6.790 Homework 5

Revision: 11/2/24 7:00PM

Solutions

Please hand in your work via Gradescope via the link at https://gradml.mit.edu/info/homeworks/. If you were not added to the course automatically, please use Entry Code R7RGGX to add yourself to Gradescope. Make sure to assign the problems to the corresponding pages in your solution when submitting via Gradescope.

1. Latex is not required, but if you are hand-writing your solutions, please write clearly and carefully. You should include enough work to show how you derived your answers, but you don't have to give careful proofs.
2. Homework is due on Tuesday November 12 at 11PM.
3. Lateness and extension policies are described at https://gradml.mit.edu/info/class_policy/.

Contents

Solution: Don't look at the solutions until you have tried your absolute hardest to solve the problems. This is especially true for optional problems that you didn't work on—it's a good idea to come back to them when studying for exams.

Robust Binary Classification (30 points)

1. In this problem will study **robust** binary classification. For any input $x \in \mathbb{R}^d$, the hypothesis function is

$$h_\theta(x) = w^T x + b, \text{ where } \theta = \{w \in \mathbb{R}^d, b \in \mathbb{R}\}.$$

The class label $y \in \{+1, -1\}$. The prediction probability of class $+1$ is given as

$$p(y = +1|x, \theta) = \frac{1}{1 + \exp(-h_\theta(x))}$$

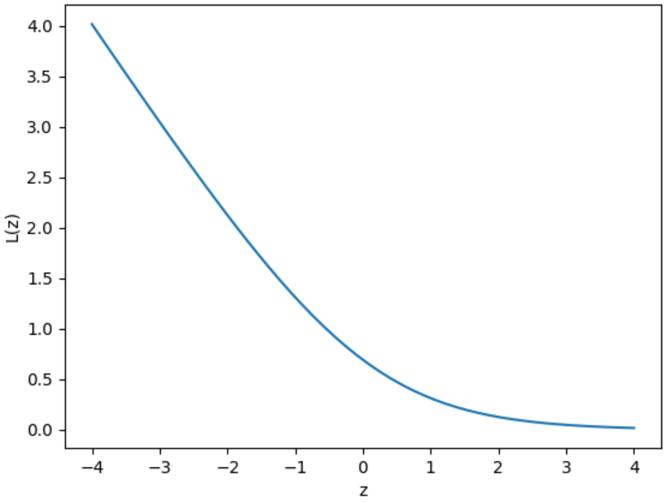
and $p(y = +1|x, \theta) = 1 - p(y = -1|x, \theta)$.

We will use the following loss function:

$$\ell(h_\theta(x), y) = -\log P(y|x, \theta) = \log\left(1 + \exp(-y h_\theta(x))\right)$$

(a) (5 points) The loss $\ell(h_\theta(x), y)$ can be written as $L(z) = \log(1 + \exp(-z))$, where $z = y h_\theta(x)$. What can we say about the monotonicity of $L(z)$ with respect to z?

Solution: The loss $L(z)$ is monotonically decreasing.



The robust binary classification problem can now be defined as:

$$\min_{\theta} \max_{\delta:||\delta|| \leqslant \epsilon} \ell(h_\theta(x + \delta), y) \tag{1}$$

where $||\cdot||$ denotes a valid norm.

To understand how this formulation works, we need an intermediate step of understanding so called dual norms. They are defined as

$$\|\theta\|_* = \max_{\|\delta\|\leqslant 1} \delta^T\theta \tag{2}$$

where, for example, $\|\cdot\|$ could be the 2-norm $\|\delta\|_2 = (\sum_{j=1}^d \delta_j^2)^{1/2}$ or the ℓ-infinity norm $\|\delta\|_\infty = \max_{j=1}^d |\delta_j|$. As a warm-up exercise, let's solve what the corresponding dual norms are.

(b) (5 points) Find what norms the dual norms $\|\theta\|_* = \max_{\|\delta\|_2\leqslant 1} \delta^T\theta$ and $\|\theta\|_* = \max_{\|\delta\|_\infty\leqslant 1} \delta^T\theta$ are.

