Anirudh Suresh
anirudh_suresh@college.harvard.edu
CS281-F17

Assignment #2 v 1.0

Due: 5:00pm October 6, 2017

Collaborators: John Doe, Fred Doe

NOTE: you must show derivations for your answers unless a question explicitly mentions that no justification is required.

Problem 1 (Spherical Gaussian, 10pts)

One intuitive way to summarize a probability density is via the mode, as this is the "most likely" value in some sense. A common example of this is using the maximum *a posteriori* (MAP) estimate of a model's parameters. In high dimensions, however, the mode becomes less and less representative of typical samples. Consider variates from a D-dimensional zero mean spherical Gaussian with unit variance:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0}_D, \mathbb{I}_D),$$

where $\mathbf{0}_D$ indicates a column vector of D zeros and $\mathbb{I}_D$ is a $D \times D$ identity matrix.

1. Compute the distribution that this implies over the distance of these points from the origin. That is, compute the distribution over $\sqrt{\mathbf{x}^\mathsf{T}\mathbf{x}}$, if $\mathbf{x}$ is a realization from $\mathcal{N}(\mathbf{0}_D, \mathbb{I}_D)$. (Note: Consider transformations of a Gamma distribution described in Murphy 2.4.5.)
2. Make a plot that shows this probability density function for several different values of D, up to $D = 100$.
3. Make a plot of the cumulative distribution function (CDF) over this distance distribution for $D = 100$. A closed-form solution may be difficult to compute, so you can do this numerically.)
4. From examining the CDF we can think about where most of the mass lives as a function of radius. For example, most of the mass for $D = 100$ is within a thin spherical shell. From eyeballing the plot, what are the inner and outer radii for the shell that contains 90% of the mass in this case?

Solution

1. Since $x \sim \mathcal{N}(\mathbf{0}_D, \mathbb{I}_D), x^T x \sim \chi^2_1$, the multivariate chi-square distribution that represents the "sum" of one squared standard normal distribution. Then,

   $$\sqrt{x^T x} \sim \chi_D,$$

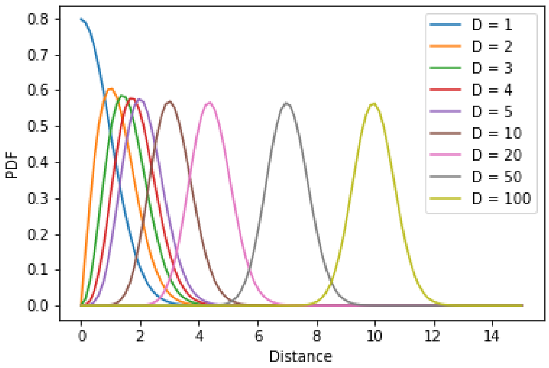
   the multivariate "half"-normal distribution. We can then use the χ distribution's PDF

   $$f(d) = \frac{2^{1-D/2} x^{D-1} e^{-x^2/2}}{\Gamma(D/2)}.$$

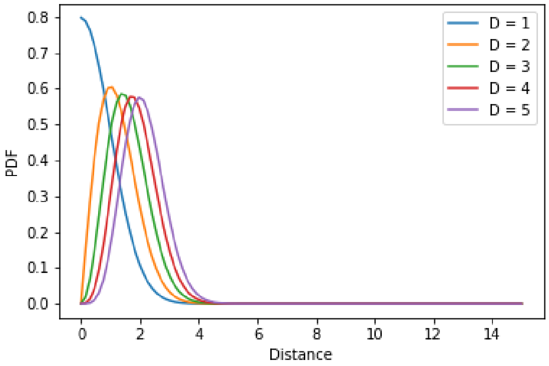
   To compute the mean and variance, we can use the parameters of the χ-distribution

   $$\mathbb{E}[\sqrt{x^T x}] = \sqrt{2}\,\frac{\Gamma((D+1)/2)}{\Gamma(D/2)}, Var(\sqrt{x^T x}) = D^2 - (\mathbb{E}[\sqrt{x^T x}])^2.$$

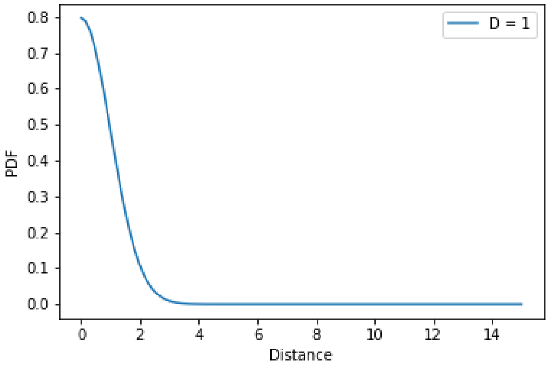
2. See plots.



