COMS 4771-2 F18 Homework 5 (due December 5, 2018)

Instructions

Submit your write-up on Gradescope as a neatly typeset (not scanned nor handwritten) PDF document by 11:59 PM of the due date.

On Gradescope, be sure to select the pages containing your answer for each problem. More details can be found on the Gradescope Student Workflow help page:


(If you don’t select the pages containing your answer to a problem, you’ll receive a zero for that problem.)

Make sure your name and your UNI appears prominently on the first page of your write-up.

Source code

Please combine all requested source code files into a single ZIP file, along with a plain text file called README that contains your name and briefly describes all of the other files in the ZIP file. Do not include the data files. Submit this ZIP file on Courseworks.

Clarity and precision

One of the goals in this class is for you to learn to reason about machine learning problems and algorithms. To reason about these things, you must be able to make clear and precise claims and arguments about them.

A clear and precise argument is not the same as a long, excessively detailed argument. Unnecessary details and irrelevant side-remarks often make an argument less clear. And non-factual statements also detract from the clarity of an argument.

Points may be deducted for answers and arguments that lack sufficient clarity or precision. Moreover, a time-economical attempt will be made to interpret such answers/arguments, and the grade you receive will be based on this interpretation.

Updates (11/24):

- In Problem 1c, find a step size that works when $w^{(1)} = (r, r)$ for any $r \in [0, 0.1]$.
- In Problem 2d, you may assume $m \geq \frac{1600}{\epsilon^2 \min\{\pi_0, \pi_1\}}$. (If you assume $m \geq \frac{1600}{\epsilon^2 \min\{\pi_0, \pi_1\}}$, then you may receive partial credit.) you may find it useful to use Chebyshev’s inequality, which is really Markov’s inequality in disguise: For any random variable $X$ and any $t > 0$,

  $$\mathbb{P}(|X - \mathbb{E}(X)| > t) \leq \frac{\text{var}(X)}{t^2}.\$$

  It follows immediately from Markov’s inequality by applying Markov’s inequality to the random variable $(X - \mathbb{E}(X))^2$.
- In Problem 2e, assume $\ell(z) \in [0, 1]$ for all $z \in \mathbb{R}$.

---

Problem 1 (30 points)

In this problem, you will study the effect of step sizes in gradient descent.

(a) Consider the function $f: \mathbb{R}^d \rightarrow \mathbb{R}$ defined by $f(w) := \|w - q\|_2$. Derive the following formula for the gradient of $f$ at any point $w \neq q$:

$$\nabla f(w) = \frac{1}{\|w - q\|_2} (w - q).$$

Explain every step of your derivation.

(b) Continue from Part (a). Suppose $d = 2$ and $q = (1, 1)$. Run gradient descent with a constant step size $\eta = 1$, starting from $w^{(1)} = (0, 0)$. Precisely characterize the sequence of iterates $(w^{(t)})_{t \geq 1}$ produced this way. Also, show a plot of $f(w^{(t)})$ as a function of $t$, for $t = 1, 2, \ldots, 50$.

(c) Continue from Part (b). Suppose $\epsilon \in (0, 0.1)$ is given. Determine the largest constant step size $\eta \in (0, 1)$ so that the sequence of iterates, starting from $w^{(1)} = (r, r)$ for any $r \in [0, 0.1]$, is eventually (i.e., as $t \rightarrow \infty$) within Euclidean distance $\epsilon$ of $q$. Your answer should be given in terms of $\epsilon$ (and cannot depend on $r$, since it must work for any $r \in [0, 0.1]$). Briefly justify your answer.

(d) Continue from Part (c). Run gradient descent with a decaying sequence of step sizes, $\eta_t = 1/\sqrt{t}$ for $t = 1, 2, \ldots$, starting from $w^{(1)} = (0, 0)$. Show a plot of $f(w^{(t)})$ as a function of $t$, for $t = 1, 2, \ldots, 50$.

No need to submit any source code for this problem.
Problem 2 (40 points)

In this problem, you will practice reasoning about various classification objectives.

(a) Suppose you face a binary classification problem with input space $\mathcal{X} = \mathbb{R}$ and output space $\mathcal{Y} = \{0, 1\}$, where it is $c$ times as bad to commit a “false positive” as it is to commit a “false negative” (for some real number $c \geq 1$). To make this concrete, let’s say that if your classifier predicts 1 but the correct label is 0, you incur a penalty of $c$ dollars; if your classifier predicts 0 but the correct label is 1, you incur a penalty of 1 dollar. (And you incur no penalty if your classifier predicts the correct label.)

Let $P$ be the probability distribution on $\mathcal{X} \times \mathcal{Y}$ that you care about, and $(X, Y) \sim P$. Suppose $P(Y = 0) = 2/3$ and $P(Y = 1) = 1/3$, and the class conditional densities of $X$ are

$$X | Y = 0 \sim \mathcal{N}(0, 1), \quad X | Y = 1 \sim \mathcal{N}(2, 1/4).$$

Let $f^* : \mathbb{R} \rightarrow \{0, 1\}$ be the classifier with the smallest expected penalty.

Assume $1 \leq c \leq 14$. Precisely specify the subset of $\mathbb{R}$ in which the classifier $f^*$ predicts 1.