Solution:
By Cauchy-Schwarz, we have that $\delta^T\theta \leqslant \|\delta\|_2\|\theta\|_2 = \|\theta\|_2$, where equality is achieved when δ and θ are parallel. Thus, the dual norm of the 2-norm is just the 2-norm.
Also, we note that

$$\max_{\|\delta\|_\infty\leqslant 1} \delta^T\theta = \max_{\|\delta\|_\infty\leqslant 1} \sum_{j=1}^d \delta_j\theta_j.$$

Since $\delta_j\theta_j \leqslant |\theta_j|$ when $|\delta_j| \leqslant 1$, with equality occurring at $\delta_j = \text{sgn}(\theta_j)$, the expression simplifies to

$$\max_{\|\delta\|_\infty\leqslant 1} \sum_{j=1}^d \delta_j\theta_j = \sum_{j=1}^d |\theta_j|,$$

so the dual of the ℓ-infinity norm is just the 1-norm.

(c) (5 points) The inner maximization problem in (3) i.e. $\max_{\delta:\|\delta\|\leqslant\epsilon} L(yh_\theta(x+\delta))$, using $L(\cdot)$ can be written as:

$$\max_{\delta:\|\delta\|\leqslant\epsilon} L(y(w^T(x+\delta)+b))$$

Find the closed form solution of this inner maximization problem when:
1) $\|\cdot\|$ is $\infty-$norm.
2) $\|\cdot\|$ is $2-$norm.
Hint: [Use the previous parts]

Solution:
Note that in $L(y(w^T(x+\delta)+b))$, the quantity $y(w^T(x+\delta)+b)$ is a scalar. As $L(z)$ is monotonically decreasing,

$$\max_{\delta:\|\delta\|\leqslant\epsilon} L(y(w^T(x+\delta)+b)) = L\left(\min_{\delta:\|\delta\|\leqslant\epsilon} y(w^T(x+\delta)+b)\right)$$

We therefore need to solve

$$\min_{\delta:\|\delta\|\leqslant\epsilon} y(w^T(x+\delta)+b)$$

This can be simplified as

$$\min_{\delta:||\delta||\leqslant\epsilon} y(w^T x + b) + yw^T\delta$$

Using the result from previous part we can write: if $||\cdot||$ is $\infty -$ norm, then

$$\min_{\delta:||\delta||\leqslant\epsilon} y(w^T x + b) + yw^T\delta = y(w^T x + b) - \epsilon||yw||_1 = y(w^T x + b) - \epsilon||w||_1$$

if $||\cdot||$ is $2 -$ norm, then

$$\min_{\delta:||\delta||\leqslant\epsilon} y(w^T x + b) + yw^T\delta = y(w^T x + b) - \epsilon||yw||_2 = y(w^T x + b) - \epsilon||w||_2$$

(d) (5 points) Using the solution from part c, provide the simplified min-max problem. Is this problem same as solving the nominal classification problem with a regularizer term? If yes, then explain why is this the case, and if not, then explain the differences.

Solution:
The original minimization problem

$$\min_{\theta} \max_{\delta:||\delta||\leqslant\epsilon} \ell(h_\theta(x+\delta), y)$$

can now be written as : $\min_{\theta} L\left(y(w^T x+b) - \epsilon||w||_1\right)$ or $\min_{\theta} L\left(y(w^T x+b) - \epsilon||w||_2\right)$.
This is not same as solving the nominal classification problem with a regularizer term as here the penalty term is inside the loss function, indicating that the penalty term plays a role only for those data-points which are close to the decision boundary.

We now consider a dataset of (x, y) pairs with $d = 1$:

$$\begin{aligned}\mathcal{D} = \{&(-1,-1),(-1,-1),(-1,-1),(-1,-1),(-1,-1),(-1,-1),\\ &(-1,-1),(-1,-1),(-1,-1),(-1,-1),(-1,-1),(-1,-1),\\ &(-1,-1),(-1,-1),(-1,-1),(-1,-1),(-1,-1),(1,-1),(2,1)\}\end{aligned} \quad (3)$$

(e) (2 points) What parameter vector θ minimizes the regular logistic regression objective? If more than one, explain. (You should do this without code).

Solution: Any pair of very large magnitude numbers (w, b) satisfying $-2w < b < -w < 0$; the larger the magnitude, the lower the objective.

