10-601: Homework 2

Due: 25 September 2014 11:59pm (Autolab)

TAs: Siddhartha Jain, Ying Yang

Name: ARAVIND SELVAN

Andrew ID: ASELVAN

Please answer to the point, and do not spend time/space giving irrelevant details. Please state any additional assumptions you make while answering the questions. For Questions 1 to 5, 6(b) and 6(c), you need to submit your answers in a single PDF file on autolab, either a scanned handwritten version or a LaTeXpdf file. Please make sure you write legibly for grading. For Question 6(a), submit your m-files on autolab.

You can work in groups. However, no written notes can be shared, or taken during group discussions. You may ask clarifying questions on Piazza. However, under no circumstances should you reveal any part of the answer publicly on Piazza or any other public website. The intention of this policy is to facilitate learning, not circumvent it. Any incidents of plagiarism will be handled in accordance with CMU's Policy on Academic Integrity.

⋆: Code of Conduct Declaration

- Did you receive any help whatsoever from anyone in solving this assignment? Yes / [✓ No.]
- If you answered *yes*, give full details: ____________ (e.g. *Jane explained to me what is asked in Question 3.4*)
- Did you give any help whatsoever to anyone in solving this assignment? Yes / [✓ No.]
- If you answered *yes*, give full details: ____________ (e.g. *I pointed Joe to section 2.3 to help him with Question 2*).

1: A probabilistic view of linear regression. (TA:- Ying Yang)

Let X and Y be two random variables, β be a constant, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ be a Gaussian random variable with zero mean and variance σ^2. We assume $Y = \beta X + \epsilon$, and that ϵ is independent of X.

(a) Show that given $X = x$, the distribution of Y is $\mathcal{N}(\beta x, \sigma^2)$

[3 points]

$P(Y) = P(\beta X + \epsilon)$; $P(\epsilon = e) = P(y - \beta x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}} \left(\frac{y - \beta x}{2\sigma^2}\right)^2$.

(Since, $(y - \beta x)^2$ is same as $(\beta x - y)^2$) ∴ it can be written as, $\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}} \frac{(\beta x - y)^2}{2\sigma^2}$. $\sim N(\beta x, \sigma^2)$ //.

(b) Let $\{(X_i, Y_i), i = 1, \cdots, n\}$ be n independent samples from the model above. Show that the maximum likelihood estimation of β, where the likelihood is with regard to the conditional distribution $Y|X$, is the least square solution

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} (Y_i - \beta X_i)^2$$

[3 points]

$$P(Y_i | x_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{y_i - \beta x_i}{\sigma}\right)^2}$$

$$\hat{\beta} = \arg\max_{\beta} \prod_{i=1}^{n} e^{-\frac{1}{2}\left(\frac{y_i - \beta x_i}{\sigma}\right)^2}$$

$$\hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} \log c - \frac{1}{2} \frac{(y_i - \beta x_i)^2}{\sigma^2}$$

values $\log c$, $\frac{1}{2}$, σ^2 can be considered as constant here,

$$\therefore \hat{\beta} = \arg\max_{\beta} \sum_{i=1}^{n} -(y_i - \beta x_i)^2$$

changing the sign.

$$\boxed{\hat{\beta} \Rightarrow \arg\min_{\beta} \sum_{i=1}^{n} (y_i - \beta x_i)^2}$$

2: One-dimensional ridge regression(TA:- Ying Yang)

Let Y and X be two random variables, and $Y = \beta X + \epsilon$ given X, where β is a constant, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$, independent of X. Given n independent sample pairs, $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$, instead of ordinary least square, here we estimate β with "ridge regression", by solving the following problem.

$$\hat{\beta} = \arg\min_{\beta} \frac{1}{2}\left(\sum_{i=1}^{n}(y_i - \beta x_i)^2 + \lambda\beta^2\right)$$

where $\lambda \geq 0$ is a tuning parameter.

(a) Give a solution in explicit formula for $\hat{\beta}$. i.e. Compute $\hat{\beta}$ using only the training data and λ.