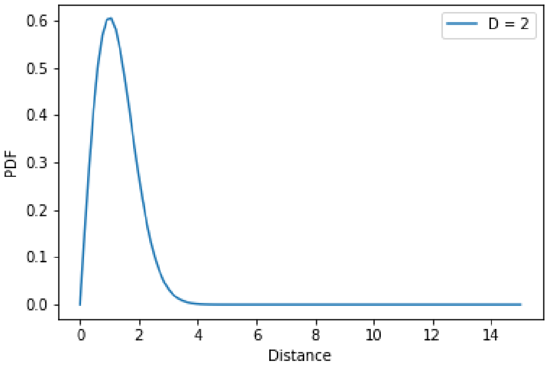
Plot of PDF for several different values of D



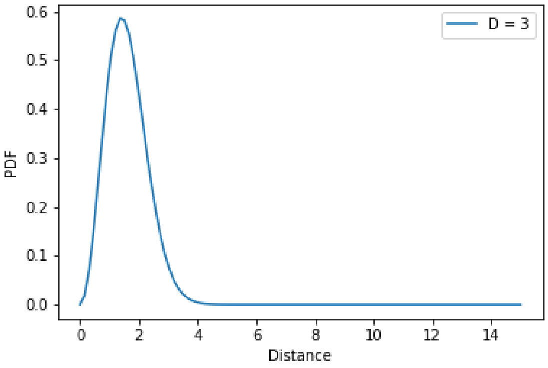
Plot of PDF for $D = 1, 2, 3, 4, 5$



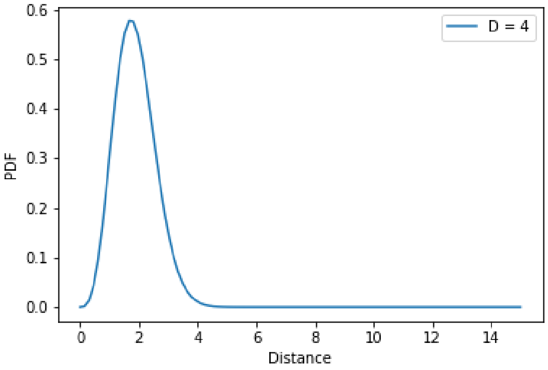
Plot of PDF for $D = 1$



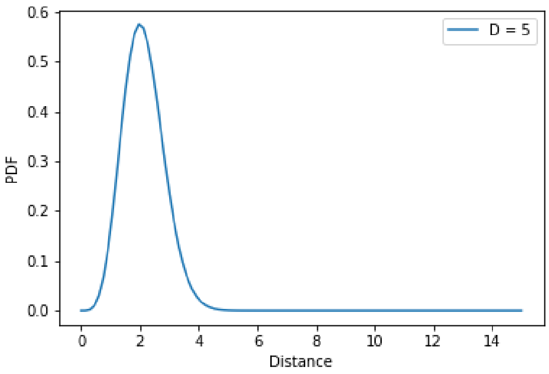
Plot of PDF for $D = 2$



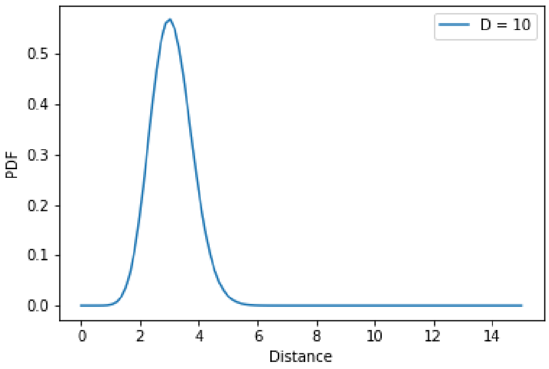
Plot of PDF for $D = 3$



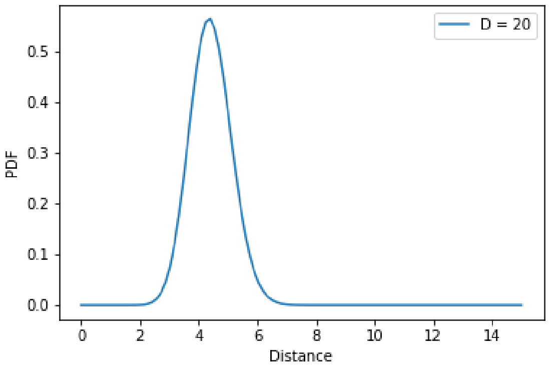
Plot of PDF for $D = 4$



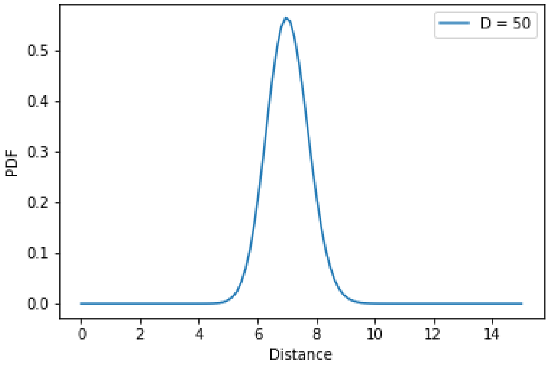
Plot of PDF for $D = 5$



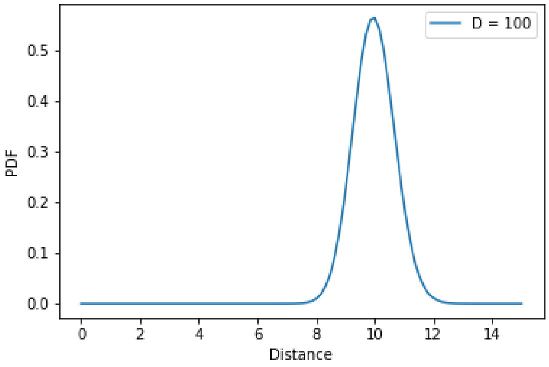
Plot of PDF for $D = 10$



Plot of PDF for $D = 20$

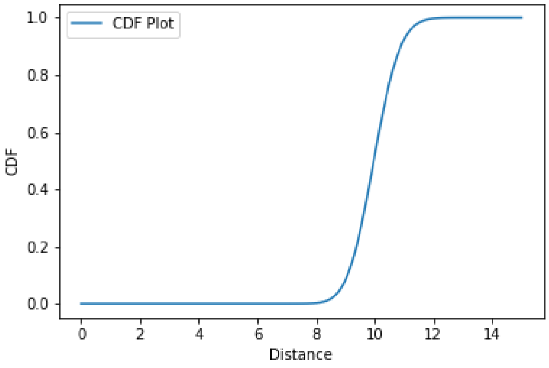


Plot of PDF for $D = 50$



Plot of PDF for $D = 100$

3. See plot.



Plot of CDF for $D = 100$

4. A safe bet seems to be between $d = 9$ and $d = 11$.

Problem 2 (Hurdle Models for Count Data, 10pts)

In this problem we consider predictive models of count data. For instance given information about the student x, can we predict how often they went to the gym that week y? A natural choice is to use a Poisson GLM i.e. y conditioned on x is modeled as a Poisson distribution.

However, in practice, it is common for count data of this form to follow a bi-modal distribution over count data. For instance, our data may come from a survey asking students how often they went to the gym in the past week. Some would do so frequently, some would do it occasionally but not in the past week (a random zero), and a substantial percentage would never do so.

When modeling this count data with generalized linear models, we may observe more zero examples than expected from our model. In the case of a Poisson, the mode of the distribution is the integer part of the mean. A Poisson GLM may therefore be inadequate when means can be relatively large but the mode of the output is 0. Such data is common when many data entries have 0 outputs and many also have much larger outputs, so the mode of output is 0 but the overall mean is not near 0. This problem is known as *zero-inflation.*

This problem considers handling zero-inflation with a two-part model called a *hurdle model.* One part is a binary model such as a logistic model for whether the output is zero or positive. Conditional on a positive output, the "hurdle is crossed" and the second part uses a truncated model that modifies an ordinary distribution by conditioning on a positive output. This model can handle both zero inflation and zero deflation.