(f) (2 points) What parameter vector θ minimizes l2-regularized logistic regression with $\lambda = 10^{-12}$ (i.e. adding λw^2 to the objective)? (This question and the rest of the parts of this question can be done using code.)

Solution: $[40.58, -60.87]$

(g) (2 points) What class probability distribution does logistic regression with these parameters assign to the point $x = 1.51$?

Solution: $P(\text{positive}) = 0.60$

(h) (2 points) What parameter vector minimizes the robust regression objective, if we want to be correct even if the input is perturbed by $\epsilon = 0.48$?

Solution: Any pair of very large magnitude numbers (w, b) satisfying $-2w < b < -w < 0$; the larger the magnitude, the lower the objective.

(i) (2 points) What class probability distribution does logistic regression with these parameters assign to the point $x = 1.51$?

Solution: $P(\text{positive}) = 1$

Learning Theory (10 points)

2. You work for Googolog and plan to train a lot of neural networks!

- You are going to try 10 different architectures
- Each with 10 different learning-rate schedules
- And measure the performance at 100 epochs (each) to avoid overtraining

You have a big pile of data. You'd like to use as much as possible for training, but you know you need to save a held-out set for validation in order to pick the best architecture, learning-rate schedule and number of training epochs.

You would like to be able to predict the risk on unseen points by computing the risk on the validation set, and guarantee that, with probability > 0.99 it is within 0.01 of the true risk. Suppose the risk on any point is bounded in $[0, 1]$.

(a) (8 points) How big does your validation set need to be?

Solution: We would like to consider $10 \times 10 \times 100 = 10000$ different hyperparameter configurations. Denote the risk of a model f evaluated on a point x as $R(x, f)$. Denote the full set of models we evaluate as F.

We would like, with probability 0.99, for the following to hold:

$$\max_{f \in F} |\mathbb{E}_x[R(x, f)] - \frac{1}{N} \sum_i R(x^{(i)}, f)| \leqslant 0.01 \tag{4}$$

where $x^{(i)}$ are validation set points and N is the number of validation set points. Note that by the union bound, we can bound the probability of any model breaking this inequality by the sum of the probabilities of each individual model breaking the inequality:

$$P(\max_{f \in F} |\mathbb{E}_x[R(x,f)] - \frac{1}{N}\sum_i R(x^{(i)}, f)| \geqslant 0.01) \\ \leqslant \sum_f P(|\mathbb{E}_x[R(x,f)] - \frac{1}{N}\sum_i R(x^{(i)}, f)| \geqslant 0.01) \quad (5)$$

By Hoeffding's inequality:

$$P(|\mathbb{E}_x[R(x,f)] - \frac{1}{N}\sum_i R(x^{(i)}, f)| \geqslant 0.01) \leqslant 2e^{-2N(0.01)^2} \quad (6)$$

Applying the union bound:

$$P(\max_{f \in F} |\mathbb{E}_x[R(x,f)] - \frac{1}{N}\sum_i R(x^{(i)}, f)| \geqslant 0.01) \leqslant 20000e^{-2N(0.01)^2} \quad (7)$$

Since we would like the left hand side to be less than $1 - 0.99 = 0.01$, we must have:

$$0.01 = 20000e^{-2N(0.01)^2} \quad (8)$$

or

$$N \approx 72543.2 \quad (9)$$

Thus, 72544 points will suffice.

(b) (2 points) Your friend says that if you increase the training set size, you'll be able to make this bound tighter. Is your friend right or wrong? Explain.

Solution: In general, no. Increasing the size of the training set will prevent overfitting to the training set, but it won't prevent overfitting to the validation set. In order to make the above bound on the validation set risk tighter, we would need to increase the validation set size.

Causality (10 points)

3. One idea from learning causal models (which we may explore later in the class) is that the relationships in the "causal" direction are more generally useful and transferrable, than the other way around. So, for example, if we are interested in the relationship between a disease

and its symptoms (which are caused by the disease), we might prefer to model

$$P(\text{symptoms} \mid \text{disease}) \tag{10}$$

rather than

$$P(\text{disease} \mid \text{symptoms}) \tag{11}$$

But, of course, we are usually called upon to predict the disease from the symptoms, and make a classifier that produces $P(\text{disease} \mid \text{symptoms})$.

