DS-GA 1003 Machine Learning: Homework 2

Due 11.59 p.m. EST, March 26, 2024 on Gradescope

(fill in your name here)
(collaborators if any)

We encourage LaTeX-typeset submissions but will accept quality scans of hand-written pages.

1 Very Random Forest

Consider building a random forest by both subsampling the data and choosing a single feature per tree randomly. For example, consider a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$ for $i = 1, .., N$. We will carry out the following procedure:

1. Randomly sample one feature index $j \in \{1, \ldots, D\}$.
2. Draw a sample of the data $\mathcal{D}_{\mathbf{k}}$ of size $M \leq N$ with replacement. These datapoints will have indices $\mathbf{k} = k_1, \ldots, k_M$.
3. Keep only the j^{th} feature of the M samples: i.e. letting $x_{i,j}$ be the i^{th} datapoint and j^{th} feature, we use data
$$\mathcal{D}^j_{\mathbf{k}} = \{(x_{(k_1,j)}, y_{(k_1)}), ..., (x_{(k_M,j)}, y_{(k_M)})\}$$
4. Build a decision tree on $\mathcal{D}^j_{\mathbf{k}}$.
5. Repeat the above process R times so that r^{th} tree T_r uses feature $j^{(r)}$ and data $\mathbf{k}^{(r)}$ for $r \in \{1, \ldots, R\}$. Using this notation, the prediction of the r^{th} tree on new input $\mathbf{x}^\star$ is $T_r(\mathbf{x}^\star; \mathcal{D}^{j^{(r)}}_{\mathbf{k}^{(r)}})$.
6. Average these random trees to construct the random forest. That is, for input $\mathbf{x}^\star$, the random forest predicts $\hat{y} = \frac{1}{R}\sum_{r=1}^{R} T_r(\mathbf{x}^\star; \mathcal{D}^{j^{(r)}}_{\mathbf{k}^{(r)}})$.

Let us call this model a Very Random Forest (VRF). In this question, we will characterize the *bias* of an VRF (in the context of bias-variance tradeoff).

(A) Write down the two term that have to be equal for a model to be unbiased. One term should be some statistic of the true data distribution and the other should be a statistic of the model output.

Solution. Write your solution for each question using the `solution` environment. Feel free to use style packages to your convenience, e.g. highlighting parts of your solution that you still need to work on. □

(B) Let us first consider a single decision tree $T(\mathbf{x}, \mathcal{D}^j)$ that uses only a single feature j but the **entire** dataset (i.e. all N data points). Assuming that the tree was trained by minimizing squared loss, what will be the tree's prediction for a test point $\mathbf{x}^\star$? I.e. write down the function that $T(\mathbf{x}^\star, \mathcal{D}^j)$ corresponds to.

(C) Now, express the model-dependent term that you wrote in part (A) in terms of your answer in (B).

(D) As $N \to \infty$, what data-dependent function does your answer in part (C) converge to? You may assume that your VRF has sufficient capacity to model this function arbitrarily well. Describe in words the source of bias.

(E) Let us build another VRF. Each tree in this forest is built by randomly sampling *two different features* instead of one in step 1. As $N \to \infty$, what data-dependent function does your answer in part (C) converge to? How does using two different features instead of one reduce the bias of VRF?

(F) Compare the bias and variance of the VRF's with the traditional random forest, where we select a random subset of the data and a random subset of features to build each tree.

Hint: Look at the generalization bound from the lecture on random forests. You only need to look at the final result, not the derivation.

2 Conditional Modelling with Gaussians

For any finite set of points $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, assume the following conditional model:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} \Bigg| \, \mathbf{x}_1, ..., \mathbf{x}_N \sim \mathcal{N}(\vec{0}, K)$$

Here, K is a covariance matrix such that $K_{ij} := k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_j\|^2)$. You may assume that K is positive-definite for any dataset $\mathcal{D}$.

