COMS 4771 HW 1 (Spring 2020)

Due: Friday Feb 21, 2020 at 11:59pm

You are allowed to write up solutions in groups of (at max) three students. These group members don't necessarily have to be the same from previous homeworks. Only one submission per group is required by the due date on Gradescope. Name and UNI of all group members must be clearly specified on the homework. No late homeworks are allowed. To receive credit, a typesetted copy of the homework pdf must be uploaded to Gradescope by the due date. You must show your work to receive full credit. Discussing possible solutions for homework questions is encouraged on piazza and with peers outside your group, but every group must write their own individual solutions. You should cite all resources (including online material, books, articles, help taken from specific individuals, etc.) you used to complete your work.

1 Statistical Estimators

Here we will study some statistical estimators.

(i) Given $a, b \in \mathbb{R}$ s.t. $a < b$, consider the density $p(x \mid \theta = (a,b)) \propto \begin{cases} 1 & \text{if } a \le x \le b \\ 0 & \text{otherwise} \end{cases}$. Suppose that n samples $x_1, \ldots, x_n$ are drawn i.i.d. from $p(x|\theta)$. What is the Maximum Likelihood Estimate (MLE) of θ given the samples?

(ii) Show that for the MLE θ_{ML} of a parameter $\theta \in \mathbb{R}^d$ and any known differentiable function $g : \mathbb{R}^d \to \mathbb{R}^k$, the MLE of $g(\theta)$ is $g(\theta_{\text{ML}})$.

(iii) For a 1-dimensional Gaussian distribution, give two examples for each of the following types of estimators for the mean parameter.

- consistent and unbiased.
- consistent, but not unbiased.
- not consistent, but unbiased.
- neither consistent, nor unbiased.

2 On Forecasting Product Demand

One way retail industry uses machine learning is to predict how much quantity Q of some product to they should buy to maximize their profit. The optimal quantity depends on how much demand D there is for the product as well as its cost for the retailer to buy C and its selling price P to the customer. Assuming that the demand D is distributed as $P(D)$, we can evaluate the expected profit considering two cases:

- if $D \geq Q$, then the retailer sells all Q items and make a profit $\pi = (P - C)Q$.
- but if $D < Q$, then the retailer can only sell D items at profit $(P - C)D$, but has lost $C(Q - D)$ on unsold items.

1. What is the expected profit if the retailer buys Q items? Simplify the expression as much as possible.
2. By taking the derivative (wrt Q) of the above expression for expected profit, show that the optimal quantity Q^* to by satisfies $Q^* = F^{-1}(1 - (C/P))$, where F is the cdf of D. That is, the optimal Q^* is when the cumulative density (of D) equals $1 - (C/P)$.

3 Evaluating Classifiers

Consider the following decision rule f_t for a two-category problem in $\mathbb{R}$. Given an input $x \in \mathbb{R}$

decide category y_1, if $x > t$; otherwise decide category y_2

(i) What is the error rate for this rule, that is, what is $P[f_t(x) \neq y]$?

(ii) Show that at for the optimally selected threshold value t (i.e., the one which gives minimum error rate), it must be the case that

$$P(X = t|Y = y_1)P(Y = y_1) = P(X = t|Y = y_2)P(Y = y_2).$$

(iii) Assume that the underlying population distribution has equal class priors (i.e., $P[Y = y_1] = P[Y = y_2]$), and the individual class conditionals (i.e., $P[X|Y = y_1]$ and $P[X|Y = y_1]$) are distributed as Gaussians. Give an example setting of the class conditionals (i.e., give an example parameter settings for the Gaussians) such that for some threshold value t, the rule f_t achieves the Bayes error rate; and similarly, give an example setting of the class conditionals such that for no threshold value t, the rule f_t achieves the Bayes error rate.

4 Analyzing iterative optimization

In this problem, we will analyze the Richardson iteration for finding $\beta \in \mathbb{R}^d$ that (approximately) minimizes $\|A\beta - b\|_2^2$ for a given matrix $A \in \mathbb{R}^{n \times d}$ and vector $b \in \mathbb{R}^n$.

Recall the Richardson iteration, given as follows:

- Initially, $\beta^{(0)} = (0, \ldots, 0) \in \mathbb{R}^d$ is the zero vector in $\mathbb{R}^d$.
- For $k = 1, 2, \ldots, N$:
  - Compute $\beta^{(k)} := \beta^{(k-1)} + \eta A^{\mathsf{T}}(b - A\beta^{(k-1)})$.

Above, $\eta > 0$ is a fixed positive number called the step size, and N is the total number of iterations. Define $M := A^{\mathsf{T}}A$ and $v := A^{\mathsf{T}}b$.

(i) Show that the matrix M is symmetric positive semi-definite.

Throughout, assume that the eigenvalues of M, denoted by $\lambda_1, \dots, \lambda_d$, satisfy $\lambda_i < 1/\eta$ for all $i = 1, \dots, d$.

