CS5785 Homework 2

This homework is due on **Tuesday, October 5th, 2021 at 11:59PM EST**. The homework is split into programming exercises and written exercises. Upload your homework to Gradescope. Your submission will have two parts:

1. A write-up as a single `.pdf` file. Submit this under the `hw2-report` assignment in Gradescope.

2. Source code and data files for all of your experiments (AND figures) in `.py` files if you use Python or `.ipynb` files if you use Jupyter Notebooks. If you use some other language, include all build scripts necessary to build and run your project along with instructions on how to compile and run your code.

   These files should be made available via a **private** GitHub[1] repository (that you share with your teammate). Place all the relevant code files for this assignment in a `hw2` folder in your repository and use the GitHub submission method under the `hw2-code` assignment in Gradescope. **Note:** Gradescope will take a snapshot of your GitHub repository during submission. Any changes to your repository after the assignment is submitted **will not** be reflected in your submission.

The write-up should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, homework number, and team member names. You are responsible for submitting clear, organized answers to the questions. You could use online LaTeX templates from Overleaf, under "Homework Assignment" and and "Project / Lab Report".
Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. Please pay attention to Canvas for announcements, policy changes, etc. and Slack for homework related questions. You are encouraged (but not required) to work in groups of 2.

If you need help

There are several strategies available to you.

- If you get stuck, we encourage you to post a question on the *#homework* channel on Slack[2]. That way, your solutions will be available to the other students in the class.
- The professor and TAs will offer office hours, which are a great way to get some one-on-one help.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. in this assignment. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

[2] https://join.slack.com/t/cs5785-2021fa-kallus/signup

PROGRAMMING EXERCISES

1. **Regularized linear regression.**

   (a) Join the House Prices: Advanced Regression Techniques competition on Kaggle. Download the training and test data. The competition page describes how these files are formatted.

   (b) Tell us about the data. How many samples are there in the training set? How many features? Which features are categorical?

   (c) What variables seem to be important? Which seem to correlate with the sale price? Plot the relationship between sale price and year of sale, garage area, lot area, and other variables of your choice. Choose 7 variables and, along with the response variable, make a scatterplot matrix (hint: look at pandas.plotting.scatter_matrix or seaborn.pairplot). Explain what you see.

   (d) (*optional*) Using statsmodels, run ordinary least squares on all the features and report which features have a 95% confidence interval that contains 0 and which do not. Comment on what this means.

   (e) Split the training data into a training (80%) and test set (20%). Try to run a variety of regression methods using sklearn methods:

   - Ordinary least squares
   - k-Nearest Neighbors with 10-fold cross validation to choose k
   - Ridge regression with 10-fold cross validation to chose λ
   - LASSO with 10-fold cross validation to chose λ
   - Backward stepwise (linear) regression with 10-fold cross validation to choose k (number of features)
   - Forward stepwise (linear) regression with 10-fold cross validation to choose k (number of features)

   Since each feature in this dataset is measured in completely different units and dimensions (e.g., LotArea vs LotFrontage), make sure that for regularized models (ridge and LASSO) you standardize your data first so that coefficients of smaller units are not unfairly penalized relative to coefficients of bigger units (hint: use sklearn.preprocessing.StandardScaler).
   (Extra hint: sklearn also has methods called RidgeCV and LassoCV.)

   For each, give a brief description of how well it works and why that might be and report the test accuracy (note: this refers to split off test set above; not the Kaggle test set).

   (f) Repeat the above, except for OLS, after adding all the quadratic features: $X_{ij}X_{ik}$ for all $j,k = 1,\ldots,p$ (this includes X_{ij}^2). (Backward/forward stepwise regression are *optional*).

   (g) Which variables are being retained by LASSO and the stepwise regression models and which are regularized away? Do these variables match your intuitions about which variables are important and which are not? Compare this to (d).

