfortunately, ML is a huge field so many topics cannot be covered in only 1-2 courses. For the sample lecture/assignment, you will prepare a lecture and an assignment on a model that we do not cover in the course (preparing material is one of the best ways to learn new things, and this also helps us decide what topics to include in future offerings of the course).

The lecture should touch on most of the below topics (I planned each "part" of the course this term to follow this structure):

1. **Motivating problem**: introduce an application that motivates the model.
2. **Model definition**: how is it defined and what are the parameters (and their sizes)?
3. **General framework and other applications**: is this solving an abstract problem, and where else would this model be useful?
4. **Inference**: how do you do things like sampling, decoding, marginalization, conditioning, and test-set predictions/evaluations? (Theoretically and with code.)
5. **MLE**: how do you compute the MLE parameters? (Theoretically and with code.)
6. **MAP**: how do you introduce a prior and compute the MAP parameters?
7. **Bayes**: how do you make Bayesian predictions with the model?
8. **Multivarate** (for one-dimensional distributions): is there a multi-variable version of the model?
9. **MNIST**: what do MNIST samples look like if you use it as a density estimator?
10. **Supervised**: how do we add features to the model and use within a generative or a discriminative classifier?
11. **Deep**: how do we add layers of hidden layers to learn features?

The particular deliverables due with this assignment are:

1. Specify which model you will focus on.
2. Give a 1-sentence-maximum description of how you might cover the topics above in the model. Note: not all topics make sense for all models, so it's ok to say that you won't cover some of them.
3. Give a short outline for what the assignment related to the model will cover.

As with the research project, the sample lecture/assignment should be done in groups of 2-3 and the deliverables above should be submitted separately from the assignment. The appendix below may help you to choose a topic.

A Lecture Topic Suggestions

Below are a set of topics that I plan to cover later in the course, so you should avoid choosing these topics:

- Multivariate Gaussians and models like linear/Gaussian discriminant analysis.
- Exponential families and multivariate Laplace/student distributions.
- Rejection sampling and importance sampling.
- Markov chains and MCMC.
- Directed and undirected graphical models ("Bayesian networks" and "Markov random fields"), conditional random fields.
- Latent Dirichlet allocation and variational inference.
- Mixture models and hidden Markov models.
- Boltzmann machines and deep belief networks.

I may also cover VAEs/GANs if we have time, but feel to choose those topics as I am not sure if we will have time and these topics are changing very quickly at the moment.

Below is a set of topics that you might see in other ML courses that I will probably not cover as well as some related keywords:

- Sequential Monte Carlo.
- Non-parameteric Bayes.
- Online learning and bandits.
- Reinforcement learning.
- Privacy and security.
- Algorithm fairness.
- Independent component analysis.
- Scaling up Gaussian processes.
- Max-margin Markov networks and structured SVMs.
- Reinforcement learning.
- Causality.
- Parallel/distributed/federated training.
- Learning theory.
- Probabilistic context-free grammars.
- Probabilistic programming.
- Sub-modularity.
- Spectral methods.
- Automatic hyper-parameter tuning.

(The above topics would probably be my starting point for making a third class, and I could imagine some of the above topics being swapped into 340/440 in future offerings.) Some of these topics are too big for one lecture (such as reinforcement learning), and one of the main purposes of doing this proposal is to help you choose the appropriate scope.