(ii) Prove (e.g., using mathematical induction) that, for any positive integer N,

$$\beta^{(N)} = \eta \sum_{k=0}^{N-1} (I - \eta M)^k v.$$

(Here, for a square matrix B, we have $B^0 = I$, $B^1 = B$, $B^2 = BB$, $B^3 = BBB$, and so on.)

(iii) What are the eigenvalues of $\eta \sum_{k=0}^{N-1} (I - \eta M)^k$? Give your answer in terms of $\lambda_1, \dots, \lambda_d$, η, and N.

(iv) Let $\hat{\beta}$ be any vector in the range of M satisfying $M\hat{\beta} = v$. Prove that

$$\|\beta^{(N)} - \hat{\beta}\|_2^2 \leq e^{-2\eta\lambda_{\min}N} \|\hat{\beta}\|_2^2,$$

where $\lambda_{\min}$ is the smallest non-zero eigenvalue of M.

Hint: You may use the fact that $1 + x \leq e^x$ for any $x \in \mathbb{R}$.

This implies that as the number of iterations N increases, the difference between our estimate $\beta^{(N)}$ and $\hat{\beta}$ decreases exponentially!

5 Designing socially aware classifiers

Traditional Machine Learning research focuses on simply improving the accuracy. However, the model with the highest accuracy may be discriminatory and thus may have undesirable social impact that unintentionally hurts minority groups[1]. To overcome such undesirable impacts, researchers have put lots of effort in the field called Computational Fairness in recent years.

Two central problems of Computational Fairness are: (1) what is an appropriate definition of fairness that works under different settings of interest? (2) How can we achieve the proposed definitions without sacrificing on prediction accuracy?

In this problem, we will focus on some of the ways we can address the first problem. There are two categories of fairness definitions: individual fairness[2] and group fairness[3]. Most works in the literature focus on the group fairness. Here we will study some of the most popular group fairness definitions and explore them empirically on a real-world dataset.

Generally, group fairness concerns with ensuring that group-level statistics are same across all groups. A group is usually formed with respect to a feature called the **sensitive attribute**. Most common sensitive features include: gender, race, age, religion, income-level, etc. Thus, group fairness ensures that statistics across the sensitive attribute (such as across, say, different age groups) remain the same.

[1] see e.g. **Machine Bias** by Angwin et al. for bias in recidivism predication, and **Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification** by Buolamwini and Gebru for bias in face recognition

[2] see e.g. **Fairness Through Awareness** by Dwork et al.

[3] see e.g. **Equality of Opportunity in Supervised Learning** by Hardt et al.

For simplicity, we only consider the setting of binary classification with a single sensitive attribute. Unless stated otherwise, we also consider the sensitive attribute to be binary. (Note that the binary assumption is only for convenience and results can be extended to non-binary cases as well.)
Notations:

Denote $X \in \mathbb{R}^d,\ A \in \{0,1\}$ and $Y \in \{0,1\}$ to be three random variables: non-sensitive features of an instance, the instance's sensitive feature and the target label of the instance respectively, such that $(X, A, Y) \sim \mathcal{D}$. Denote a classifier $f : \mathbb{R}^d \to \{0,1\}$ and denote $\hat{Y} := f(X)$.

For simplicity, we also use the following abbreviations:

$$\mathbb{P} := \mathbb{P}_{(X,A,Y)\sim D} \qquad \text{and} \qquad \mathbb{P}_a := \mathbb{P}_{(X,a,Y)\sim D}$$

We will explore the following are three fairness definitions.
- *Demographic Parity (DP)*

$$\mathbb{P}_0[\hat{Y} = \hat{y}] = \mathbb{P}_1[\hat{Y} = \hat{y}] \qquad \forall \hat{y} \in \{0,1\}$$

(equal positive rate across the sensitive attribute)

- *Equalized Odds (EO)*

$$\mathbb{P}_0[\hat{Y} = \hat{y} \mid Y = y] = \mathbb{P}_1[\hat{Y} = \hat{y} \mid Y = y] \qquad \forall \hat{y},\ y \in \{0,1\}$$

(equal true positive- and true negative-rates across the sensitive attribute)

- *Predictive Parity (PP)*

$$\mathbb{P}_0[Y = y \mid \hat{Y} = \hat{y}] = \mathbb{P}_1[Y = y \mid \hat{Y} = \hat{y}] \qquad \forall \hat{y},\ y \in \{0,1\}$$

(equal positive predictive- and negative predictive-value across the sensitive attribute)

Part 0: The basics.

(i) Why is it not enough to just remove the sensitive attribute A from the dataset to achieve fairness as per the definitions above? Explain with a concrete example.

Part 1: Sometimes, people write the same fairness definition in different ways.

