Homework 1

36-708, Spring 2022

Due Friday February 11 at 5PM EST

- Please attach all code to your homework. In an RMarkdown document for example, this can be done in one line (see Yihui Xie's website for how to do this).
- Please either type up your solutions or use neat handwriting. Solutions that cannot be read easily by the TA will not be graded.

1 k-nearest-neighbor regression error

In this question, we derive bounds on the squared risk of k-nearest-neighbor regressor, when the regression function is assumed to be Lipschitz.

1.1 Deriving optimality of the regression function

To get started, we first establish optimality of the regression function as a regressor. Consider the following setup.

- Data model: $(X, Y) \sim \mathbb{P}_{X,Y}$ is a distribution on $\mathbb{R}^d \times \mathbb{R}$.
- Quantity of interest: any regressor $f : \mathbb{R}^d \to \mathbb{R}$ versus the regression function,

$$f^\star(x) := \mathbb{E}\left[Y \mid X = x\right].$$

- Metric: squared risk of f defined as

$$\mathcal{R}(f) := \mathbb{E}\left[(Y - f(X))^2\right].$$

(a) Show that the squared risk of the regression function $f^\star$ is smaller than any other regressor. In other words, show that

$$\mathcal{R}(f^\star) \leq \mathcal{R}(f).$$

1.2 Expressing excess risk as a bias-variance decomposition

We now express the excess risk of the k-nearest-neighbor regressor in terms of a bias-variance decomposition. In this example, we consider a version under fixed design (non-random X design) as follows.

- Data model: Treat $x_1, \ldots x_n$ as fixed points in $\mathbb{R}^d$ and let $Y_i \sim \mathbb{P}$ with mean $\mathbb{E}(Y_i) \equiv f^\star(x_i)$ for each $i \in \{1, \ldots, n\}$ and variance $\mathbb{V}(Y_i) \equiv \sigma^2$. Alternatively, we can think of $X_1, \ldots, X_n$ as random but we condition on them.
- Quantity of interest: k-nearest-neighbor estimate defined as $\widehat{f}(x) := \frac{1}{k}\sum_{i \in \mathcal{N}_k(x)} Y_i$ where $\mathcal{N}_k(x)$ is the set of indices of the k closest points to x (here, 'closeness' is defined with respect to the Euclidean norm).

- Metric: first, define the point-wise squared risk of $\widehat{f}$ defined as

  $$\mathcal{R}(\widehat{f}, x) := \mathbb{E}\left[(Y - \widehat{f}(x))^2\right],$$

  where $x \in \mathbb{R}^d$ and $Y \sim \mathbb{P}$ sampled *independently of* $Y_1, Y_2, \ldots, Y_n$. Then, the point-wise excess squared risk is defined as

  $$\mathcal{R}(\widehat{f}, x) - \mathcal{R}(f^\star, x).$$

(a) Establish a bias-variance decomposition for the point-wise excess risk:

$$\mathcal{R}(\widehat{f}, x) - \mathcal{R}(f^\star, x) = b(x)^2 + v(x),$$

where the squared bias $b(x)^2 = \left[\frac{1}{k}\sum_{i \in \mathcal{N}_k(x)}(f^\star(x_i) - f^\star(x))\right]^2$ and the variance $v(x) = \frac{\sigma^2}{k}$.

1.3 Bounding the bias with additional assumptions

To help bound the bias term, we make the following assumptions on the input points and the regression function. Specifically, suppose that the regression function $f^\star$ is L-Lipschitz.

(a) Suppose that $x_1, \ldots, x_n$ are evenly spaced on a d-dimensional unit ball. Prove that for any $i \in \mathcal{N}_k(x)$,

$$\|x_i - x\|_2 = O\left((k/n)^{1/d}\right) \quad \text{as } n \to \infty.$$

Hint: the volume of a d-dimensional ball with radius r is $\frac{\pi^{d/2} r^d}{\Gamma(d/2+1)}$.

(b) Bound the squared bias as $b(x)^2 = O\left(L^2(k/n)^{2/d}\right)$. Pause to think about the relationship between the bias and L, k, n, and d. Does this dependence qualitatively make sense?

(c) Finally, bound the point-wise excess risk:

$$\mathcal{R}(\widehat{f}, x) - \mathcal{R}(f^\star, x) = O\left(L^2(k/n)^{2/d} + \sigma^2/k\right).$$

1.4 Optimizing the bias-variance tradeoff

Now, let us optimize the bound on the excess risk over k. You may assume $L = \sigma = 1$ for this example.

(a) To get an idea of what the terms in the bound look like and what the best k might be, plot the individual terms in the bound for the excess risk and the bound itself for $n = 100$ and $d = 1$ as a function of k to visualize the tradeoff. Find the value of k that minimizes the excess risk.

(b) For general n and d, analytically optimize the bound over k and find the best choice of k as a function of n and d.

(c) Plugin this value of k to arrive at a bound on the excess squared risk of k-nearest-neighbor regressors in terms of n and d.

(d) To get a feel for the rate, plot the number of samples n required to achieve excess squared risk of 0.1 as a function of the dimension d. Do you find anything worrying about this plot?

2 k-nearest neighbor classification error rate

Similar to the previous question, we now derive bounds on the 0-1 risk of k-nearest-neighbor *classifier*, when the regression function is L-Lipschitz. To simplify calculations, you may use the bounds derived in the previous question where it was assumed $x_1, \ldots, x_n$ are evenly-spaced on a d-dimensional ball.

2.1 Deriving optimality of the Bayes classifier

To get started, we first establish optimality of the Bayes classifier. Consider the following setup.