(A) For any set of $(N+1)$ points $\mathbf{x}_1, \ldots, \mathbf{x}_{N+1}$, write the down the distribution of $y_1, \cdots y_N, y_{N+1} \mid \mathbf{x}_1, \cdots \mathbf{x}_N, \mathbf{x}_{N+1}$ under the assumed model.

(B) Given a dataset $\mathcal{D} = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$ and a test point $\mathbf{x}_{N+1}$, how do you make a prediction using this model?

(C) Suppose that the test point $\mathbf{x}_{N+1}$ is an outlier, e.g. assume $\min_{1 \le i \le N} \|\mathbf{x}_{N+1} - \mathbf{x}_i\| > 1000$. Using your answer in part (B), what is your prediction for $\mathbf{x}_{N+1}$?

(D) Now, imagine you fit a flexible neural network f_θ on the same dataset $\mathcal{D}$. Consider the same outlier $\mathbf{x}_{N+1}$ as in part (C). What can you say about the prediction $f_\theta(\mathbf{x}_{N+1})$? Does your answer change depending on your choice of network f_θ?

Hint: Think about the constraints a neural network may impose on the prediction for $\mathbf{x}_{N+1}$.

(E) Compare the predictions you made in parts (C) and (D). Are they the same? If not, explain the difference. If one has a problem, suggest a way to solve it.

3 Neural Networks for Reconstruction

In class, we saw how neural networks are a flexible model class that can be used for supervised regression or classification tasks, i.e. predicting y from $\mathbf{x}$. Let us consider a neural network that tries to predict $\mathbf{x}$ from $\mathbf{x}$ instead. In other words, given an input $\mathbf{x} \in \mathbb{R}^D$, the neural network will be trained to reconstruct the same $\mathbf{x}$.

Let us consider the network's architecture as follows:

$$\mathbf{h} = a(\mathbf{V}\mathbf{x})$$
$$\hat{\mathbf{x}} = \mathbf{W}\mathbf{h}$$

Here, our network contains a single hidden layer of size H, hence $\mathbf{V}$ is a $H \times D$ matrix and $\mathbf{W}$ is a $D \times H$ matrix. $a(\cdot)$ is an element-wise non-linear activation function that we will leave undecided for now. We have omitted bias terms to simplify the math.

I Gradients

(A) Write down the loss function L of our network for a single data point $\mathbf{x}$, assuming element-wise squared loss. Compute the derivative $\frac{\mathrm{d}L}{\mathrm{d}\mathbf{h}}$. Do not leave any symbolic expressions of the form $\frac{dy}{dx}$, i.e. actually compute the derivative in terms of known quantities like $\mathbf{W}$ or $\mathbf{x}$.

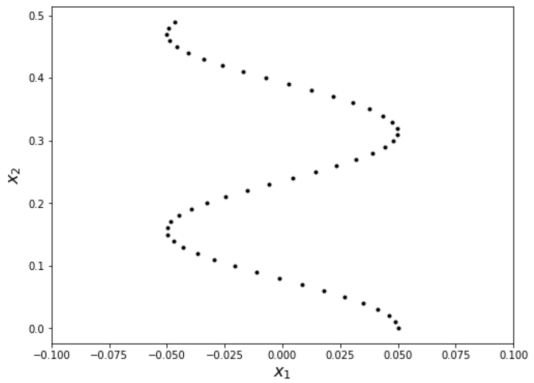
(B) Let $v := \mathbf{V}_{1,1}$ be the element in the first row and first column of $\mathbf{V}$. Compute the derivative $\frac{\mathrm{d}\mathbf{h}}{\mathrm{d}v}$. You can assume that $\frac{\mathrm{d}a(x)}{\mathrm{d}x} = a'(x)$. Next, using the chain rule and your answer in (A), compute the derivative $\frac{\mathrm{d}L}{\mathrm{d}v}$. As in (A), do not leave any symbolic expressions of the form $\frac{dy}{dx}$.