Suppose that the first part of the process is governed by probabilities $p(y > 0 \mid x) = \pi$ and $p(y = 0 \mid x) = 1 - \pi$; and the second part depends on $\{y \in \mathbb{Z} \mid y > 0\}$ and follows a probability mass function $f(y \mid \mathbf{x})$ that is truncated-at-zero. The complete distribution is therefore:

$$\begin{aligned} P(y = 0 \mid x) &= 1 - \pi \\ P(y = j \mid x) &= \pi \frac{f(j \mid \mathbf{x})}{1 - f(0 \mid \mathbf{x})}, \ j = 1, 2, ... \end{aligned}$$

One choice of parameterization is to use a logistic regression model for π:

$$\pi = \sigma(\mathbf{x}^\top \mathbf{w}_1)$$

and use a Poisson GLM for f with mean parameters λ (see Murphy 9.3):

$$\lambda = \exp(\mathbf{x}^\top \mathbf{w}_2)$$

(a) Suppose we observe N data samples $\{(x_n, y_n)\}_{n=1}^N$. Write down the log-likelihood for the hurdle model assuming an unspecified mass function f. Give an maximum likelihood estimation approach for the specified parts of the model.

(b) Assume now that we select Poisson distribution for f. Show that the truncated-at-zero Poisson distribution (as used in the hurdle model) is a member of the exponential family. Give its the sufficient statistics, natural parameters and log-partition function.

(c) What is the mean and variance of a truncated Poisson model with mean parameter λ? If we observe n i.i.d. samples from a truncated Poisson distribution, what is the maximum likelihood estimate of λ? (Note: Give an equation which could be solved numerically to obtain the MLE.)

(d) Now assume that we using a hurdle model as a GLM with f as a Poisson distribution. Show that this is a valid GLM (exponential family for y), derive its log-likelihood, and give its sufficient statistics.

Solution

1. Defining $I_{n,0}$ and N_0 as the indicator that $y_n = 0$ and the number of counts of 0 among the ys, respectively, and $I_{n,+}$ and N_+ as the indicator that $y_n > 0$ and the number of counts of nonzero ys, respectively,

$$\mathcal{L}(\pi, \lambda) = \sum_{n=1}^{N} \log(p(D_n \mid \pi, \lambda)) = \sum_{n=1}^{N} \log(\pi^{I_{n,0}}) + \log(((1-\pi)\frac{f(j_n \mid x)}{1 - f(0 \mid x)})^{I_{n,+}})$$

$$= \sum_{n=1}^{N} I_{n,0} \log(\pi) + I_{n,+} \log((1-\pi)\frac{f(j_n \mid x)}{1 - f(0 \mid x)})$$

$$= N_0 \log(\pi) + \sum_{n, j_n > 0} \log\left[(1-\pi)\frac{f(j_n \mid x)}{1 - f(0 \mid x)}\right]$$

The MLE for π can be found

$$\frac{\partial \mathcal{L}}{\partial \pi} = \frac{N_0}{\pi} - \sum_{n, j_n > 0} \frac{1 - f(0 \mid x)}{(1-\pi) f(j \mid x)} = 0$$

If we could condense the sum into equally valued js,

$$\pi_{MLE} = \frac{N_0 f(j \mid x)}{N_0 f(j \mid x) + N_+(1 - f(0 \mid x))}.$$

2.

$$P(y = j \mid x) = \frac{\lambda^j e^{-\lambda}}{j!(1 - e^{-\lambda})}$$

$$= \frac{1}{j!} \frac{\lambda^j e^{-\lambda}}{1 - e^{-\lambda}}$$

$$= \frac{1}{j!} \exp\{\log(\frac{\lambda^j e^{-\lambda}}{1 - e^{-\lambda}})\}$$

$$= \frac{1}{j!} \exp\{j \log(\lambda) - \lambda - \log(1 - e^{-\lambda})\}$$

$$= h(j) \exp\{\theta^T \phi(j) - A(\theta)\},$$

where $h(j) = \frac{1}{j!}$, $\theta = \log(\lambda)$,$\phi(j) = j$, and $A(\theta) = \lambda + \log(1 - e^{-\lambda}) = e^{\theta} + \log(1 - e^{-e^{\theta}})$.