- Data model: $(X, Y) \sim \mathbb{P}_{X,Y}$ is a distribution on $\mathbb{R}^d \times \{0, 1\}$.
- Quantity of interest: any classifier $c : \mathbb{R}^d \to \{0, 1\}$ versus the Bayes classifier $c^\star(x) = \mathbb{1}\{f^\star(x) \geq 1/2\}$ where $f^\star$ is the regression function.
- Metric: 0-1 risk defined as $r(c) := \mathbb{P}(Y \neq c(X))$.

(a) Show that the 0-1 risk of the Bayes classifier $c^\star$ is smaller than any other classifier c, meaning

$$r(c^\star) \leq r(c).$$

2.2 Bounding the excess risk of any plug-in classifier

Now, we bound the excess 0-1 risk of any plug-in classifier in terms the squared risk of the corresponding regressor. Consider the following setup.

- Data model: $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n) \overset{iid}{\sim} \mathbb{P}_{X,Y}$.
- Quantity of interest: any regressor $\widehat{f}$ and the corresponding plug-in classifier which is constructed such that $\widehat{c}(x) := \mathbb{1}\{\widehat{f}(x) \geq 1/2\}$.
- Metric: 0-1 risk of classifier $\widehat{c}$ defined as $r(\widehat{c}) = \mathbb{P}(Y \neq \widehat{c}(X))$ where $(X, Y) \sim \mathbb{P}_{X,Y}$ sampled independently from $(X_i, Y_i)_{i=1}^n$ and excess squared risk defined as $r(\widehat{c}) - r(c^\star)$.

(a) Show that

$$r(\widehat{c}) - r(c^\star) \leq 2\sqrt{\mathcal{R}(\widehat{f}) - \mathcal{R}(f^\star)}.$$

2.3 Bounding the excess risk of k-nearest-neighbor classifier

Finally, we bound below the excess 0-1 risk of k-nearest-neighbor classifier, when the regression function is Lipschitz. To simplify calculations, you may use the bounds derived in the first problem where it was assumed $x_1, \ldots, x_n$ are evenly-spaced on a d-dimensional ball.

(a) Write the k-nearest neighbor classifier as a function of the k-nearest neighbor regressor.

(b) Find an upper bound on the excess risk of the k-nearest neighbor classifier assuming $f^\star$ is L-Lipschitz.

3 k-nearest-neighbor regression and conformal prediction in action

In this question we will construct prediction intervals using split conformal inference.
Note: You do not need to implement k-nn yourself (some nice libraries for R and Python are `FNN` and `scikit-learn`, respectively). However, you will need to write your own implementation of split conformal inference.

To get started, we observe a bias-variance tradeoff in action and contrast empirical tuning of k with theoretical tuning as in Problem 1. Consider the following setup.

- Data model: $X \sim U[0, 1]$, and $(Y \mid X = x) \sim N(\sin(4x), x/3)$.
- Quantity of interest: k-nearest-neighbor regressor.
- Metric: mean squared error.

(a) Draw 400 points from $\mathbb{P}_{X,Y}$ and divide them into 4 sets of 100 points each. We refer to them as $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{valid}}$, $\mathcal{D}_{\text{calib}}$, and $\mathcal{D}_{\text{eval}}$ to be used in this and the following problems.

(b) Estimate k-nn regressors for $k \in \{1, \dots, 100\}$ on $\mathcal{D}_{\text{train}}$.

(c) Plot the empirical mean squared error of all 100 regressors on $\mathcal{D}_{\text{valid}}$ as a function of k. Compare it to the bias-variance tradeoff that you plotted above for $n = 100$ and $d = 1$ in Question 1. Is the regression function Lipschitz in this case?

(d) Pick the value $k^\star$ which minimizes the empirical mean squared error. Contrast it with the one computed from the calculation of best k in Question 1 for $n = 100$ and $d = 1$.

(e) Plot the original data, the regression function, and three k-nearest-neighbor regressors (one that shows undersmoothing, one that shows oversmoothing, and one that is corresponds to $k^\star$) evaluated on a fine enough grid of points in $[0, 1]$.

3.1 Conformal prediction in action

To get a sense of uncertainty of our regression estimates, we now construct prediction intervals using a split conformal method as follows. We fix the learnt regressor as the best k-nearest-neighbor regressor that we got above from empirical tuning.

(a) Use $\mathcal{D}_{\text{calib}}$ along with the learnt regressor to compute the residuals and compute appropriate quantiles for the purposes of 90% marginal coverage.

(b) For each x_i in $\mathcal{D}_{\text{eval}}$, form the corresponding conformal prediction interval and plot these 100 intervals along with the observed y_i, the learnt regressor, and the regression function.

(c) Repeat this procedure (including the previous part where you picked the best empirical regressor) 100 times and compute the average coverage over these repetitions. Is it close to 90%?

4 k-nearest-neighbor classification in action

In this question, we will play with the k-nearest-neighbor classifier and replicate two sets of results (one with synthetic data and one with real data) from Hastie et al. (2009).

4.1 k-nearest-neighbor classifier on easy versus hard classification problems

We mentioned in the class that the error (either in regression or classification) need not necessarily have a certain shape as a function of the tuning parameters. We observe this in action below for k-nearest-neighbor classification.

(a) Replicate the results in the left panel of Hastie et al. [2009, Figure 13.5]. The setup is described in Section 13.3.1 on page 468.

4.2 k-nearest-neighbor classifier on image scene classification

We now play with the k-nearest-neighbor classifier on a real dataset, where it performs better than most other classifiers.

(a) Replicate the test error reported in Hastie et al. [2009, Section 13.3.2] for the 5-nearest-neighbor classifier. The dataset is available at https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite). Use the same train and test split as provided in the dataset.

(b) Additionally, experiment with varying values of k and plot the test error as a function of k for some other choices. Does $k = 5$ provide the best test error?

References

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.