   (h) Train your best-performing classifier with all of the training data, and generate test labels on the Kaggle test set. Submit your results to Kaggle and report the accuracy. Is it higher or lower than the cross validation accuracy? Why might that happen? Is it higher or lower than the held-out validation accuracy? Why might that happen?

2. **Sentiment analysis of online reviews.**

   In this assignment you will use several machine learning techniques from the class to identify and extract subjective information in online reviews. Specifically, the task for this assignment is **sentiment analysis**. According to *Wikipedia*, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall contextual polarity of a document. It has been shown that people's attitudes are largely manifested in the language they adopt. This assignment will walk you through the mystery and help you better understand our posts online!

   Important: Use your own implementations for this question. Any existing bag-of-words or naive Bayes implementations are **NOT** allowed.

   (a) Download Sentiment Labelled Sentences Data Set. There are three data files under the root folder. *yelp_labelled.txt*, *amazon_cells_labelled.txt* and *imdb_labelled.txt*. Parse each file with the specifications in *readme.txt*. Are the labels balanced? If not, what's the ratio between the two labels? Explain how you process these files.

   (b) **Pick your preprocessing strategy.** Since these sentences are online reviews, they may contain significant amounts of noise and garbage. You **may or may not** want to do one or all of the following. Explain the reasons for each of your decision (**why or why not**).

   - Lowercase all of the words.
   - Lemmatization of all the words (i.e., convert every word to its root so that all of "running," "run," and "runs" are converted to "run" and and all of "good," "well," "better," and "best" are converted to "good"; this is easily done using nltk.stem).
   - Strip punctuation.
   - Strip the stop words, e.g., "the", "and", "or".
   - Something else? Tell us about it.

   (c) **Split training and testing set.** In this assignment, for each file, please use the first 400 instances for each label as the *training set* and the remaining 100 instances as *testing set*. In total, there are 2400 reviews for training and 600 reviews for testing.

   (d) **Bag of Words model.** Extract features and then represent each review using bag of words model, i.e., every word in the review becomes its own element in a feature vector. In order to do this, first, make one pass through all the reviews in the *training set* (**Explain why** we can't use *testing set* at this point) and build a dictionary of unique words. Then, make another pass through the review in both the *training set* and *testing set* and count up the occurrences of each word in your dictionary. The ith element of a review's feature vector is the number of occurrences of the ith dictionary word in the review. Implement the bag of words model and report feature vectors of any two reviews in the training set.

   (e) **Pick your postprocessing strategy.** Since the vast majority of English words will not appear in most of the reviews, most of the feature vector elements will be 0. This suggests that we need a postprocessing or normalization strategy that combats the huge variance of the elements in the feature vector. You may want to use one of the following strategies. Whatever choices you make, explain **why** you made the decision.

   - *log-normalization.* For each element of the feature vector x, transform it into $f(x) = log(x+1)$.
   - *l1 normalization.* Normalize the $l1$ norm of the feature vector, $\hat{\mathbf{x}} = \frac{\mathbf{x}}{|\mathbf{x}|}$.

- *l*2 *normalization.* Normalize the *l*2 norm of the feature vector, $\hat{\mathbf{x}} = \frac{\mathbf{x}}{\|\mathbf{x}\|}$.
- *Standardize* the data by subtracting the mean and dividing by the variance.

(f) **Sentiment prediction.** Train a naive Bayes model on the training set and test on the testing set. Report the classification accuracy and confusion matrix.

(g) **Logistic regression.** Now repeat using logistic regression classification, and compare performance (*you can use existing packages here*). Try using both L2 (ridge) regularization and L1 (lasso) regularization and report how these affect the classification accuracy and the coefficient vectors (hint: sklearn has a method called LogisticRegressionCV; also note that sklearn doesn't actually have an implementation of *unregularized* logistic regression). Inspecting the coefficient vectors, what are the words that play the most important roles in deciding the sentiment of the reviews?

