CS 189:289A Introduction to Machine Learning
Fall 2021 Jennifer Listgarten and Jitendra Malik

HW7

Due 12/7 at 11:59pm

- We prefer that you typeset your answers using LaTeX or other word processing software. If you haven't yet learned LaTeX, one of the crown jewels of computer science, now is a good time! Neatly handwritten and scanned solutions will also be accepted for the written questions.
- In all of the questions, **show your work**, not just the final answer.

Deliverables:

1. Submit a PDF of your homework to the Gradescope assignment entitled "HW7 Write-Up". **Please start each question on a new page.**
   - In your write-up, please state with whom you worked on the homework. This should be on its own page and should be the first page that you submit.
   - In your write-up, please copy the following statement and sign your signature next to it. (Mac Preview and FoxIt PDF Reader, among others, have tools to let you sign a PDF file.) We want to make it *extra* clear so that no one inadvertently cheats. *"I certify that all solutions are entirely in my own words and that I have not looked at another student's solutions. I have given credit to all external sources I consulted."*

1 "Sample complexity" of coupon collecting

Willy Wonka has a chocolate factory where the produced chocolates have d different types of cards hidden under the chocolate wrappers. To encourage people to consume these chocolates, a game is announced: Willy Wonka will draw a card type uniformly at random after n days and if someone has that card in their collection, they will be allowed to enter Willy Wonka's factory and participate in a questionable experiment.

Charlie is a consumer of Willy's chocolates and he visits a particular local store every day to buy his chocolates. The store contains equal number of chocolates of each card type at any given time. Every time a chocolate with a particular card hidden beneath the wrapper is bought, another chocolate containing an identical card is put in its place immediately. Whenever Charlie buys a chocolate, he does that by picking up a chocolate uniformly at random from the store.

(a) If Charlie wants his probability of winning to be at least $1 - \delta$, how many *distinct* card types should he have in his collection before Willy Wonka draws the card?

Solution: Say Charlie has c distinct card types. Willy Wonka then picks one of Charlie's card types with probability $\frac{c}{d}$. To ensure that this probability exceeds $1 - \delta$, we observe that

$$\frac{c}{d} \geq 1 - \delta \implies c \geq d(1 - \delta),$$

and hence Charlie should have at least $d(1 - \delta)$ types of distinct cards to ensure that his probability of winning is at least $1 - \delta$.

(b) Suppose that Charlie visited the particular local store all n days and bought 1 chocolate at random each day before Willy's draw of the random card. What is the probability that Charlie wins a prize from Willy's draw?

Solution: Suppose that Willy Wonka picked the i-th card. Charlie *loses* the prize if all his n cards are from the $d - 1$ cards that Willy Wonka did not pick. This happens with probability $(\frac{d-1}{d})^n$.

Since the choice of Willy Wonka's card does not affect this probability, the probability of losing is $(\frac{d-1}{d})^n$. Thus

$$\mathbb{P}(\text{win}) = 1 - \mathbb{P}(\text{lose}) = 1 - \left(\frac{d-1}{d}\right)^n.$$

(c) Now assume $n = \alpha d$. What does Charlie's probability of winning converge to as d gets large?

Hint: you may make use of the fact that

$$\lim_{d\to\infty} \left(1 + \frac{x}{d}\right)^d = e^x.$$

Solution: We can approximate the probability of winning for large d and $n = \alpha d$ as follows:

$$1 - \left(\frac{d-1}{d}\right)^{\alpha d} = 1 - \left(1 - \frac{1}{d}\right)^{\alpha d} = 1 - \left(1 - \frac{\alpha}{\alpha d}\right)^{\alpha d} \approx 1 - (e^{-\alpha})$$

(d) Now, consider the following function estimation problem. We want to learn a completely unstructured function f on a finite domain $\mathcal{D}$ of size (not dimensionality) d. We collect a training data of size n from random samples, i.e., we have the dataset $\{(x_i, f(x_i)), i = 1, \ldots, n\}$ where each x_i is drawn uniformly at random from $\mathcal{D}$. How big of a training set do we need to collect to ensure that with probability at least $1 - \delta$, we will successfully estimate the function at a point which is drawn uniformly at random from the domain $\mathcal{D}$? What happens as d gets large?

Hint: you may make use of the approximation $\ln(1/(1 - a)) \sim a$ for small a.

Solution: Note that since the function is completely unstructured, we can successfully estimate the function at a random point only if we know the function value at that point. Hence, correct estimation of f when presented with a random point is possible only if we have already observed the function value at that point in our training set. This observation reveals that this problem is equivalent to the game described in part (b). In particular, successful estimation of f at a random point is equivalent to Charlie having a card when Willy Wonka picks a random prize-winner card.

As a result, the probability of getting the function value at a random point correct after collecting n training data points is equal to the probability of Charlie winning the game after buying n chocolates. This probability was computed in part (c) as $1 - \left(\frac{d-1}{d}\right)^n$.