II Representation

(C) How does our choice of H affect the neural network's performance? Assuming that our optimization procedure successfully finds a network with minimum loss, describe the characteristics of an optimal network in the cases: (i) $H > D$, (ii) $H = D$, and (iii) $H < D$.

In each of these cases, is zero loss always possible? If not, why not? If sometimes, give examples of data distributions where zero loss is or is not possible.

(D) How does our choice of a affect our neural network's performance? Consider two activation functions: (i) $a(x) = \frac{1}{1+\epsilon^{-x}}$ (the sigmoid function), and (ii) $a(x) = x$ (the identity function). For the 2D input data described by the plot below, what might the hidden representation $\mathbf{h}$ look like for each of these activation functions? Why?



4 Building a Latent Variable Model

In this question, you will build a latent variable model by making certain assumptions about your data. Now, you have data about N patients that were treated with a new drug to control their blood cholesterol (BC) level. For each patient $1 \leq i \leq N$ in the dataset, you observe a single scalar change in BC and no other data; let us denote this value as x_i. A biologist collaborator tells you that there potentially exists K patient characteristics that affect how BC changes due to drug intake. Since these factors are unmeasured, you build a model with latent variables $z_{i,k}$ that indicate the presence of characteristic k for patient i.

Denote the characteristic vector for patient i as $\mathbf{z}_i = [z_{i,1}, \cdots, z_{i,K}]$ where $z_{i,k} \in \{0, 1\}$ (i.e. assume that the latent variables are binary). Furthermore, assume that $(x_i, \mathbf{z}_i) \perp\!\!\!\perp (x_j, \mathbf{z}_j)$ for $i \neq j$.

(A) Choose a prior distribution for the characteristic vectors $\mathbf{z}_i \sim p(\mathbf{z})$ and justify your choice, i.e. describe the assumptions that you made.

(B) The biologist tells you that given any patient's characteristics $\mathbf{z}_i$, the observed BC can be modeled as:

$$v_{i,k} \sim \mathcal{N}(\mu_k, 1), \qquad \epsilon_i \sim \mathcal{N}(0, 1), \quad x_i = \sum_{k=1}^{K} z_{i,k} v_{i,k} + \epsilon_i$$

where $\{\mu_k\}_{1 \leq k \leq K}$ are known quantities. Write down the the likelihood distribution $p(x_i \mid \mathbf{z}_i)$ explicitly.

(C) Under your choice of prior in part (A) and the likelihood specified by the biologist in part (B), write down the implied marginal distribution $p(x_i)$ in terms of known quantities like $\{\mu_k\}_{1 \leq k \leq K}$. How many modes can this distribution have?

(D) How would you infer $\mathbf{z}_i$ for any patient i? Write down one way to do it in terms of observed data $\{x_i\}_{1 \leq i \leq N}$ and known quantities $\{\mu_k\}_{1 \leq k \leq K}$.

(E) Now consider a different likelihood model than in (B). Instead of giving you exact values, suppose that the biologist now tells you a range for each μ_k in $\{\mu_k\}_{1 \leq k \leq K}$. What prior would you place on these means and why?

(F) Now consider yet another likelihood model, where we let each possible value of $\mathbf{z}_i$ parameterize the likelihood with a separate mean $\mu_{\mathbf{z}_i} = \mathbb{E}[x_i | \mathbf{z}_i]$? What are the trade-offs of this model compared to our original one in (B)? For example, what happens when K is large?

5 Bias-Variance Tradeoff and Regularization

In this question, you will study the bias and variance of the linear regression estimator with and without ℓ_2 regularization. The input points $\mathbf{x}_i$ are P dimensional and are sampled i.i.d. from $\mathcal{N}(0, I_P)$. The targets are given by $y_i = \mathbf{x}_i^T \boldsymbol{\theta}^* + \epsilon_i$ where the noise ϵ_i are sampled from $\mathcal{N}(0, \sigma^2)$. We have access to $N = P + 1$ data points.

