Your Name
email@fas.harvard.edu
CS281-F17

Assignment #2
Due: 5:00pm October 9 2017

Collaborators: Jane Doe

Homework 2: Models - Solutions

Grading Instructions

In the solutions, you will see several highlighted checkpoints. These each have a label that corresponds to an entry in the Canvas quiz for this problem set. The highlighted statement should clearly indicate the criteria for being correct on that problem. If you satisfy the criteria for a problem being correct, mark "Yes" on the corresponding position on the Canvas quiz. Otherwise, mark "No". Your homework scores will be verified by course staff at a later date.

That being said, many of the problems in this course will be proofs. If you find a proof that isn't referred to by the course solution, don't worry. If you're uncertain about the proof, make a Piazza post. If you're certain, mark it correct (and we'll look at it during verification).

Introduction

There is a mathematical component and a programming component to this homework. Please submit your PDF and Python files to Canvas, and push all of your work to your GitHub repository. If a question requires you to make any plots, please include those in the writeup.

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:

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

where $\mathbf{0}_D$ indicates a column vector of D zeros and $\mathbf{I}_D$ 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 $\mathcal{N}(\mathbf{0}_D, \mathbf{I}_D)$. (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. Note that $d^2 = x_1^2 + \cdots + x_D^2$ is distributed as the sum of the squares of D independent normal random variables. Each $x_i^2 \propto \exp -\frac{x_i^2}{2}$. A change of variables shows that this is a Gamma distribution with shape parameter $k = 1/2$ and scale parameter $\theta = 2$. The sum of D such Gamma random variables is just a Gamma random variable with $k = D/2$. Therefore, $d^2 \sim Gam(k = D/2, \theta = 2)$. By another transformation of variables we get that $p(d) = 2dGam(d^2; k = D/2, \theta = 2)$.

   Check 1 (4 pts): Recognize that d^2 follows a Gamma distribution for 2 pts. Alternatively, $d^2 \sim Chi^2(D)$ (special case of Gamma) would get you full credit.

   Correctly give the parameters of the d distribution for 2pt.

2. The probability density functions are shown in Figure (a).

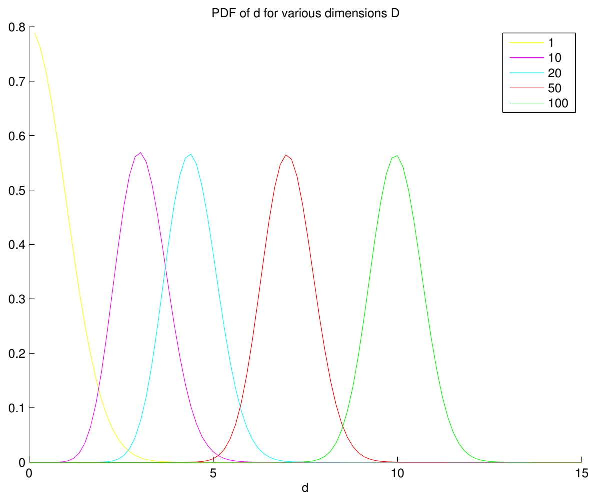
   Check 2 (2 pts): Correctly plot the pdfs for 2 pts.

3. The CDF for D=100 is shown in Figure (b).

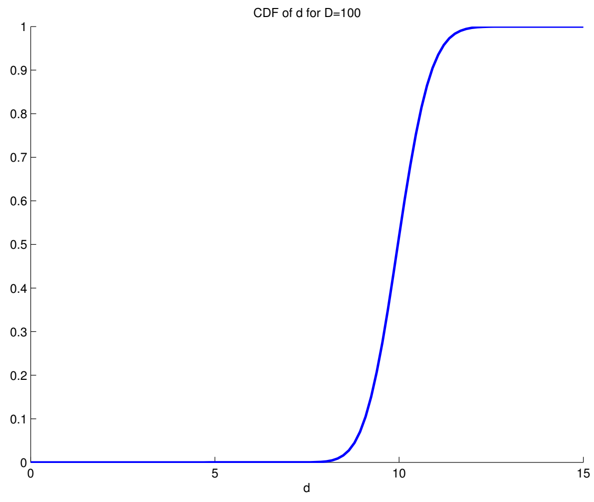
   Check 3 (2pts): Correctly plot the CDF for 2 pts.

