10-601: Homework 2

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

TAs: Siddhartha Jain, Ying Yang

Name: Dawei

Andrew ID: daweiwan

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? 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? 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 vector, 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]

Since β is a constant scalar, and x is an observed hence constant value, βx is constant. That being said, we're deviating the original Gaussian distribution $N(0, \sigma^2)$ by an amount of βx, which yields $N(\beta x, \sigma^2)$ by common sense. That is, $Y \sim N(\beta x, \sigma^2)$. ■

(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]

In this case, the logarithmic likelihood function

$$LL\left(Y_1|X_1, Y_2|X_2, \cdots, Y_n|X_n : \beta\right) = \sum_{i=1}^{n} \ln f\left(Y_i|X_i : \beta\right) \tag{1}$$

$$= \sum_{i=1}^{n} \ln \left[\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(Y_i - \beta X_i)^2}{2\sigma^2} \right) \right] \tag{2}$$

$$= n \ln \frac{1}{\sigma\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \beta X_i)^2 \tag{3}$$

which, reaches its maximum when $\sum_{i=1}^{n} (Y_i - \beta X_i)^2$ is at its minimum.
Therefore the maximum likelihood estimation β is

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

■

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 $\hat{\beta}$.

[3 points]

We try to find an explicit solution by its partial differentiation:

$$\frac{\partial}{\partial \beta} \frac{1}{2} \left(\sum_{i=1}^{n} (y_i - \beta x_i)^2 + \lambda \beta^2 \right) = \frac{1}{2} \left(\sum_{i=1}^{n} 2(y_i - \beta x_i)(-x_i) + 2\lambda\beta \right) \tag{4}$$

$$= \sum_{i=1}^{n} (\lambda + x_i^2)\beta - \sum_{i=1}^{n} x_i y_i. \tag{5}$$

where we assume that $\lambda\beta^2$ is summed (it doesn't matter too much). Letting (5) be zero yields:

$$\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} (\lambda + x_i^2)} \tag{6}$$

■

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

[2 points]

$\hat{\beta}$ is a monotonically decreasing function with respect to λ. For $\lambda = 0$, $\hat{\beta}$ devolves into the standard linear regression estimate; as $\lambda \to \infty$, $\beta \to 0$, meaning that the correlation between X and Y is getting weaker and weaker, and is finally diminished when $\lambda = \infty$. This is explained by the fact that the term that suggests a correlation, namely $(y_i - \beta x_i)^2$, is becoming trivial as λ grows. ■

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 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 the models above yields to a lower training error?

[*5 points*]

The maximum likelihood estimate for the first model is known to be

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2} \tag{7}$$

However, we cannot directly assign $\hat{\beta}_2^2 \leftarrow \hat{\beta}_1$ since $\hat{\beta}_1$ might be negative. Consider:

$$\hat{\beta}_2 = \arg\min_{\beta} \sum_{i=1}^n \left(\beta^2 x_i - y_i\right)^2 \tag{8}$$

To find the minimum points, we take its first and second order derivatives:

$$\frac{\partial}{\partial \beta} \sum_{i=1}^n \left(\beta^2 x_i - y_i\right)^2 = \sum_{i=1}^n 2\left(2\beta x_i\right)\left(\beta^2 x_i - y_i\right) = 4\beta^3 \sum_{i=1}^n x_i^2 - 4\beta \sum_{i=1}^n x_i y_i \tag{9}$$

$$\frac{\partial^2}{\partial^2 \beta} \sum_{i=1}^n \left(\beta^2 x_i - y_i\right)^2 = 12\beta^2 \sum_{i=1}^n x_i^2 - 4 \sum_{i=1}^n x_i y_i \tag{10}$$

If $\sum_{i=1}^n x_i y_i \geq 0$, we have two minimum points at $\pm\hat{\beta}_1^{0.5}$, and the second model will have the same performance as the first model, yielding the same training error; otherwise, the only minimum point would be 0, and the second model will have worse performance, yielding a higher training error than the first model since it suggests no correlation between X and Y (i.e., not trained).

Formally, the least square solution for the second model is:

$$\hat{\beta}_2 = \begin{cases} \pm\left[\frac{\sum_{i=1}^n x_i y_i}{\sum_{i=1}^n x_i^2}\right]^{0.5} & \sum_{i=1}^n x_i y_i > 0 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

Intuitively 1. is always better than 2. since β can be assigned with arbitrary real value, unlike β^2, which is limited to non-negatives. ■

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*]

$\beta_1 = 1, \beta_2 = 0$. Computed with the following statements in Octave:

```
> A = ones(101, 2); A(101, 2) = 0; y = ones(101, 1);
> P = inv(A'*A)*A'*y;
```

■

(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*]

$\beta_1 = 0, \beta_2 = 1$. Computed with the following statements in Octave:

```
> B = ones(103, 2); B(101, 2) = 0; B(102:103, 1) = 0; z = ones(103, 1);
> Q = inv(B'*B)*B'*z;
```

■

(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*]

Consider k, m, y as the coordinates of a point in a 3-dimensional space, i.e., $P(k, m, y)$. In the original dataset, there are 100 points stacking at $A(1, 1, 1)$, (almost) guaranteeing that the model output would pass that particular point. However, there is no constraint on the line's orientation, such that any point further supplied can be decisive on β. In (a), β describes a line from A to (1, 0, 1); In (b), β describes a line from A to $(0, 1, 1)$. It is expected, geometrically, that β should be fluctuating wildly with such *deliberately crafted* points.