So, for example, if we trained a classifier for malaria based on data in a tropical country, we might end up with $P_{\text{tropical}}(\text{malaria} \mid \text{symptoms})$.

(a) (8 points) It doesn't seem good to use this same classifier in the US. If we know the incidence of malaria in our tropical population is 1%, and the incidence in US is 0.001%, how could we make use of our $P_{\text{tropical}}(\text{malaria} \mid \text{symptoms})$ at Mass General?

Solution: This is a case of distribution shift: specifically, label-shift only distribution shift since $P(\text{malaria})$ is different between the US and the tropical region, but we may expect $P(\text{symptoms} \mid \text{malaria})$ to be the same. In this case, we may make the following adjustment (denoting malaria as M and symptoms as S):

$$\begin{aligned} P_{US}(M \mid S) &= \frac{\frac{P_{US}(M)}{P_{\text{tropical}}(M)} P_{\text{tropical}}(M \mid S)}{\frac{P_{US}(M)}{P_{\text{tropical}}(M)} P_{\text{tropical}}(M \mid S) + \frac{1-P_{US}(M)}{1-P_{\text{tropical}}(M)}(1 - P_{\text{tropical}}(M \mid S))} \\ &= \frac{\frac{1}{1000} P_{\text{tropical}}(M \mid S)}{\frac{1}{1000} P_{\text{tropical}}(M \mid S) + \frac{10^5-1}{10^5-10^3}(1 - P_{\text{tropical}}(M \mid S))} \\ &= \frac{99 P_{\text{tropical}}(M \mid S)}{99999 - 99900 P_{\text{tropical}}(M \mid S)} \end{aligned} \tag{12}$$

(b) (2 points) If a patient came in with symptoms that would, using the tropical model, made us predict that they had malaria with probability 0.75, what prediction would this US-adjusted model make?

Solution: That they had malaria with probability approximately .003.

Day-Night (15 points)

4. You are training your robot outside in the daylight and come up with a good classifier for whether the terrain in front of it is grass. Now, you get data from night-time and everything looks different! Luckily, there were some examples in your original training data with substantial shadows, and some gathered at dusk. Letting $D_{\text{day}} = (x_i, y_i)_{i=1}^{n}$ be the first, labeled data set, and $D_{\text{night}} = (x_i)_{i=n+1}^{m}$ be the night-time data that you are going to have to make predictions on, let's explore using importance reweighting to address this problem.

(a) (2 points) We need to begin by training a classifier. What training data would we use (inputs and targets)?

Solution: We can use D_{day}, the only labeled data we have.

(b) (3 points) We would typically use logistic regression for this classifier. Would it work to use a decision tree?

Solution: In general, no, because decision trees don't output probabilities of each class being correct but instead just output the predicted class itself. However, we could use a decision tree if it output estimated probabilities at the leaves.

(c) (3 points) Given the trained classifier, explain what problem we need to solve next. Is there a way of using the weights we learned on the daytime-only data to predict on night-time data, or do we need to retrain? If not, how do we make predictions?

Solution: We can't use the classifier trained on D_{day} directly since the distribution over x is shifted at night (co-variate shift). We have to retrain.

(d) (5 points) Assuming we do retrain, explain a strategy for approximating the importance-reweighted objective, while using a standard pre-existing logistic regression package.

Solution: We can duplicate the points x in D_{day} (by different amounts for each x) until the distribution over x in the new D_{day} is close to the distribution of x in D_{night}. We can then retrain from this augmented data.

(e) (2 points) Now let's consider a fake domain in which $x \in \mathbb{R}^2$, and we have

$$y = f(x_1) \quad \text{if } x_2 = 1 \tag{13}$$

$$y = g(x_1) \quad \text{otherwise} \tag{14}$$

Furthermore, assume that our labeled data set D_1 only has x values in which $x_2 = 1$. Now we have a new set of *unlabeled* examples, D_2, which only has x values in which $x_2 = 0$. Explain what the importance re-weighting method would do in this case.