$f(\beta) = \frac{1}{2}\left(\sum_{i=1}^{n}(y_i - \beta x_i)^2 + \lambda\beta^2\right) \Rightarrow$ Differentiating and equating to 0 *[3 points]*

$$\hat{\beta} = \frac{1}{2}\left(\sum_{i=1}^{n} -2x_i(y_i - \beta x_i) + 2\beta\lambda\right) = 0$$

$$\hat{\beta} = \sum_{i=1}^{n} -x_i y_i + \beta\sum_{i=1}^{n} x_i^2 + \beta\lambda) = 0.$$

$$\boxed{\hat{\beta} = \sum_{i=1}^{n} x_i y_i \Big/ \left(\lambda + \sum_{i=1}^{n} x_i^2\right)}$$

(b) When λ goes from 0 to infinity, how does $\hat{\beta}$ change? Give a brief explanation of your answer.

[2 points]

When $\lambda = 0. \Rightarrow \boxed{\hat{\beta} = \sum_{i=1}^{n} \frac{x_i y_i}{x_i^2}}$; The value $\hat{\beta}$ depends upon (x_i).

when $\lambda = \infty \Rightarrow \boxed{\hat{\beta} = 0}$

3: Least square (TA:- Ying Yang)

Suppose X and Y are random variables. Let $(x_1, y_1), \cdots, (x_n, y_n)$ be n pairs of independent samples. Compute the least square solutions for the following models. $\epsilon \sim N(0, \sigma^2)$

1. $Y = \beta X + \epsilon$
2. $Y = \beta^2 X + \epsilon$

Which of the models above yields to a lower training error? (Hint: The answer may depend on the training samples. If so, please explain in what case one is better than the other.)

[*5 points*]

for $Y = \beta X + \epsilon$.

$\sum_{i=1}^{n} (y_i - \beta x_i)^2 \Rightarrow$ Differentiating and equating to 0.

$$\partial/\partial\beta \left(\sum_{i=1}^{n} (y_i - \beta x_i)^2\right) = 0.$$

$$\Rightarrow 2\sum_{i=1}^{n} -x_i(y_i - \beta x_i) = 0 \; ; \; 2\sum_{i=1}^{n} -x_i y_i + x_i(\beta x_i) = 0 \Rightarrow \boxed{\beta = \sum_{i=1}^{n} \frac{x_i y_i}{x_i^2}}$$

for $Y = \beta^2 X + \epsilon$:

$\sum_{i=1}^{n} (y_i - \beta^2 x_i)^2 \Rightarrow$ Differentiating and equating to 0.

$$\partial/\partial\beta \left(\sum_{i=1}^{n} (y_i - \beta^2 x_i)^2\right) = 0 \Rightarrow 2\sum_{i=1}^{n} -x_i y_i + x_i(\beta^2 x_i) = 0$$

$$\Rightarrow 2\sum_{i=1}^{n} -x_i y_i = \sum_{i=1}^{n} -x_i \beta^2 \Rightarrow \boxed{\beta^2 = \sum_{i=1}^{n} \frac{x_i y_i}{x_i^2}}$$

The first model ($Y = \beta X - \epsilon$) has lower training error because for the negative values the second model won't fit the data correctly but the first would do it correctly.

4: Behavior of linear regression (TA:- Siddhartha Jain)

Suppose you know the number of keyboard and mice sold at various locations around the world and from that you want to estimate the number of computers sold using linear regression. Your model is $Y = \beta_1 k + \beta_2 m$ where Y is the number of computers sold, k is the number of keyboards sold and m is the number of mice sold. You get 101 observations such that 100 of them have 1 keyboard, 1 mouse and 1 computer, but the 101st has 1 keyboard, 0 mouse, and 1 computer.

For **(a)** and **(b)**, you can use `regress` in Matlab to compute the answers.

(a) What are the optimal values of β_1, β_2 in the scenario above.

[3 points]

The optimal values of

$\beta_1 = 1$

$\beta_2 = 0.$

(b) Now suppose you get two additional observations, both with 0 keyboard, 1 mouse, and 1 computer. What are the optimal β values now?

[3 points]

The optimal values of

$\beta_1 = 0.3377.$

$\beta_2 = 0.6689.$

(c) As you should notice, the optimal values for β fluctuate wildly with the addition of even very few observations. This is a problem as then it's hard to converge on a set of values for β. Why is this behavior happening? Given an arbitrary dataset X, Y, how can we test whether such behavior might occur?