We need this probability to exceed $1 - \delta$, so we have

$$1 - \left(\frac{d-1}{d}\right)^n > 1 - \delta \implies \left(\frac{d}{d-1}\right)^n > \frac{1}{\delta} \implies n \geq \frac{\ln(1/\delta)}{\ln\left(1/(1 - \frac{1}{d})\right)}.$$

For the case when d gets large, we use the approximation above. Substituting $x = 1/d$, we find that

$$n \gtrapprox d \ln(1/\delta)$$

suffices for this case.

2 Correlated features in ridge regularization

In this problem, we briefly explore what happens to ridge regularization when there are correlated features. For simplicity throughout this problem, we are going to assume that the true pattern is a proportion w^* of the first feature $x[1]$ in each sample. So $y_i = w^* \mathbf{x}_i[1]$ when we have n training data points $\{(\mathbf{x}_i, y_i)\}_1^n$.

Suppose that there were only two features, and the first and second features are just multiples of each other, that is, $x[2] = \beta x[1]$ where β is some nonzero constant. In this case, the second feature is a perfect "alias" of the first feature for the training data.

Recall the ridge regression objective:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 .$$

Compute $\hat{\mathbf{w}}$ as a function of the $\mathbf{x}_i[1]$s, β, λ, and w^*. Comment on what the solution reduces to for $\beta \to \infty$, $\beta = 0$, and $\lambda \to \infty$, and explain these special cases.

Solution: We know that the solution $\hat{\mathbf{w}}$ takes the following form:

$$\hat{\mathbf{w}} = (\mathbf{X}^\top \mathbf{X} + \lambda I)^{-1} \mathbf{X}^\top \mathbf{y} .$$

Let $\Sigma^2 = \sum_{i=1}^n \mathbf{x}_i[1]^2$ Note that $\mathbf{X}^\top \mathbf{X}$ is a 2×2 matrix with the following entries:

$$\mathbf{X}^\top \mathbf{X} = \begin{bmatrix} \Sigma^2 & \beta\Sigma^2 \\ \beta\Sigma^2 & \beta^2\Sigma^2 \end{bmatrix} .$$

We can also see that $\mathbf{X}^\top \mathbf{y}$ is a 2×1 vector with the following entries:

$$\mathbf{X}^\top \mathbf{y} = \begin{bmatrix} w^*\Sigma^2 \\ w^*\beta\Sigma^2 \end{bmatrix} .$$

Now, we can proceed with computing $\hat{\mathbf{w}}$:

$$\begin{aligned}
\hat{\mathbf{w}} &= \begin{bmatrix} \Sigma^2 + \lambda & \beta\Sigma^2 \\ \beta\Sigma^2 & \beta^2\Sigma^2 + \lambda \end{bmatrix}^{-1} \begin{bmatrix} w^*\Sigma^2 \\ w^*\beta\Sigma^2 \end{bmatrix} \\
&= \frac{1}{\Sigma^2\lambda + \beta^2\Sigma^2\lambda + \lambda^2} \begin{bmatrix} \beta^2\Sigma^2 + \lambda & -\beta\Sigma^2 \\ -\beta\Sigma^2 & \Sigma^2 + \lambda \end{bmatrix} \begin{bmatrix} w^*\Sigma^2 \\ w^*\beta\Sigma^2 \end{bmatrix} \\
&= \frac{1}{\Sigma^2\lambda + \beta^2\Sigma^2\lambda + \lambda^2} \begin{bmatrix} \lambda w^*\Sigma^2 \\ \lambda w^*\beta\Sigma^2 \end{bmatrix} \\
&= \frac{w^*\Sigma^2}{\Sigma^2 + \beta^2\Sigma^2 + \lambda} \begin{bmatrix} 1 \\ \beta \end{bmatrix} .
\end{aligned}$$

As $\beta \to \infty$, both entries in $\hat{\mathbf{w}}$ will go to zero, though the second entry goes to zero asymptotically slower. Thus, as β grows, more of the relative weight is assigned to the second feature compared to the first feature. For $\beta = 0$, the weight corresponding to the second feature is 0, and the weight corresponding to the first feature is:

$$\frac{w^*\Sigma^2}{\Sigma^2 + \lambda} ,$$

Which is the standard "shrinkage" we see in ridge regression. Note that if both $\beta = 0$ and $\lambda = 0$, we recover the true weight vector

$$\begin{bmatrix} w^* \\ 0 \end{bmatrix}.$$

For $\lambda \to \infty$, both entries in $\hat{\mathbf{w}}$ go to zero.

3 Clip Loss

You have seen examples of different loss functions like the squared-error loss and the hinge-loss. This question explores a different loss function.

Let $S = \{(\mathbf{x}_1, y_1), \ldots (\mathbf{x}_n, y_n)\}$ be a set of n points sampled i.i.d. from a distribution $\mathcal{D}$. This is the training set with $\mathbf{x}_i \in \mathbb{R}^d$ being the features and $y_i \in \{-1, 1\}$ being the labels.

We are thinking about a linear classifier that is going to look at the sign of $\mathbf{w}^\top \mathbf{x}$ to make a decision as to whether the label is $+1$ or -1.