(ii) Show that the following two definitions for *Demographic Parity* is equivalent under our setting:

$$\mathbb{P}_0[\hat{Y} = 1] = \mathbb{P}_1[\hat{Y} = 1] \iff \mathbb{P}[\hat{Y} = 1] = \mathbb{P}_a[\hat{Y} = 1] \qquad \forall a \in \{0,1\}$$

(iii) Generalize the result of the above equivalence and state an analogous equivalence relationship of two equality when $A \in \mathbb{N}$, and $\hat{Y} \in \mathbb{R}$.

Part 2: In this part, we will explore the COMPAS dataset (available in `hw1data.zip`). The task is to predict two year recidivism. Download the COMPAS dataset from the class's website. In this dataset, the target label Y is `two_year_recid` and the sensitive feature A is `race`.

(iv) Develop the following classifiers: (1) MLE based classifier, (2) nearest neighbor classifier, and (3) naïve-bayes classifier, for the given dataset.

For MLE classifier, you can model the class conditional densities by a Multivariate Gaussian distribution. For nearest neighbor classifier, you should consider different values of k and the distance metric (e.g. L_1, L_2, L_∞). For the naïve-bayes classifier, you can model the conditional density for each feature value as count probabilities.

(you may use builtin functions for performing basic linear algebra and probability calculations but you should write the classifiers from scratch.)

You must submit your code to Courseworks to receive full credit.

(v) Which classifier (discussed in previous part) is better for this prediction task? You must justify your answer with appropriate performance graphs demonstrating the superiority of one classifier over the other. Example things to consider: how does the training sample size affects the classification performance.

(vi) To what degree the fairness definitions are satisfied for each of the classifiers you developed? Show your results with appropriate performance graphs.

For each fairness measure, which classifier is the most fair? How would you summarize the difference of these algorithms?

(vii) Choose any one of the three fairness definitions. Describe a real-world scenario where this definition is most reasonable and applicable. What are the potential disadvantage(s) of this fairness definition?

(You are free to reference online and published materials to understand the strengths and weaknesses of each of the fairness definitions. Make sure cite all your resources.)

(viii) [Optional problem, will not be graded] Can an algorithm simultaneously achieve high accuracy and be fair and unbiased on this dataset? Why or why not, and under what fairness definition(s)? Justify your reasoning.

6 Email spam classification case study

Download the datafile `email_data.tar.gz`. This datafile contains email data of around 5,000 emails divided in two folders 'ham' and 'spam' (there are about 3,500 emails in the 'ham' folder, and 1,500 emails in the 'spam' folder). Each email is a separate text file in these folders. These emails have been slightly preprocessed to remove meta-data information.

(i) (Embedding text data in Euclidean space) The first challenge you face is how to systematically embed text data in a Euclidean space. It turns out that one successful way of transforming text data into vectors is via "Bag-of-words" model. Basically, given a dictionary of all possible words in some order, each text document can be represented as a word count vector of how often each word from the dictionary occurs in that document.

Example: suppose our dictionary D with vocabulary size 10 ($|D| = 10$). The words (ordered in say alphabetical order) are:

1: also

2: football

3: games

4: john

5: likes

6: Mary

7: movies

8: to

9: too

10: watch

Then any text document created using this vocabulary can be embedded in $\mathbb{R}^{|}D|$ by counting how often each word appears in the text document.

Say, an example text document t is:

```
John likes to watch football.  Mary likes movies.
```

Then the corresponding word count vector in $|D| = 10$ dimensions is:

```
[ 0 1 0 1 2 1 1 1 0 1]
```

(because the word "also" occurs 0 times, "football" occurs 1 time, etc. in the document.)

While such an embedding is extremely useful, a severe drawback of such an embedding is that it treats similar meaning words (e.g. watch, watches, watched, watching, etc.) independently as separate coordinates. To overcome this issue one should preprocess the entire corpus to remove the common trailing forms (such as "ing", "ed", "es", etc.) and get only the root word. This is called word-stemming.

Your first task is to embed the given email data in a Euclidean space by: first performing word stemming, and then applying the bag-of-words model.

Some useful references:

- Bag-of-words: http://en.wikipedia.org/wiki/Bag-of-words_model
- Word stemming: http://en.wikipedia.org/wiki/Stemming

(ii) Once you have a nice Euclidean representation of the email data. Your next task is to develop a spam classifier to classify new emails as `spam` or `not-spam`. You should compare performance of naive-bayes, nearest neighbor (with L_1, L_2 and L_∞ metric) and decision tree classifiers.

(you may use builtin functions for performaing basic linear algebra and probability calculations but you should write the classifiers from scratch.)

You must submit your code to Courseworks to receive full credit.

(iii) Which classifier (discussed in part (ii)) is better for the email spam classification dataset? You must justify your answer with appropriate performance graphs demonstrating the superiority of one classifier over the other. Example things to consider: you should evaluate how the classifier behaves on a holdout 'test' sample for various splits of the data; how does the training sample size affects the classification performance.