3.

$$\frac{dA(\theta)}{d\theta} = e^{\theta} + \frac{e^{-e^{\theta} + \theta}}{1 - e^{-e^{\theta}}} = \lambda + \frac{\lambda e^{-\lambda}}{1 - e^{-\lambda}} = \frac{\lambda}{1 - e^{-\lambda}} = \mathbb{E}[X]$$

$$\frac{d^2 A(\theta)}{d\theta^2} = \lambda + \frac{e^{-\lambda}(1-\lambda)\lambda}{1 - e^{-\lambda}} - \frac{e^{-2\lambda}\lambda^2}{(1 - e^{-\lambda})^2} = \frac{e^{\lambda}(-1 + e^{\lambda} - \lambda)\lambda}{(-1 + e^{\lambda})^2} = \mathbb{E}[X^2]$$

$$Var(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \frac{e^{\lambda}(-1 + e^{\lambda} - \lambda)\lambda}{(-1 + e^{\lambda})^2} - \frac{\lambda^2 e^{2\lambda}}{(e^{\lambda} - 1)^2} = \frac{\lambda + \lambda^2}{1 - e^{-\lambda}}$$

Using the expression for the MLE of an exponential family,

$$\lambda_{MLE} = \frac{\sum_d j}{N}$$

4. For the hurdle model, when $y = 1$, we can use several similar exponential family components: $h(y) = \frac{1}{y!}$, $\theta = [\log(\lambda), \log(\pi)]$,$\phi(y) = [y, 1]$, $A(\theta) = \lambda + \log(1 - e^{-\lambda}) = e^{\theta} + \log(1 - e^{-e^{\theta}})$. If we consider this exponential family without $A(\theta)$, then we end up with a form of the exponential family that sums to $A(\theta)$ over its support. If we then multiply this function by π, and construct an alternate form in which we multiply $A(\theta)$ by $1 - \pi$ and then divide both by $A(\theta)$, then we have the two components of the hurdle model defined above. Hence, this constitutes a valid GLM. We can derive the log-likelihood from the form given in part (a)

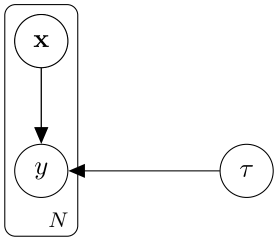
$$\mathcal{L}(\pi, \lambda) = N_0 \log(\pi) + \sum_{n, j_n > 0} \log\big[(1 - \pi)\frac{\lambda^{j_n} e^{-\lambda}}{(1 - e^{-\lambda}) j_n!}\big]$$

$$= N_0 \log(\pi) + \sum_{n, j_n > 0} \log\big[(1 - \pi)\frac{\lambda^{j_n}}{(e^{\lambda} - 1) j_n!}\big].$$

Throughout this problem, I mistakenly treated the probability of $y = 0$ to be π and the probability of $y > 0$ to be $1 - \pi$. However, the mechanics of the problem are otherwise the same.

Problem 3 (Directed Graphical and Naive Bayes, 10pts)

To draw the DGMs for this problem, we recommend using the `tikzbayesnet` *library. For example the following is drawn in LaTeX:*



This problem focuses on modeling a joint distribution of random variables, $p(y, x_1, \dots, x_V)$, consisting of discrete variables. These variables represent a class label $y \in \{1, \dots, C\}$ and features $x_1 \dots, x_V$ each of each can take on a values $x_v \in \{0, 1\}$.

(a) Let $V = 4$. Use the chain rule to select any valid factorization of this joint distribution into univariate distributions. Draw the directed graphical model corresponding to this factorization.

(b) What is the sum of the sizes of the *conditional probability tables* associated with this graphical model. Can you reduce the order of magnitude of this value with a different DGM?

(c) Now consider a naive Bayes factorization of this model, given by,

$$p(y, x_1, \dots, x_V) \approx p(y) \prod_v p(x_v | y).$$

Draw a directed graphical model for this factorization. What is the size of the conditional probability tables required to fully express any factored distribution of this form?

(d) In class, we parameterized naive Bayes such that the class distribution is Categorical with a Dirichlet prior, and the class-conditional distributions are Bernoulli with a Beta prior. Extend the graphical model above to show the generative model of N data points and include the parameters and hyper-parameters as random variables.

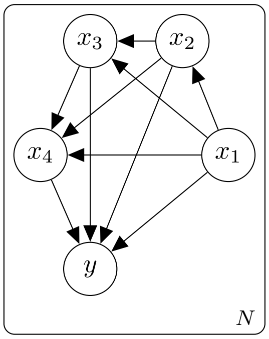
(e) Assuming the data obeys the naive Bayes assumptions, answer the following questions as true/false using your directed graphical model. Justify your answer.

- For a given example, features x_1 and x_2 are independent.
- The class labels y are always conditionally independent of the class-conditional parameters.
- Upon observing the class distribution parameters, the class labels are conditionally independent.
- Upon observing the class distribution parameters, the features are conditionally independent.
- Upon observing the class distribution hyper-parameters, the class labels are conditionally independent.