4. It looks like the central 90% of the mass lies roughly between $d = 9$ and $d = 11$.

   Check 4 (2 pts): Observe that the mass is mostly between 9 and 11. Anything between 8 and 12 is acceptable.



(a) PDFs of d for various dimensions D.



(b) CDF for d when D=100.

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:

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

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

(a) The joint likelihood is

$$p(y|\mathbf{X}, \mathbf{w}_1, \mathbf{w}_2) \propto \prod_{n=1}^{N} (1-\pi_n)^{I(y_n=0)} [\pi_n \frac{f(y_n|\lambda_n)}{1-f(0|\lambda_n)}]^{1-I(y_n=0)}$$

and the log-likelihood is

$$\begin{aligned} L(\mathbf{w}_1, \mathbf{w}_2) &= L_1(\mathbf{w}_1) + L_2(\mathbf{w}_2) \\ &= \sum_{y_n>0} \mathbf{x}_n^T \mathbf{w}_1 - \sum_{n=1}^{N} \log(1+e^{\mathbf{x}_n^T \mathbf{w}_1}) + \sum_{y_n>0} \{\log f(y_n|e^{\mathbf{x}_n^T \mathbf{w}_2}) - \log(1-f(0|e^{\mathbf{x}_n^T \mathbf{w}_2}))\} \end{aligned}$$

So parameter estimation can be done separately for the zero and positive part.

Check 5: (2pts). Give the correct form of the joint likelihood for 1 pt. If the form is incorrect but you recognize the factorial structure between the binary model and Poisson model, get 0.5 pt instead. Give the correct form of the log-likelihood for 0.5 pt. Write it as a function of the parameters $\mathbf{w_1}, \mathbf{w_2}$ and quickly explain how the parameters can be estimated for 0.5 pt.

(b) For Poisson distribution, the probability mass function is $p(y=j|\lambda) = \frac{\lambda^j}{j!}e^{-\lambda}$, so the probability of getting zero count is $e^{-\lambda}$. Therefore, the probability mass function of a truncated Poisson distribution is $p(y=j|\lambda) = \frac{\lambda^j}{j!}\frac{e^{-\lambda}}{1-e^{-\lambda}}$, $j = 1, 2, ...$, which is still a member of exponential family. To write it in the exponential family form, $p(y=j|\lambda) = \frac{1}{j!}e^{j\log\lambda - (\lambda + \log(1-e^{-\lambda}))}$. Thus, the sufficient statistic is y, natural paramter is $\log\lambda$ and log-partition funciton is $\lambda + \log(1-e^{-\lambda})$.

Check 6: (2pts). Give the correct form for the truncated Poisson distribution for 1 pt. Give all the exponential family forms for 1 pt.

(c) To get the mean and variance of truncated Poisson distribution, one could start with probability mass function or use the property of exponential family. $E(y|\lambda) = \sum_{j>0} jP(y=j|\lambda) = \sum_{j>0} \frac{jPois(j|\lambda)}{1-e^{-\lambda}} = \frac{\lambda}{1-e^{-\lambda}}$ and $Var(y|\lambda) = E(y^2|\lambda) - E(y|\lambda)^2 = \sum_{j>0} \frac{j^2 Pois(j|\lambda)}{1-e^{-\lambda}} - \frac{\lambda^2}{(1-e^{-\lambda})^2} = \frac{\lambda^2+\lambda}{1-e^{-\lambda}} - \frac{\lambda^2}{(1-e^{-\lambda})^2} = \frac{\lambda}{1-e^{-\lambda}} - \frac{\lambda^2 e^{-\lambda}}{(1-e^{-\lambda})^2}$. Alternatively, from log partition function, we could also get:

$$\begin{aligned} &\text{Let } \eta = \log\lambda,\ A(\eta) = e^{\eta} + \log(1-e^{-e^{\eta}}) \\ E(y) &= A'(\eta) = e^{\eta} + \frac{e^{-e^{\eta}}e^{\eta}}{1-e^{-e^{\eta}}} = \lambda + \frac{e^{-\lambda}\lambda}{1-e^{-\lambda}} = \frac{\lambda}{1-e^{-\lambda}} \\ Var(y) &= A''(\eta) = \frac{e^{\eta}}{1-e^{-e^{\eta}}} - \frac{e^{2\eta}e^{-e^{\eta}}}{(1-e^{-e^{\eta}})^2} = \frac{\lambda}{1-e^{-\lambda}} - \frac{\lambda^2 e^{-\lambda}}{(1-e^{-\lambda})^2} \end{aligned}$$

To get MLE $\hat{\lambda}$, one could directly set the derivative of loglikelihood to zero (and also check the second derivative to ensure that it's a maximum). Or use the property of exponential family, that is at MLE, the emipircal mean of the sufficient statistcs equals its expectation:

$$\frac{\hat{\lambda}}{1-e^{-\hat{\lambda}}} = \bar{y} \ ,$$

which could easily solved numerically.

Check 7: (3pts). Correctly compute the mean for 1 pt. Correctly compute the variance for 1pt. For MLE, gives the form given in the solution or give an equation where the derivative is set to 0 suffices for 1pt.

(d) First of all, for a Poisson hurdle model, the output density is in the exponential family, for each sample $(y, \mathbf{x})$, we could write the loglikelihood in the exponential family form:

$$\begin{aligned}
&\mathbb{1}(y>0)\log(\frac{\pi}{1-\pi}) + \mathbb{1}(y>0)\{y\log\lambda - \lambda - \log(1-e^{-\lambda})\} + \log(1-\pi) \\
&= y\log\lambda + \mathbb{1}(y>0)\{\log(\frac{\pi}{1-\pi}) - \lambda - \log(1-e^{-\lambda})\} + \log(1-\pi)
\end{aligned}$$

where sufficient statistics $\phi(y) = (y,\ \mathbb{1}(y>0))$ and canonical paramters $\eta(\theta) = (\log\lambda,\ \log(\frac{\pi}{1-\pi}) - \lambda - \log(1-e^{-\lambda}))$ Then, the model maps the paramters to a linear combination of inputs:

$$\pi = \sigma(\mathbf{x}^T\mathbf{w}_1),\ \lambda = \exp(\mathbf{x}^T\mathbf{w}_2)$$

There is also an invertible mapping between means of sufficent statistics and the paramters, hence a link function could be specified between linear combination of inputs and means.

The joint log likelihood for N samples is

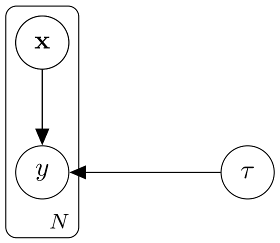
$$\begin{aligned}
L(\mathbf{w}_1, \mathbf{w}_2) &= L_1(\mathbf{w}_1) + L_2(\mathbf{w}_2) \\
&= \sum_{y_n>0} \mathbf{x}_n^T\mathbf{w}_1 - \sum_{n=1}^{N} \log(1+e^{\mathbf{x}_n^T\mathbf{w}_1}) + \sum_{y_n>0}\{y_n\mathbf{x}_n^T\mathbf{w}_2 - e^{\mathbf{x}_n^T\mathbf{w}_2} - \log(1-\exp(-e^{\mathbf{x}_n^T\mathbf{w}_2}))\}
\end{aligned}$$

The sufficient statistics are $\{I\{(y_n)>0\},\ n = 1, 2..., N\}$ and $\{\sum_{y_n>0} y_n\mathbf{x}_n\}$.

Check 8: (3pts). Show that the model is a valid GLM for 1pt. Give the log likelihood for 1 pt, and give the sufficient statistics for 1 pt.

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, $p(y, x_1, \ldots, x_V)$, consisting of discrete variables. These variables represent a class label $y \in \{1, \ldots, C\}$ and features $x_1 \ldots, x_V$ 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,

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

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.