An example of a precise specification is $[0, 5c] \cup [6c, +\infty)$.

(b) Same as Part (a), except now instead assume $c \geq 15$.

(c) Let $Y \sim P$ be a random variable taking values in $\{-1, +1\}$, with $P(Y = +1) = \mu$ and $P(Y = -1) = 1 - \mu$; here, $\mu$ is some number in $(0, 1)$. Define the loss function $\ell_{\text{msq}} : \mathbb{R} \rightarrow \mathbb{R}$ by

$$\ell_{\text{msq}}(z) := \left(\max\{1 - z, 0\}\right)^2.$$  

(This is the “modified” squared loss.) What is the minimizer of the function $f : \mathbb{R} \rightarrow \mathbb{R}$ given by $f(s) := \mathbb{E}[\ell_{\text{msq}}(Ys)]$? Your answer should be given in terms of $\mu$. Clearly explain every step of your derivation.

(d) Suppose you have a classifier $f : \mathcal{X} \rightarrow \{0, 1\}$, and you would like to estimate its balanced error rate (BER) (which was defined in lecture). You have access to an iid sample $(X_1, Y_1), \ldots, (X_m, Y_m)$ from the distribution $P$ over $\mathcal{X} \times \{0, 1\}$ that you care about, which you assume satisfies $0 < P(Y_1 = 1) < 1$. Precisely describe an estimator $\hat{\text{BER}} = \hat{\text{BER}}(f; (X_1, Y_1), \ldots, (X_m, Y_m))$ of the balanced error rate of $f$ based on this iid sample with the following property: for any $\varepsilon \in (0, 1)$, if $m \geq \frac{1600}{\min\{\pi_0, \pi_1\}}$, then with probability at least 0.99 (over the random draw of the iid sample),

$$|\hat{\text{BER}} - \text{BER}| \leq \varepsilon.$$

Above, $\pi_y = P(Y_1 = y)$ for each $y \in \{0, 1\}$. Prove that your estimator has the required property.\(^2\)

Hint: Use the probability tail inequalities from previous lectures.

(e) Precisely describe an estimator of the $\ell$-loss risk $\mathbb{E}_{(X, Y) \sim P}[\ell(Yh(X))]$ of a scoring function $h : \mathcal{X} \rightarrow \mathbb{R}$ that is based on iid examples collected in the “Type 3” fashion described in lecture. Your estimator should be unbiased and, assuming $q(x) > 0.1$ for all $x \in \mathcal{X}$ and $\ell(z) \in [0, 1]$ for all $z \in \mathbb{R}$, should have variance $O(1/n)$, where $n$ is the total number of “coins” tossed in the collection process. You may assume $n$ is known and can be used by the estimator. Prove that your estimator has these properties.

\(^2\)The constant 1600 appearing in the property is chosen to give you a lot of leeway in the proof; it should be possible to obtain a stronger property where this constant is much smaller.
Problem 3 (30 points)

In this problem, you will train neural networks (loosely defined) to solve a regression problem.

1. Sign-up for a Kaggle account with your Columbia e-mail address.
2. Join the Kaggle competition for this assignment at the following URL:
   https://www.kaggle.com/t/5139175f73b2489aa3d231c4d6299c89
3. Set your “Team Name” to your UNI (e.g., abc1234).

The learning problem is to predict the release year of a song from its audio features. The training data
(data.csv), available from the Kaggle competition page, consists of 90 highly-engineered audio features for
463715 songs from the past century. Each song is also labeled with its release year. Your task is to use this
data to train a neural network to predict the release year of a song.

The accuracy of your neural network will be judged by its expected absolute loss on test examples. The
features of the test examples can also be downloaded from the Kaggle competition page (testdata.csv).
The absolute loss is given by
\[
\ell(\hat{y}, y) = |\hat{y} - y|.
\]
(This does not necessarily mean you have to use the absolute loss in training, but it might be a good idea.)

Note that neural networks also include linear models, with or without hand-designed feature expansion. (A
reasonable baseline is a linear model on the original features.)

Evaluation (15 points)

Submit your predictions for the test examples in the format specified in the sample submission file
(sample.csv), again available on the Kaggle competition page. You can submit predictions 10 times
per day. The performance on 30% of the test examples will be reported on the “public leaderboard”. But you
will ultimately be evaluated on the remaining 70% of the test examples (the “private” test set).

The final submission must be made by 11:59 PM UTC on December 5th, which is 6:59 PM
EST on December 5th.

You need to get expected absolute loss on the “private” test set less than 6.55 to earn full credit on this part.
If your expected absolute loss is higher than that, points will be deducted.

Write-up (15 points)

In your homework write-up, describe the methodology you used to train your neural network. You don’t
need to (for instance) strictly follow the model selection procedures described in class. But, you should apply
the principles you learned in class and justify your approach in your write-up.

Give detailed descriptions of all the methods you tried (e.g., feature expansions, training algorithms, selection
criteria), as well as the process by which you arrived at your final predictor. Give enough details for the final
predictor so that another student in the class will be able to replicate it exactly. Give proper citations for
anything you did not come up with or implement yourself.

Please submit your source code on Courseworks. Do not submit any data files.

Extra credit

Up to five points of extra credit will be awarded for the highest ranked submissions on the “private”
leaderboard.