(f) For the next problem, we will utilize naive Bayes for a problem where each example has a *bag* or multiset of items. A bag is a set that may contain multiple instances of the same value. One approach is to ignore this property and use x_v as an indicator function for each item type. An alternative is to model x_v with sample space $\{0, \dots, D\}$, where D is the maximum times an item appears and to use a Dirichlet-Categorical for the class-conditional. Give one benefit and one drawback of this approach. Propose a third option for modeling this distribution.

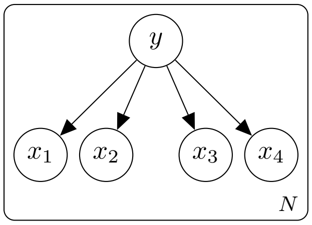
Solution

1. $p(y, x_1, x_2, x_3, x_4) \propto p(y \mid x_1, x_2, x_3, x_4) p(x_4 \mid x_1, x_2, x_3) p(x_3 \mid x_1, x_2) p(x_2 \mid x_1) p(x_1)$

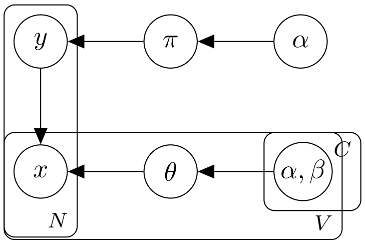


2. The sum of the sizes of the conditional probability tables associated with this model is $16+8+4+2 = 30 * C$, or $31 * C$ if you include the size of the trivial conditional probability table for the marginal $p(x_1)$. It doesn't seem that a different DGM would be applicable unless we were given more information about potential independence relationships/patterns. Otherwise, we could only change the order of information flow to condition on y, but that would seem to only increase the combined size of the conditional probability tables.

3. The size of the conditional probability tables required to express the factored distribution in this case is $2^V * (C-1)$.



4. See below



5.
   - Unconditional on the class y, features x_1 and x_2 are not independent. Given the class y, the two features will be independent.
   - Via d-separation, we can infer that there is independence unconditional of the features. If we condition on the features, then "explaining away" deprives us of the independence.
   - By the same d-separation property as in bullet 1, we know that the class labels are conditionally independent given observation of the class distribution parameters. There is conditional independence.

- The features are not conditionally independent given the class distribution parameters, because observing one feature's value gives us information as to the value of y, which can skew our information about the other features' values.
- Unlike in bullet 3, there is no conditioning on the class distribution parameters, so d-separation still indicates dependence. This is because knowing one class label can inform one about the value of the class distribution parameters and thus change the expected distribution over the class labels.

6. One benefit is this takes into account the important information conveyed by repeated instances of a word. However, the fact that we create a numerical "indicator" of the number of uses of each word implies that we will have v features for any sentence, even if the sentence is small/consists of very few words. Another drawback is that we have to construct a discrete distribution over D possible appearances, where D may represent a large number for a common word like "the." Hence, our feature probability distribution tables will be large. Another way to make this model is to use a zero-inflated Poisson (ZIP; aka the hurdle model in Question 2), so that a word has an inflated probability of appearing zero times in a sentence and a count distribution according to the Poisson distribution conditional on more than zero appearances in the sentence.

Problem 4 (Naive Bayes Implementation, 10pts)

You will now implement a naive Bayes classifier for for sentiment classification. For this problem you will use the IMDB Movie Reviews dataset which consists of positive and negative movie reviews . Here are two example reviews:

> `there is no story! the plot is hopeless! a filmed based on a car with a stuck accelerator, no brakes, and a stuck automatic transmission gear lever cannot be good! ... i feel sorry for the actors ... poor script ... heavily over-dramatized ... this film was nothing but annoying, stay away from it!` [negative review]
>
> `i had forgotten both how imaginative the images were, and how witty the movie ... anyone interested in politics or history will love the movie's offhand references - anyone interested in romance will be moved - this one is superb.` [positive review]

As noted in the last problem, it is common to think of the input data as a bag/multiset. In text applications, sentences are often represented as a *bag-of-words*, containing how many times each word appears in the sentence. For example, consider two sentences:

- `We like programming. We like food.`
- `We like CS281.`

A vocabulary is constructed based on these two sentences:

`["We", "like", "programming", "food", "CS281"]`

Then the two sentences are represented as the number of occurrences of each word in the vocabulary (starting from position 1):

- `[0, 2, 2, 1, 1, 0]`
- `[0, 1, 1, 0, 0, 1]`

We have included a utility file `utils.py` that does this mapping. For these problems you can therefore treat text in this matrix representation.