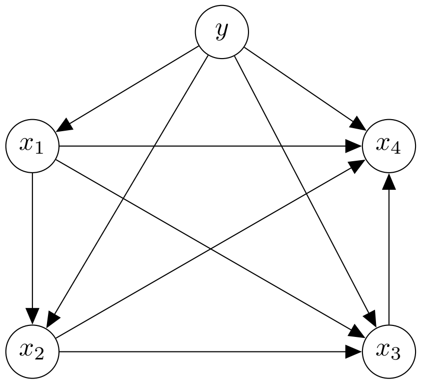
- For a given example, features x_1 and x_2 are independent.
- The class labels y are always conditionally independent of the class-conditional parameters.
- Upon observing the class distribution parameters, the class labels are conditionally independent.
- Upon observing the class distribution parameters, 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 $\{0, \ldots, D\}$, 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

(a) Without knowing any more about the random variables, we can only factor this joint distribution using the chain rule of probability.

$$p(y, x_1, x_2, x_3, x_4) = p(x_4|x_3, x_2, x_1, y)p(x_3|x_2, x_1, y)p(x_2|x_1, y)p(x_1|y)p(y)$$

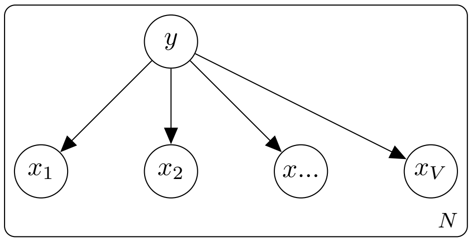


Check 9: (2 pt) Used chain rule for factorization (any ordering of the RV is okay) for 1 point. Drew corresponding directed graphical model for 1 point. Get 0 points for this problem if you fail to note any conditional dependence (arrows) between xs. Get 1 point for the problem if you missed some between-x arrows.

(b) We would have to store $c+2c+2^2c+2^3c+2^4c$ values for this joint distribution. Without any additional information about the dependencies among the variables, this would only be reduced if we factored the joint distribution differently, i.e. $p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)p(x_4|x_3, x_2, x_1)p(y|x_4, x_3, x_2, x_1)$ has table of size $2 + 2^2 + 2^3 + 2^4 + 2^4c$

Check 10: (1 pt) Give the correct table size. Minimal values are okay too, i.e. only need to store $C2^4 - 1$ for $p(y)$, etc.

(c) The DGM representation of a naive Bayes factorization is given by:

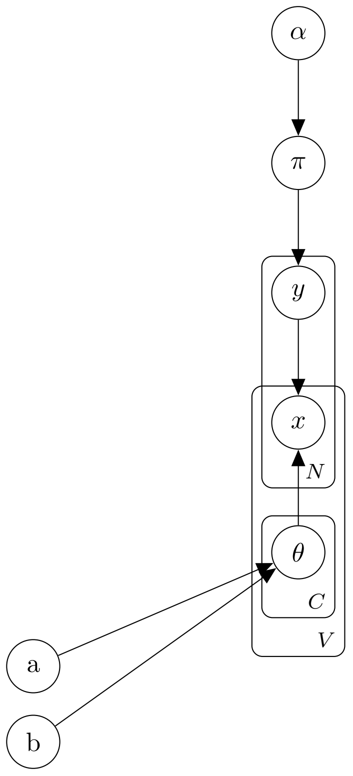


Note how this is constructed. All the conditional probabilities in the model are conditioned on y, hence the arrows from y to $\mathbf{x}$. There are N samples to be sorted, and the lower right N means that this factorization is repeated for each sample.

Now the joint probability distribution can be factored as $p(y, x_1, ..., x_v) = p(y)p(x_1|y)...p(x_v|y)$. Each $p(x_i|y)$ requires storing 2C values. There are v of these binary variables, so we need to store $C + 2Cv$ values total.

Check 11 (1 pt): Draw correct DGM (it is okay if you don't have the plate with N). Give correct value for table size (minimal version is okay too).

(d) Extending the DGM, we have



Check 12 (1.5 pts): Correct graph:
0.5 pts for correct placement of class parameters and hyperparameters
0.5 pts for correct placement of class conditional parameters
0.5 pts for plating
If you did not plate everything like above, that is okay (i.e leaving the V nodes of x unrolled, as in the previous figure.)