To test whether such behavior might occur, investigate whether there is a dense point cloud dwelling somewhere in the joint observed input/output space. ■

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, 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]

Here we need to estimate the mean and variance for each individual feature:

$$\ln L\left(X^j|Y; \mu_Y, \sigma_Y^2\right) = k_Y \ln \frac{1}{\sigma_Y^{(j)}\sqrt{2\pi}} - \frac{1}{2(\sigma_Y^{(j)})^2} \sum_{\Omega} \left(X_Y^{(j)} - \mu_Y^{(j)}\right)^2 \tag{12}$$

where the sum is performed over all data points labeled as class Y. Letting the derivatives be zero,

$$\frac{\partial \ln L}{\partial \mu_Y^{(j)}} = \frac{1}{(\sigma_Y^{(j)})^2}\left[-k_Y \mu_Y^{(j)} + \sum_{\Omega} x_Y^{(j)}\right] = 0 \tag{13}$$

$$\frac{\partial \ln L}{\partial (\sigma_Y^{(j)})^2} = -\frac{1}{2} k_Y \frac{2 \cdot 2\pi}{2\pi(\sigma_Y^{(j)})^2} + \frac{1}{2}\frac{1}{(\sigma_Y^{(j)})^4} \sum_{\Omega} \left(X_Y^{(j)} - \hat{\mu}_Y^{(j)}\right)^2 = 0 \tag{14}$$

we can obtain a maximum likelihood estimate: ■

$$\hat{\mu}_Y^{(j)} = \frac{1}{k_Y} \sum_{\Omega} x_Y^{(j)} \quad (\sigma_Y^{(j)})^2 = \frac{1}{k_Y} \sum_{\Omega} \left(X_Y^{(j)} - \mu_Y^{(j)}\right)^2 \tag{15}$$

(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?

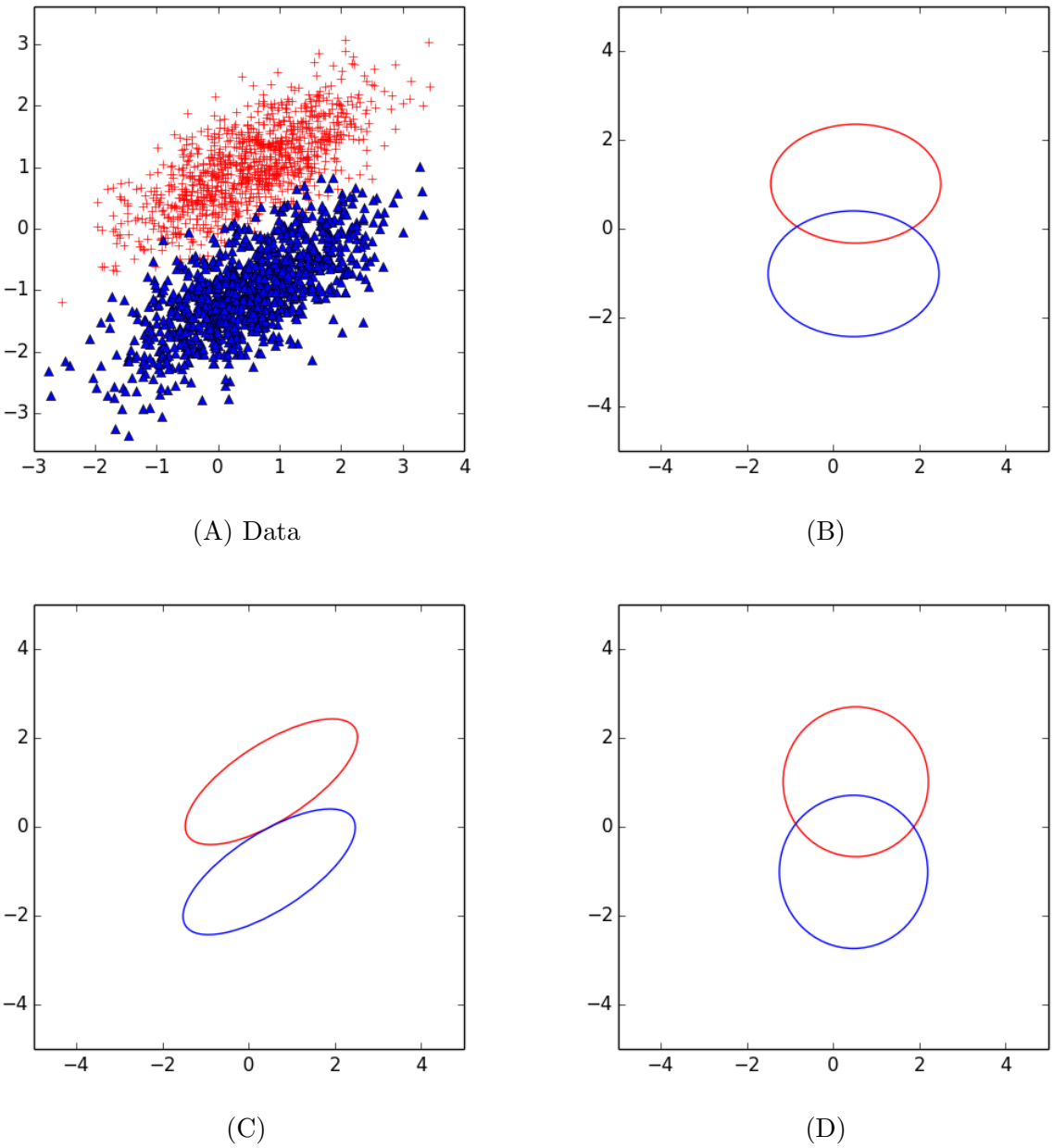
[3 points]