(h) **N-gram model.** Similar to the bag of words model, but now you build up a dictionary of n-grams, which are contiguous sequences of words. For example, "Alice fell down the rabbit hole" would then map to the 2-grams sequence: ["Alice fell", "fell down", "down the", "the rabbit", "rabbit hole"], and all five of those symbols would be members of the n-gram dictionary. Try $n = 2$, repeat (d)-(g) and report your results.

(i) **Algorithms comparison and analysis.** According to the above results, compare the performances of *naive Bayes*, *logistic regression*, *naive Bayes with 2-grams*, and *logistic regression with 2-grams*. Which method performs best in the prediction task and why? What do you learn about the language that people use in online reviews (e.g., expressions that will make the posts positive/negative)? *Hint:* Inspect the weights learned from logistic regression.

Written Exercises

1. **Ridge Regression.** Recall that ridge regression is the solution to:

$$\min_{\beta \in \mathbb{R}^{1+p}} \sum_{i=1}^{n} (Y_i - \beta^T X_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

where β_0 is the intercept and is *excluded* from the penalty term.

In this excercise we will show that the ridge regression solution can be obtained by ordinary least squares on an augmented data set. Let us augment the $\mathbf{X}$ matrix and $\mathbf{Y}$ vector with p additional rows, so that:

$$\mathbf{X}_{\text{aug}} = \begin{bmatrix} 1 & X_{11} & X_{12} & \dots & X_{1p} \\ 1 & X_{21} & X_{22} & \dots & X_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & \dots & X_{np} \\ 0 & \sqrt{\lambda} & 0 & \dots & 0 \\ 0 & 0 & \sqrt{\lambda} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \sqrt{\lambda} \end{bmatrix} \quad \mathbf{Y}_{\text{aug}} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{1}$$

Show that OLS on $\mathbf{X}_{\text{aug}}$, $\mathbf{Y}_{\text{aug}}$ is the same as ridge on $\mathbf{X}$, $\mathbf{Y}$ with penalty parameter λ. Comment briefly on how we can think of ridge regression as including more prior information about the value of beta as reflected by adding new data points not present in the observed dataset.

2. **Naive Bayes classifiers.** In a medical study, 100 patients all fell into one of three classes: Pneumonia, Flu, or Healthy. The following database indicates how many patients in each class had fever and headache.

Pneumonia

Fever	Headache	Count
T	T	5
T	F	0
F	T	4
F	F	1
	Total:	10

Flu

Fever	Headache	Count
T	T	9
T	F	6
F	T	3
F	F	2
	Total:	20

Healthy

Fever	Headache	Count
T	T	2
T	F	3
F	T	7
F	F	58
	Total:	70

Consider a patient with fever and no headache.

(a) Assuming that the above counts represent the *whole* population, what *probability* would a Bayes optimal classifier assign to each of the three propositions that the patient has Pneumonia, Flu, or neither? Show your work. (For this question, the three values should sum to 1.)

(b) What probability would a naïve Bayes classifier assign to each of the three propositions that the patient has Pneumonia, Flu, or neither? Show your work. (For this question, the three values should sum to 1.)

3. **Naive Bayes for data with nominal attributes.** Given the training data in the table below (Buy Computer data), predict the class of the following new example using Naive Bayes classification: age≤30, income=medium, student=yes, credit-rating=fair. Please show your work.

ID	age	income	student	credit-rating	Class: buys-computer
1	≤30	high	no	fair	no
2	≤30	high	no	excellent	no
3	31…40	high	no	fair	yes
4	>40	medium	no	fair	yes
5	>40	low	yes	fair	yes
6	>40	low	yes	excellent	no
7	31…40	low	yes	excellent	yes
8	≤30	medium	no	fair	no
9	≤30	low	yes	fair	yes
10	>40	medium	yes	fair	yes
11	≤30	medium	yes	excellent	yes
12	31…40	medium	no	excellent	yes
13	31…40	high	yes	fair	yes
14	>40	medium	no	excellent	no