Define the *clip loss* of a linear classifier $\mathbf{w} \in \mathbb{R}^d$ as

$$\text{loss}(\mathbf{w}^\top \mathbf{x}, y) = \text{clip}(y\mathbf{w}^\top \mathbf{x})$$

Where clip is the function

$$\text{clip}(z) = \begin{cases} 1 & \text{if } z < 0 \\ 0 & \text{if } z \geq 1 \\ 1 - z & \text{otherwise.} \end{cases}$$

For any d-dimensional vector $\mathbf{w}$, define the *risk* of $\mathbf{w}$ as

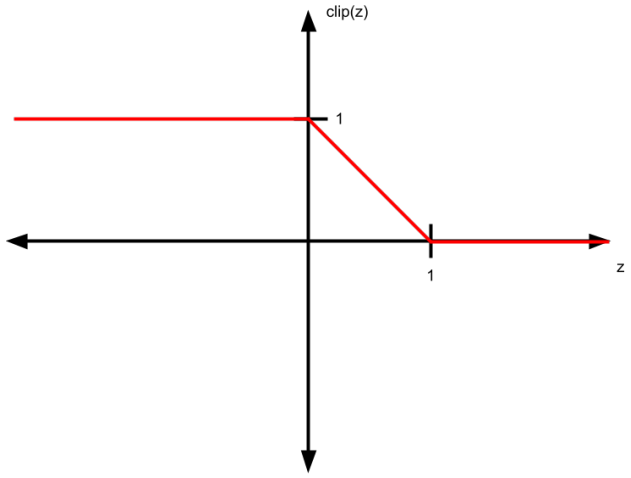
$$R[\mathbf{w}] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\text{loss}(\mathbf{w}^\top \mathbf{x}, y)]\,,$$

and the *empirical risk* of $\mathbf{w}$ as

$$R_S[\mathbf{w}] = \frac{1}{n}\sum_{i=1}^{n} \text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i)\,.$$

(a) Draw the clip loss function. Is the function clip convex? Explain your answer.

Solution:



It is not convex. Drawing the function shows that the line segment from $(-1, 1)$ to $(1, 0)$ lies below the graph of the clip function, but extending out this line further then lies above the graph.

(b) Prove that if $R_S[\mathbf{w}] = 0$ and $\|\mathbf{w}\|_2 = 1$, then the hyperplane defined by $\mathbf{w}$ has a classification margin ≥ 1 on this training set.

Solution: The margin of the normalized hyperplane is defined as

$$\min_{1 \leq i \leq n} y_i(\mathbf{w}^\top \mathbf{x}) \,.$$

If $R_S[\mathbf{w}] = 0$, then since $\text{clip}(z) \geq 0$ this quantity is greater than or equal to 1 for all $1 \leq i \leq n$.

(c) Prove that $\mathbb{E}_S[R_S[\mathbf{w}]] = R[\mathbf{w}]$. Here, the outer expectation is being taken over the randomly drawn training set.

Solution:

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i)\right] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[\text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i)\right] = \frac{1}{n}\sum_{i=1}^{n} R[\mathbf{w}] = R[\mathbf{w}]$$

(d) Prove that $\text{Var}(R_S[\mathbf{w}]) \leq \frac{1}{n}$.

Solution:

$$\begin{aligned}
\text{Var}(R_S[\mathbf{w}]) &= \mathbb{E}\left[(R_S[\mathbf{w}] - R[\mathbf{w}])^2\right] \\
&= \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{E}(\text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i) - R[\mathbf{w}])(\text{loss}(\mathbf{w}^\top \mathbf{x}_j, y_j) - R[\mathbf{w}]) \\
&= \frac{1}{n^2}\sum_{i=1}^{n} \mathbb{E}\left[(\text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i) - R[\mathbf{w}])^2\right] \\
&= \frac{1}{n}\mathbb{E}\left[(\text{loss}(\mathbf{w}^\top \mathbf{x}, y) - R[\mathbf{w}])^2\right] \\
&\leq \frac{1}{n}
\end{aligned}$$

Here, the first line is the definition of variance and part (c), the second line expands the square, the third line follows because $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ are independent. The fourth line follows because the $(\mathbf{x}_i, y_i)$ are identically distributed. The last line follows because the clip loss is nonnegative and bounded above by 1.

Alternate proof of first 4 steps:

$$\begin{aligned}
\text{Var}(R_S[\mathbf{w}]) &= \text{Var}(\frac{1}{n}\sum_{i=1}^{n} \text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i)) \\
&= \frac{1}{n^2}\,\text{Var}(\sum_{i=1}^{n} \text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i)) \\
&= \frac{1}{n^2}\sum_{i=1}^{n} \text{Var}(\text{loss}(\mathbf{w}^\top \mathbf{x}_i, y_i), \text{ by i.i.d} \\
&= \frac{1}{n}\,\text{Var}(\text{loss}(\mathbf{w}^\top \mathbf{x}, y))
\end{aligned}$$

(e) Is it possible to have an S and $\mathbf{w}$ such that $R_S[\mathbf{w}] = 0$, but $R[\mathbf{w}] > 0$? Explain your answer.

Solution: Yes. Consider the case when $n = 1$. Then it is possible to classify the single data point correctly while classifying all of the opposite class incorrectly.