- Implement a Naive Bayes classifier using a Bernoulli class-conditional with a Beta prior where each feature is an indicator that a word appears at least once in the bag.
- Implement a Naive Bayes classifier using a Categorical class-conditional with a Dirichlet prior. Here the features represent that count of each word in the bag.
- For both models, experiment with various settings for the priors. For the Dirichlet prior on the class, begin with $\boldsymbol{\alpha} = \mathbf{1}$ (Laplace Smoothing). Do the same for the class-conditional prior (be it Dirichlet or Beta). Keeping uniformity, vary the magnitude to .5 and smaller. If the classes are unbalanced in the dataset, does it help to use a larger α for the less-often occuring class? Optionally, choose class-conditional priors based on an outside text source. Validate your choices on the validation set, and report accuracy on the test set.
- (Optional) With the bag-of-words representation, would the model be able to capture phrases like "don't like"? An alternative to the bag-of-words model is known as the bag-of-bigrams model, where a bigram is two consecutive words in a sentence. Modify `utils.py` to include bigram features with either model and see if they increase accuracy.
- (Optional Reading) *Baselines and Bigrams: Simple, Good Sentiment and Topic Classification* `http://www.aclweb.org/anthology/P/P12/P12-2.pdf#page=118`

Solution

- Implemented
- Implemented
- If the classes are unbalanced in the dataset, it could be indicative of a scarcity of certain classes in the general lexicon; however, it could also be a telling sign of a biased dataset. In that case, using larger α values (up to a reasonable bound) may be helpful in terms of mitigating data sparsity. Below is the table for Beta-Binomial Validation and Test:

α, β	Validation Accuracy	Test Accuracy
0	50	50
0.1	48.1	50.1
0.2	47.9	49.9
0.3	47.5	49.7
0.4	47.5	49.7
0.5	47.4	49.6
0.6	47.2	49.8
0.8	47.3	49.7
1.0	47.5	49.7

Below is the table for Dirichlet-Categorical Validation and Test:

α	Validation Accuracy	Test Accuracy
0	86.4	84.1
0.1	86.6	84.5
0.2	86.5	84.7
0.3	86.5	85.5
0.4	86.6	85.5
0.5	86.7	86
0.6	86.8	86.2
0.8	86.9	86.5
1.0	86.8	86.4

The latter table shows that higher priors on α up to $\alpha = 1$ yield higher prediction accuracy on the test data.

Problem 5 (Logistic Regression with Autograd, 15pts)

In the previous problem, you implemented a Naive Bayes classifier for sentiment classification on the IMDB Movie Reviews dataset. In this problem, you will apply logistic regression to the same task.

(a) ℓ_1-regularized logistic regression. Consider a model parameterized by $\mathbf{w}$:

$$\begin{aligned} p(\mathbf{w}) &= \frac{1}{2b}\exp(-\frac{\|\mathbf{w}\|_1}{b}) \\ p(y=1|\mathbf{x},\mathbf{w}) &= \sigma(\mathbf{w}^\top \mathbf{x}) \\ p(y=0|\mathbf{x},\mathbf{w}) &= 1-\sigma(\mathbf{w}^\top \mathbf{x}) \end{aligned}$$

where $\sigma(\cdot)$ is the sigmoid function. Note that we are imposing a Laplacian prior on $\mathbf{w}$, see Murphy, 2.4.4.

(i) Given a dataset $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^N$, derive the necessary gradient updates for MAP of $\mathbf{w}$.[a]

(ii) Show that for some constant λ, MAP inference of $\mathbf{w}$ is equivalent to minimizing

$$-\frac{1}{N}\sum_{i=1}^N \log p(y^{(i)}|\mathbf{x}^{(i)},\mathbf{w}) + \lambda\|\mathbf{w}\|_1$$

(b) Implementation using PyTorch automatic differentiation.[b]

(i) Using the bag-of-words feature representation from the previous question, train a logistic regression model using PyTorch autograd and `torch.nn`. Report test accuracy. Select regularization strength λ based on validation accuracy.

(ii) Which 5 words correspond to the largest weight indices, per class, in the learnt weight vectors? Which 5 words correspond to the least weight indices?

(iii) Study how sparsity (i.e percentage of zero elements in a vector) of the parameter vector changes with different values of λ. Again, tune λ on the validation set and report the test accuracies on the test set. Suggested values to try are $\{0, 0.001, 0.01, 0.1, 1\}$. You can treat parameters with $< 1e-4$ absolute values as zeros.