(A) We are given an estimator $\boldsymbol{\theta}$. Express the *risk* $R(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}\sim\mathcal{N}(0,I_P),\ y|\mathbf{x}\sim\mathcal{N}(\mathbf{x}^T\boldsymbol{\theta}^*,\sigma^2)}[(\mathbf{x}^T\boldsymbol{\theta} - y)^2]$ in terms of bias and variance of the estimator and the noise terms. Indicate what each term corresponds to.

(B) Use a ridge parameter $\lambda \geq 0$, i.e. the coefficient of ℓ_2 regularization. Express the estimator $\boldsymbol{\theta}$ learned with ridge regression in terms of the data matrix $X = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{P\times N}$, the noise vector $E = [\epsilon_1, ..., \epsilon_N] \in \mathbb{R}^N$, the true parameter $\boldsymbol{\theta}^*$, and the ridge λ.

(C) Give the formulae of the bias $\mathbb{E}_{X,E}[\boldsymbol{\theta}] - \boldsymbol{\theta}^*$ and variance $\mathbb{E}_{X,E}[\boldsymbol{\theta}^2] - \mathbb{E}_{X,E}[\boldsymbol{\theta}]^2$ of the rigde regression estimator $\boldsymbol{\theta}$. The formulae should be expressed in terms of the ridge λ and as an expectation over X.

(D) Set $\lambda = 0$. In this ridgeless case, is the estimator unbiased? Express the variance in terms of the eigenvalues of the matrix (XX^T) and comment on its behavior when the number of features is large.

Hint: Marchenko-Pastur distribution describes the eigenvalues of XX^T when P is large. Commenting on the behavior of the minimum eigenvalue will suffice to explain the behavior of the variance.

(E) To mitigate the large variance, we want to use a non-zero ridge. In this ridge case, is the estimator unbiased? Use the formulae of the variance to explain how the ridge parameter mitigates the large variance.

6 Neural Networks and Overparameterization

In class, we learned that having more parameters than needed to perfectly learn a training dataset $(\mathbf{x}_i, y_i)$ for $i = 1, ..., N$ facilitates the optimization problem. Consider the dataset $(\mathbf{x}_1 = (-1, 1), y_1 = 0), (\mathbf{x}_2 = (1, -1), y_2 = 0), (\mathbf{x}_3 = (1, 1), y_3 = 1), (\mathbf{x}_4 = (-1, -1), y_4 = 1)$.

(A) Construct parameters of a neural network with two neurons and ReLU link function $\mathbf{x} \to \max(\boldsymbol{\theta}_1^T \mathbf{x} + b_1, 0) + \max(\boldsymbol{\theta}_2^T \mathbf{x} + b_2, 0)$ that achieves perfect accuracy on the training data. You can either write out the parameters $(\boldsymbol{\theta}_i, b_i)$ explicitly, or draw the hyperplanes $\boldsymbol{\theta}_i^T \mathbf{x} + b_i = 0$ corresponding to each neuron in the input space. Explain how your neural network achieves perfect accuracy.

Use parameters you constructed in (A) to generate new parameters in the overparameterized neural network with $H \geq 2$ neurons by splitting neurons

$$\max(\boldsymbol{\theta}_i^T \mathbf{x} + b_i, 0) \to \max(\alpha \boldsymbol{\theta}_i^T \mathbf{x} + \alpha b_i, 0) + \max((1 - \alpha) \boldsymbol{\theta}_i^T \mathbf{x} + (1 - \alpha) b_i, 0)$$

for some $\alpha \in [0, 1]$. The new parameter vector after neuron splitting generates the same neural network function. Substitute parameters of each one of H neurons by using either one of the two neurons' parameters. Consider fixed α's after splitting neurons.

(B) How many such parameter vectors are there? The collection of neural network parameters is a $3H$ dimensional vector. How does the number of the parameter vectors scale compare to the number of the parameters when H is large?

(C) Using your answer in part (B), compare the cases of no overparameterization ($H = 2$) vs infinite overparameterization ($H \to \infty$) and comment on how overparameterization facilitates the optimization problem.