[3 points]

The model is not completely robust to outliers as it has high variance and this is why such behaviour is happening. One way to test such behaviour is to plot the data and verify from the graph whether are there any outliers.

5: Gaussian Naive Bayes. (TA:- Ying Yang)

Let $Y \in \{0, 1\}$ be class labels, and let $X \in \mathbb{R}^p$ denote a p-dimensional feature.

(a) In a Gaussian naive Bayes model, where the conditional distribution of each feature is a one-dimensional Gaussian. Given n independent training data points, $\{(X_1, Y_1), \cdots, (X_n, Y_n)\}$, give a maximum-likelihood estimate (MLE) of the conditional distribution of feature $X^{(j)}, j = 1, \cdots, p$, $(X^{(j)}|Y \sim N(\mu_Y^{(j)}, (\sigma_Y^{(j)})^2))$.

[4 points]

$\mu_Y = 1/n \sum_{i=1}^{n} x_i$;

$\sigma_Y = 1/n \sum_{i=1}^{n} (x_i - \bar{\mu})^2$ //

(b) In a full Gaussian Bayes model, we assume that the conditional distribution $\Pr(X|Y)$ is a multidimensional Gaussian, $X|Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$, where μ is the mean vector and $\Sigma \in \mathbb{R}^{p \times p}$ is the covariance matrix. Suppose the prior of Y is already given. How many parameters do you need to estimate in Gaussian naive Bayes model? How many in a full Gaussian Bayes model?

[3 points]

Number of parameters needed for,

* Gaussian naive Bayes model = $4p$.

* Full Gaussian Bayes model = $2 \times \left[P + \left(\frac{P(P+1)}{2}\right)\right]$

(c) In a two dimensional case, we can visualize how Naive Bayes behaves when input features are correlated. A data set shown in Figure 1 (A), where red points are in Class 0, blue points are in Class 1. The conditional distributions are two-dimensional Gaussians. In (B) (C) and (D), the ellipses represent conditional distributions for each class. The centers of ellipses show the mean and the contours show the boundary of two standard deviations. Which of them is most likely to be the true conditional distribution? Which of them is most likely to be estimates by a Gaussian naive Bayes model? If we assume the prior probabilities for both classes are equal, which model will achieve a higher accuracy on the training data?

[3 points]

Most likely to the true conditional distribution = Figure (C.)

Most likely to be estimated by a Gaussian naive Bayes model = Figure(B)

Figure (C) will achieve higher accuracy on the training data.

6: Text classification using Naive Bayes. (TA:- Siddhartha Jain& Ying Yang)

In this assignment, you are going to program a naive Bayes classifier to classify documents from a serious European magazine "economist" (Class 1) and a not-so-serious American megazine "the onion" (Class 0).

1. **Data description**
   If you load the `handout.mat` into Octave (or Matlab) with `load handout.mat`, you will see the following matrices, `Xtrain, Ytrain, Xtest, Ytest`. We also provided a dictionary of V tokens (or words) in `dictionary.mat`, and denote the tokens in the dictionary by indices, $\{1, 2, \cdots, V\}$. There are n training documents and m testing documents. For each document, we counted the number of occurrence of each token, resulting in a vector $(c_1, c_2, \cdots, c_V)$. Each row in `Xtrain` and `Xtest` is such a vector for one document. `Ytrain` and `Ytest` are $n \times 1$ and $m \times 1$ binary class labels.

2. **Model description (multinomial model)**
   We view a document as an ordered sequence of word events. Suppose we have a document with label $Y = y \in \{0, 1\}$, which contains q words in total, we use $W_i = j$ to denote the event that the ith word is the jth token in the dictionary, $j \in \{1, 2, \cdots, V\}$. With a naive Bayes model, we assume that the q word events in one class are independent, and have an identical multinomial distribution with V outcomes.

   Learning the conditional probability
   Given one training document in Class y, if we do not use smoothing (or pseudocounts), we estimate the conditional probability for a word event W in the following way,

$$\begin{aligned}\Pr(W = j|Y = y) &= \frac{\text{number of occurrence of token } j}{\text{total number of words}} \\ &= \frac{\text{number of occurrence of token } j}{\text{total number of occurrence of all } V \text{ tokens}}\end{aligned}$$

   In `Xtrain`, you are given multiple training documents in one class, you should think in a way as concatenating them all into a large document. You need to use additive smoothing (or pseudocount) http://en.wikipedia.org/wiki/Additive_smoothing in your implementation, setting $\alpha = 1$.