Solution: Importance re-weighting would not work in this case because there is no overlap between the D_1 and D_2 sets. In other words, there is no way to obtain the distribution of x generating D_2 by re-weighting the points in D_1.

Fairness (20 points)

5. Read the linked Zafar et al. paper, Sections 1–4: `https://www.jmlr.org/papers/volume20/18-262/18-262.pdf`. It's okay to skip parts about SVMs

In the following, we will focus on Overall Misclassification Rate (Equation 3.3)

(a) (1 points) When is $g_\theta(y, x)$ non-zero? (it's defined after Equation 4.4)

Solution: This requires:

$$y d_\theta(x) < 0 \tag{15}$$

In other words, the classifier is incorrect.

(b) (5 points) The paper says (just after Equation 4.4) "*if a decision boundary satisfies Eq. (3.3), then the (empirical) covariance defined above will be (approximately) zero (for a sufficiently large training set)*."

Write a formal version of this statement, prove it, and state any additional assumptions that need to be made for this to hold.

Solution: Equation 3.3 says that the probability of misclassification is independent of z:

$$P(g_\theta(y, x) < 0 \mid z = 1) = P(g_\theta(y, x) < 0 \mid z = 0) \tag{16}$$

In other words, $g_\theta(y, x) < 0$ is independent of z.

Since $g_\theta(y, x)$ is never positive:

$$\mathbb{E}[g_\theta(y, x) \mid z] = P(g_\theta(y, x) < 0 \mid z)\mathbb{E}[g_\theta(y, x) \mid g_\theta(y, x) < 0, z] \tag{17}$$

We assume that the *magnitude* of misclassification is independent of z given that there is a misclassification. Thus,

$$\mathbb{E}[g_\theta(y, x) \mid z] = P(g_\theta(y, x) < 0)\mathbb{E}[g_\theta(y, x) \mid g_\theta(y, x) < 0] = \mathbb{E}[g_\theta(y, x)] \tag{18}$$

is independent of z.

Next, the covariance between z and $g_\theta(y, x)$ is:

$$\begin{aligned}
\mathrm{Cov}(z, g_\theta(y, x)) &= \mathbb{E}[(z - \mathbb{E}[z])(g_\theta(y, x) - \mathbb{E}[g_\theta(y, x)])] \\
&= \mathbb{E}[(z - \mathbb{E}[z])g_\theta(y, x)] \\
&= \mathbb{E}[(z - \mathbb{E}[z])g_\theta(y, x) \mid z = 1]P(z = 1) + \mathbb{E}[(z - \mathbb{E}[z])g_\theta(y, x) \mid z = 0]P(z = 0) \\
&= \mathbb{E}[(1 - P(z = 1))g_\theta(y, x) \mid z = 1]P(z = 1) - \mathbb{E}[P(z = 1)g_\theta(y, x) \mid z = 0]P(z = 0) \\
&= P(z = 0)P(z = 1)[\mathbb{E}[g_\theta(y, x) \mid z = 1] - \mathbb{E}[g_\theta(y, x) \mid z = 0]] \\
&= 0
\end{aligned}$$

(c) (5 points) If we are using a classifier where the signed distance to decision boundary $d_\theta(x) = \frac{\theta^T x}{||\theta||}$, what is a value of c in Equation 4.5 that allows a gap of ϵ in the error rate between the two groups ($z = 1$ and $z = 0$)? Assume that the groups have equal proportions. Also, assume that all misclassifications are exactly distance 1 from the decision boundary (do not use any additional assumptions you made in part b).

Solution: First, we assume without loss of generality that $z = 1$ has higher misclassification rate by ϵ:

$$P(y\frac{\theta^T x}{\parallel \theta \parallel} < 0 \mid z = 1) = P(y\frac{\theta^T x}{\parallel \theta \parallel} < 0 \mid z = 0) + \epsilon \tag{19}$$

Using the misclassification distance from the decision boundary:

$$\mathbb{E}[\min(0, y\frac{\theta^T x}{\parallel \theta \parallel}) \mid z] = -P(y\frac{\theta^T x}{\parallel \theta \parallel} < 0 \mid z) \tag{20}$$