[a] You only need to consider the case where $\forall i, w_i \neq 0$. If $\exists i, w_i = 0$, we can use its subgradients instead.

[b] `https://github.com/harvard-ml-courses/cs281/blob/master/cs281-f17/sections/04/walkthrough.ipynb`.

Solution

1. (a)

$$p(w \mid D) \propto p(y \mid x, w)p(w)$$

$$p(w \mid D) \propto \big[\prod_{i=1}^N p(y^{(i)} \mid x^{(i)}, w)\big](\frac{1}{2b}\exp\{-\frac{\|\mathbf{w}\|_1}{b}\})$$

$$p(w \mid D) \propto \big[\sum_{i=1}^N \log p(y^{(i)} \mid x^{(i)}, w)\big] + \log\frac{1}{2b} - \frac{\|\mathbf{w}\|_1}{b}$$

$$\frac{\partial p(w \mid D)}{\partial w} = \big[\frac{\partial \log p(y^{(i)} \mid x^{(i)}, w)}{\partial w}\big] - \frac{1}{b}\frac{\partial\|\mathbf{w}\|_1}{\partial w}$$

Then, the gradient updates are

$$[\sum_{i=1}^{N} I\{y^{(i)} = 1\}\frac{\exp\{-w^T x^{(i)}\}}{(1+\exp\{-w^T x^{(i)}\})^2}x^{(i)} - I\{y^{(i)} = 0\}\frac{\exp\{-w^T x^{(i)}\}}{(1+\exp\{-w^T x^{(i)}\})^2}x^{(i)}] \pm \frac{1}{b}.$$

(b)

$$\arg\max_w p(w \mid D) \propto \arg\max_w (p(D \mid w)p(w))$$

Because the log function is monotonically increasing,

$$= \arg\max_w (\log p(D \mid w) + \log p(w))$$

$$= \arg\min_w (-\log p(D \mid w) - \log p(w))$$

$$= \arg\min_w (-\frac{1}{N}\log\sum_{i=1}^{N} p(D^{(i)} \mid w) + \lambda\|\mathbf{w}\|_1)$$

for some λ. Hence, MAP inference in this case is equivalent to minimizing this objective with a regularization term.

2. (a) Regularization strength of $\lambda = 0.05$ was chosen based on validation accuracy. The test accuracy was 84.20%.

   (b) For Bad Movies
   The five largest in decreasing order: bad, worst, nothing, no, waste
   The five smallest in increasing order: great, best, well, love, excellent

   For Good Movies
   The five largest in decreasing order: great, best, well, love, excellent
   The five smallest in increasing order: bad, worst, nothing, no, waste

   (c) The table is below.

λ	$\%0s$	Test Accuracy
0	52.50	52.3
0.001	52.56	71.7
0.01	52.37	81.5
0.05	52.23	84.2
0.1	51.94	81.5
0.2	51.18	67.6
1	50.09	59.6

Problem 6 (Neural Networks, 5pts)

In the previous problem, we have implemented a Logistic Regression classifier using PyTorch. Logistic Regression can be seen as a 1-layer neural network. With PyTorch automatic differentiation, implementing a multi-layer neural network only requires incremental change to our logistic regression implementation.

(a) Implement a multi-layer neural network for IMDB classification and report accuracy on the test set. You are free to design the network structure (number of hidden units, activation function) and choose the optimization methods (SGD or ADAM, regularization or not, etc.).

(b) (Optional) Implement sentiment classification based on Convolutional Neural Networks. We recommend reading Yoon Kim (2014) *Convolution Neural Networks for Sentence Classification*. Note that in this part, you need to treat the text as a sequence of word vectors instead of bag-of-words. You can do this by forwarding `batch.text[0]` to `torch.nn.Embedding(vocab_size, embedding_dim)` after setting the weights using the pretrained vectors from `text_field.vocab.vectors`.

Solution

1. Using the same parameter $\lambda = 0.05$ as set in 5, I constructed a neural network that took in a vocabulary counts vector, applied a linear dimension reduction to $\mathbb{R}^{1000}$, applied the ReLU function, applied another linear dimension reduction to $\mathbb{R}^{100}$, applied the sigmoid function, and performed a final linear dimension reduction to $\mathbb{R}^2$. On the test data, this model yielded 77.2% accuracy. It is interesting how accuracy went down upon addition of layers; perhaps more care is needed when examining optimal layer structure and relationships between layers and their nodes.