In the naive Bayes model, we need $2p$ parameters, with p for the mean vector μ, and the rest for the diagonal elements in Σ. In the full model, we need $0.5p(p+1) + p$ parameters, where p comes from the mean vector, and $0.5p(p+1)$ comes from the covariance matrix, since it's symmetric. ■

(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]

(C) is most likely to be the true conditional distribution. (B) is most likely to be estimates of a Gaussian naive Bayes model. The full Gaussian model always has equal or better performance than the naive Bayes one since it has additional degrees of freedom. ■



(A) Data (B)

(C) (D)

Figure 1: Figure of Q6 (c)

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**
   All data files are in `Onion_vs_Economist`. 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 `Xtrain` 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 the event $W_i = j$ 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 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,

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

   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

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

   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.

$$y* = \arg\max_y (q! \prod_{j=1}^{V} \frac{\Pr(W = j|Y = y)}{c_j!} \Pr(Y = y) \tag{16}$$

$$= \arg\max_y (\sum_{i=1}^{V} \log \Pr(W = j|Y = y) + \log \Pr(Y = y)) + \log(q!) - \sum_{i=1}^{V} \log(c_j!) \tag{17}$$

$$= \arg\max_y (\sum_{i=1}^{V} \log \Pr(W = j|Y = y) + \log \Pr(Y = y)) + \text{constant} \tag{18}$$

In your implementation, to avoid multiplying very small probabilities and underflow, you should use the logarithmic transformation as in Equation 18.

For **(a)** submit your m-files to autolab. For **(b)** and **(c)**, write your solutions in your pdf.

(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 describe 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.

The functions are quite short. I just pasted them here for completeness.

```
function model = nb_train(x, y)
  cumcount = [!y'*x; y'*x]; % concatenate & count;
  model.prior = [!y'*!y; y'*y] / rows(y); % compute priors;
  model.condpr = bsxfun(@rdivide, cumcount + 1, sum(cumcount, 2) + columns(x));
endfunction
```

```
function Pred_nb = nb_test(model, x)
  likelihood = bsxfun(@plus, log(model.condpr)*x', model.prior);
  Pred_nb = diff(likelihood)' > 0;
endfunction
```

```
function accuracy = nb_run(Xtrain, Ytrain, Xtest, Ytest)
  model = nb_train(Xtrain, Ytrain);
  Pred_nb = nb_test(model, Xtest);
  accuracy = mean(Pred_nb == Ytest);
  save('Pred_nb.mat', 'Pred_nb');
endfunction
```

[*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.

[*5 points*]

This can be done by typing the following commands (provided that files are loaded):

```
model = nb_train(Xtrain, Ytrain, Xtest, Ytest);
[~, idx] = sort(abs(diff(log(model.condpr))), "descend");
dict(idx(1: 5));
```

and the most discriminative words are: *percent*, *monday*, *yankees*, *sox*, *schmuck*. ■

(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.

[*5 points*]

Naive Bayes: Given the class label, the presence of words are conditionally mutually independent. It would be impossible to proof or refute the dependence among any pair of words based on finite number of observations. But in this problem's scenario per se, it might be feasible to find some by common sense. For example, *the* and *onion* could be one; or *the* and *economist*, since they're known to be appearing together. ■

Total: 60