Applying the earlier equation:

$$\mathbb{E}[\min(0, y\frac{\theta^T x}{\parallel \theta \parallel}) \mid z = 1] = \mathbb{E}[\min(0, y\frac{\theta^T x}{\parallel \theta \parallel}) \mid z = 0] - 4\epsilon \tag{21}$$

Equation 4.5 bounds the covariance between z and $g_\theta(y, x)$ to have maximum magnitude c. In this case,

$$g_\theta(y, x) = \min(0, y\frac{\theta^T x}{\parallel \theta \parallel}) \tag{22}$$

The absolute value of the covariance between this and z is:

$$\begin{aligned} |\mathbb{E}[(z - \mathbb{E}[z]) \min(0, y\frac{\theta^T x}{\|\theta\|})]| &= \frac{1}{4}|\mathbb{E}[\min(0, y\frac{\theta^T x}{\|\theta\|}) \mid z = 1] - \mathbb{E}[\min(0, y\frac{\theta^T x}{\|\theta\|}) \mid z = 0]| \\ &= \frac{\epsilon}{4} \end{aligned}$$

Thus, $c = \frac{\epsilon}{4}$ allows a gap of ϵ in error rate.

(d) (2 points) Why are there no z values in Equation 4.9?

Solution: The dependence on z is in the selection of datasets $\mathcal{D}_i^-$.

(e) (2 points) If we optimize Equation 4.13 to obtain classifier θ, will we need z values at prediction time? Why would it be better if we did not?

Solution: No, we will not need z at prediction time; the prediction only depends on x. Since our goal is to produce a fair classifier after training, if we have succeeded, then we should not need to use z so that our predicted output does not unnecessarily depend on z.

(f) (5 points) Suppose that after running extensive experiments, we find that if we train a classifier on a population of n_0 examples with $z = 0$ and n_1 examples with $z = 1$, the classification error rate ($P(y \neq \hat{y})$) on unseen points with $z = 0$ will be $\frac{c}{n_0}$ and the error rate on unseen points with $z = 1$ will be $\frac{c}{n_1}$. We know that the underlying data distribution has $z = 0$ and $z = 1$ in equal proportion.

Unfortunately, we have lost our labels for which points are in group $z = 0$ and $z = 1$. Suppose we wish to guarantee (with probability 99%) our classifier will be fair on test points up to some small error ϵ:

$$|P(y \neq \hat{y} \mid z = 0) - P(y \neq \hat{y} \mid z = 1)| \leqslant \epsilon \tag{23}$$

Suppose we draw a training set of n pairs (x, y) to train our classifier. We can show that in order to guarantee the above property, the number of training points must scale as ϵ^{-p}. What is p?

Solution: Since the two groups $z = 1$ and $z = 0$ are in equal proportion, we can expect on average that there will be $\frac{n}{2}$ points in each group. However, there will be some variation in the exact number. Specifically,

$$Var(n_1) = \sum_i Var(\mathbf{1}(z^{(i)} = 1)) = O(n) \tag{24}$$

where $z^{(i)}$ denotes the group of point i. Thus, we may expect n_1 and n_0 to be roughly $\frac{n}{2} \pm k\sqrt{n}$ for some constant k (with high probability).
This means that the error on group $z = 1$ is with high probability in:

$$[\frac{c}{\frac{n}{2} + k\sqrt{n}}, \frac{c}{\frac{n}{2} - k\sqrt{n}}] \tag{25}$$

with the same bounds applying for $z = 0$. We would like the difference in error to be on the order of ϵ:

$$\frac{c}{\frac{n}{2} - k\sqrt{n}} - \frac{c}{\frac{n}{2} + k\sqrt{n}} = \epsilon \tag{26}$$

Simplifying,

$$\frac{2k\sqrt{n}}{\frac{n^2}{4} - k^2 n} = \frac{\epsilon}{c} \tag{27}$$

Assuming n is large, we may approximate:

$$\frac{2k\sqrt{n}}{\frac{n^2}{4}} = \frac{\epsilon}{c} \tag{28}$$

Thus,

$$n^{-\frac{3}{2}} = \frac{\epsilon}{8kc} \tag{29}$$

or

$$n = O(\epsilon^{-\frac{2}{3}}) \tag{30}$$

$p = \frac{2}{3}$.