   Learning the prior
   Assume the prior distribution of label Y is binomial, without smoothing, it is estimated as

$$\Pr(Y = y) = \frac{\text{number of documents in Class y}}{\text{total number of documents}}$$

   Making prediction
   Now given the test document of length q,

$$\begin{aligned}y* &= \arg\max_y \Pr(Y = y|W_1, \cdots, W_q) = \arg\max_y \frac{\prod_{i=1}^q \Pr(W_i|Y = y)\Pr(Y = y)}{\Pr(W_1, \cdots, W_q)} \\ &= \arg\max_y (\prod_{i=1}^q \Pr(W_i|Y = y)\Pr(Y = y))\end{aligned}$$

However, we are only given the word counts of the document, $(c_1, c_2, \cdots, c_V)$, and we can only compute the multinomial probability.

$$y* = \arg\max_y \left(q!(\prod_{j=1}^{V} \frac{\Pr(W=j|Y=y)^{c_j}}{c_j!}) \Pr(Y=y) \right) \tag{1}$$

$$= \arg\max_y \left((\sum_{j=1}^{V} c_j \log \Pr(W=j|Y=y)) + \log \Pr(Y=y) + \log(q!) - \sum_{j=1}^{V} \log(c_j!) \right) \tag{2}$$

$$= \arg\max_y \left((\sum_{j=1}^{V} c_j \log \Pr(W=j|Y=y)) + \log \Pr(Y=y) + \text{constant} \right) \tag{3}$$

$$= \arg\max_y \left((\sum_{j=1}^{V} c_j \log \Pr(W=j|Y=y)) + \log \Pr(Y=y) \right) \tag{4}$$

In your implementation, to avoid multiplying very small probabilities and underflow, you should use the logarithmic transformation as in Equation 4.

(a) Create following three octave functions and save them in three files, `nb_train.m`, `nb_test.m` and `nb_run.m`.

```
model = nb_train(Xtrain, Y_train)
Pred_nb = nb_test(model, Xtest)
accuracy = nb_run(Xtrain, Ytrain, Xtest, Ytest)
```

`model` is a structure that describes the model you learned. `Pred_nb` is a $m \times 1$ binary vector, which denotes your prediction for the testing documents. In `nb_run`, return the prediction accuracy computed by `accuracy = mean(Pred_nb==Ytest)` , and use `save('Pred_nb.mat','Pred_nb')` to save your prediction into a mat file.

Note: Your score will be determined by your classification accuracy on the test dataset you've been given as well as the held-out dataset that has not been released.

[15 points]

(b) For the jth token in the dictionary, we can compute the following log-ratio,

$$\left| \log \frac{Pr(W=j|Y=1)}{Pr(W=j|Y=0)} \right|$$

Use this log-ratio as a measure, find the top five words that are most discriminative of the classes, report them in your pdf.

Percent, Sox, Yankees, Monday, Schmuck.

[5 points]

(c) State the Naive Bayes assumption. Are there any pairs of words that violate the Naive Bayes assumption? If so, give 1 example of such pairs and explain why they might be violating the Naive Bayes assumption.

The Naive Bayes assumes that any pairs of words that occur in the

[5 points]

Submission instruction: For Q6 **(b)** and **(c)**, write your solutions in your pdf. For Q6 **(a)**

compress your three m-files along with your pdf using `tar` and name it as "hw2.tgz". A good way to do so is to put your m-files and your pdf into a directory. Open a terminal and `cd` to that directory. Then run

```
tar -cvf hw2.tgz *.m  *.pdf
```

Make sure when you decompress `hw2.tgz`, you can directly see the four files, instead of seeing another layer of directory. Otherwise Autolab may fail.

Total: 60

distribution must be independant of each other. But, in our example dataset copus, for example 'wars' and 'spies' pair of words co-occur i.e). they are dependent. Therefore Navie Bayes assumption doesn't holds true here.