(e)
- F, CI given class label
- F, independent so long as x_i is not observed.
- T, the class labels are CI given that the parents are observed.
- F There are still 'active paths' between each feature through y.
- F There are still 'active paths' between each class label through the class distribution parameters.

Check 13 (3 pts): 0.5 points per bullet point (2.5), and 0.5 points for providing short explanations for all.

(f) **Check 14**: (1.5 pts) 0.5 point each for listing a benefit, drawback, and a reasonable alternative approach. For instance, a benefit of the indicator function approach is that it overlooks outliers (people that use one word excessively in one review). One drawback is that it does not consider how much each word is used in the reviews.

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

- **Check 15** (4 pts) Implement with Bernoulli class-conditional with Beta prior

  Let C by the number of classes, indexed by k. Let V be the size of the vocabulary, indexed by j. Let N be the number of reviews. Let N_{jk} be the number of words in class k that had word j present. Let N_k be the number of reviews of class k. Let $\text{Beta}(\alpha, \beta) = \text{Beta}(\beta_1, \beta_0)$. Let the Bernoulli class conditional distribution have a Beta prior parameterized by (β_1, β_0). Let the categorical class distribution have a Dirichlet prior parameterized by $(\alpha_1, ..., \alpha_C)$.

  You used an estimate for the class conditional distribution of the form

  $$\theta_{kj} = \frac{N_{jk} + \beta_1}{N_k + \beta_0 + \beta_1}$$

  by using the posterior mean, or

  $$\theta_{kj} = \frac{N_{jk} + \beta_1 - 1}{N_k + \beta_0 + \beta_1 - 2}$$

  by using the posterior mode.

  You used an estimate for the class distribution of the form

  $$\pi_k = \frac{N_k + \alpha_k}{N + \sum_{k'} \alpha_{k'}}$$

  by using the posterior mean, or

  $$\pi_k = \frac{N_k + \alpha_k - 1}{N + \sum_{k'} \alpha_{k'} - C}$$

  by using the posterior mode.

- **Check 16** (4 pts) Implement with Categorical class-conditional with Dirichlet prior

  Here, let θ_{kj} be a catrogrical distribution over counts for class k and word j. Let Q be the the maximum number of times any word appears in any review of any class. Let counts be indexed by m. Let N_{kjm} be the number of reviews of class k where word j appeared m times. Let the categorical class conditional have a Dirichlet prior parameterized by $(\gamma_1, ..., \gamma_Q)$. You used an estimate for the class conditional distribution of the form

  $$\theta_{kjm} = \frac{N_{kjm} + \gamma_m}{N_k + \sum_{m'} \gamma_{m'}}$$

  by using the posterior mean, or

  $$\theta_{kjm} = \frac{N_{kjm} + \gamma_m - 1}{N_k + \sum_{m'} \gamma_{m'} - Q}$$

  by using the posterior mode.

- **Check 17** (2 pts) Explored various settings for the priors.

  You probably found that using priors of magnitude 1 and below worked better than anything higher. It may have thrown off your data counts to use priors above 1. The larger magnitude you use with a uniform prior, the more "washed away" your count distributions get. Using a non-uniform prior on the class distribution would be useful if you expected to find a non-unform distribution over classes in the real world (test set) and your training data had an even proportion of classes (which it did).

Accuracies on test set should be around 87%, for both Bernoulli and Categorical Naive Bayes.

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

(a) (i) To find the necessary gradient updates:

$$\begin{aligned} p(\mathbf{w}|y,\mathbf{x}) &\propto p(y|\mathbf{x},\mathbf{w})p(\mathbf{w}) \\ \log p(\mathbf{w}|y,\mathbf{x}) &= \log p(\mathbf{w}) + \log p(y|\mathbf{x},\mathbf{w}) + C \\ &= -\frac{\|\mathbf{w}\|_1}{b} + \sum_{i=1}^{N} y^{(i)}\log\sigma(\mathbf{w}^\top\mathbf{x}^{(i)}) + (1-y^{(i)})\log(1-\sigma(\mathbf{w}^\top\mathbf{x}^{(i)})) + C \\ \frac{\partial\log p(\mathbf{w}\,|\,y,\mathbf{x})}{\partial\mathbf{w}} &= -\frac{1}{b}\partial\|\mathbf{w}\|_1 + \sum_{i=1}^{N} y^{(i)}\frac{\sigma(\mathbf{w}^\top\mathbf{x}^{(i)})(1-\sigma(\mathbf{w}^\top\mathbf{x}^{(i)}))}{\sigma(\mathbf{w}^\top\mathbf{x}^{(i)})}\mathbf{x}^{(i)} - (1-y^{(i)})\frac{\sigma(\mathbf{w}^\top\mathbf{x}^{(i)})(1-\sigma(\mathbf{w}^\top\mathbf{x}^{(i)}))}{1-\sigma(\mathbf{w}^\top\mathbf{x}^{(i)})}\mathbf{x}^{(i)} \end{aligned}$$

Where $\partial\|\mathbf{w}\|_1$ is the subgradient of $\|\mathbf{w}\|_1$[1], which is generally a set. In this problem set we only consider the case where $\forall i\ (w_i \neq 0)$, so $\partial\|\mathbf{w}\|_1$ is a single vector $\text{sign}(\mathbf{w})$, and the gradient is:

$$= -\frac{1}{b}\text{sign}(\mathbf{w}) + \sum_{i=1}^{N} y^{(i)}(1-\sigma(\mathbf{w}^\top\mathbf{x}^{(i)}))\mathbf{x}^{(i)} - (1-y^{(i)})\sigma(\mathbf{w}^\top\mathbf{x}^{(i)})\mathbf{x}^{(i)}$$
$$= -\frac{1}{b}\text{sign}(\mathbf{w}) + \sum_{i=1}^{N}(y^{(i)} - \sigma(\mathbf{w}^\top\mathbf{x}^{(i)}))\mathbf{x}^{(i)}$$

Check 18 (1 pt) Derived the correct gradient updates for MAP

(ii)

$$\begin{aligned}
\mathbf{w}_{\text{MAP}} &= \arg\max \log p(\mathbf{w} \mid y, \mathbf{x}) \\
&= \arg\max \log p(y \mid \mathbf{w}, \mathbf{x}) + \log p(\mathbf{w}) \\
&= \arg\max \sum_{i=1}^{N} \log p(y^{(i)} \mid \mathbf{x}^{(i)}, \mathbf{w}) + \log p(\mathbf{w}) \\
&= \arg\min -\frac{1}{N}\sum_{i=1}^{N} \log p(y^{(i)} \mid \mathbf{x}^{(i)}, \mathbf{w}) - \log\frac{1}{2b} + \frac{\|\mathbf{w}\|_1}{b} \\
&= \arg\min -\frac{1}{N}\sum_{i=1}^{N} \log p(y^{(i)} \mid \mathbf{x}^{(i)}, \mathbf{w}) + \lambda\|\mathbf{w}\|_1, \quad \lambda = \frac{1}{b}
\end{aligned}$$

Check 19 (1 pt) Show that for some λ, MAP of w is equivalent to minimizing the above loss.

(b) Implementation

(i) **Check 20** (5 pts) Implemented LR model and get at least 80% test accuracy.

The provided code can get 88.55% test accuracy.

(ii) **Check 21** (1 pts) Give reasonable positive words and negative words.

For example, positive: great, excellent, loved, facorite, wonderful and negative: worst, waste, awful, boring.

(iii) **Check 22** (2 pts) Try various values of λ. Sparsity increases with λ.

For example: $\lambda = 0$, sparsity is 0.001; $\lambda = 0.001$, sparsity is 0.93; $\lambda = 0.01$, sparsity is 0.997; $\lambda = 0.1$, sparsity is 0.99; $\lambda = 1.0$, sparsity is 0.99.

[1] `https://see.stanford.edu/materials/lsocoee364b/01-subgradients_notes.pdf`. page 5

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

(a) **Check 23** (5 pts) Implemented multi-layer neural network with at least 80% test accuracy.

The provided code can get 89.10% test accuracy.