Week 5 Homework: Smoothness and Lipschitz Regression

QTM 490R: Machine Learning Theory

In this homework, we'll briefly review Bounded Variation Regression and then explore Lipschitz Regression, another form of smooth regression. We will focus on the one-dimensional case, although it extends very naturally to higher dimensions. Then we'll look into rates of convergence, comparing this new method to the stuff we've been using.

1 Review of Bounded Variation Regression

In class, we talked about using least squares regression to fit a function of *bounded total variation.* If we are fitting $\mu(x) = \mathrm{E}[Y_i \mid X_i = x]$ for covariates $X_i \in [0,1]$, this estimator is

$$\begin{aligned}
\hat{\mu} &= \underset{\substack{m \\ \rho_{TV}(m) \le B}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\{Y_i - m(X_i)\}^2 && \text{where} \\
\rho_{TV}(m) &= \int_0^1 |m'(x)|dx && \text{for differentiable } m \qquad (1)\\
&= \sup_{\substack{\text{increasing sequences} \\ 0=x_1<x_2<\ldots<x_k=1}} \sum_j |m(x_{j+1}) - m(x_j)| && \text{generally .}
\end{aligned}$$

The set of functions we're optimizing over, those with $\rho_{TV}(m) \le B$, is a set of functions that doesn't vary too much in total. It does, however, include both functions that vary slowly throughout the interval $[0,1]$ and those that vary quickly for a small part of it.

Exercise 1 *To get a sense of what the constraint $\rho_{TV}(m) \le B$ means, calculate $\rho_{TV}(m)$ for the following functions on $[0,1]$. These are repeats from class.*

1. $m(x) = x$
2. $m(x) = x^2$
3. $m(x) = e^x$

4. $m(x) = \sin(\pi x)$

5. $m(x) = \begin{cases} 0 & \textit{if } x < 1 \\ 1 & \textit{if } x = 1 \end{cases}$

6. $m(x) = \sin(1/x)$

Exercise 2 *For the following, which are a bit subtler, give an upper bound on $\rho_{TV}(m)$. If it's infinite, explain why.*

1. $m(x) = x \sin(1/x)$

2. $m(x) = x^2 \sin(1/x)$

3. $m(x) = x^{3/2} \sin(1/x)$

Hint: It might be hard to find the upper bound by looking at the graph of some of these functions. Instead, find the derivative and a corresponding upper bound for it, if possible. Your upper bound doesn't have to be tight. When you are bounding a sum, use the triangle inequality by adding the upper bound for each term.

2 Lipschitz Regression

In some cases, it may be implausible that $\mu(x)$ varies quickly anywhere. In that case, we may prefer to fit a *Lipschitz function*, for example by solving the following least squares problem.

$$\begin{aligned} \hat{\mu} &= \underset{\substack{m \\ \rho_{Lip}(m) \le B}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \{Y_i - m(X_i)\}^2 && \text{where} \\ \rho_{Lip}(m) &= \sup_{x \in [0,1]} |m'(x)| && \text{for differentiable } m \\ &= \sup_{\substack{x_1, x_2 \in [0,1] \\ x_1 \ne x_2}} \frac{|m(x_2) - m(x_1)|}{|x_2 - x_1|} && \text{generally .} \end{aligned} \tag{2}$$

We call $\rho_{Lip}(m)$ the *Lipschitz constant* of the function m. Let's interpret the general definition visually. It's a maximum of the absolute value of the slope of the functions's secants.

Equivalence. Our definition for differentiable functions is equivalent because (i) derivatives are included in the set of slopes we're maximizing over, as they are the slopes of tangents, which are just very short secants (ii) every slope in this set is equal to a derivative, as the mean value theorem tells us that the slope of the secant drawn from $x = a$ to $x = b$ is equal to the derivative of the function at some point between a and b.

2.1 Finding Lipschitz Constants

Exercise 3 *To get a sense of what this new type of constraint $\rho_{Lip}(m) \le B$ means, calculate $\rho_{Lip}(m)$ for the examples from Exercise 1. Bound it or explain why it's infinite for the examples from Exercise 2. Is $\rho_{TV}(m) \le \rho_{Lip}(m)$ for all of these examples? If so, either prove that it's true for all functions m on $[0,1]$ or find a counterexample.*

2.2 Fitting the Lipschitz Model

Now consider a least squares estimator based on a data-specific version of the Lipschitz constant.

$$\begin{aligned} \hat{\mu} &= \operatorname*{argmin}_{\substack{m \\ \hat{\rho}_{Lip}(m) \le B}} \frac{1}{n}\sum_{i=1}^{n}\{Y_i - m(X_i)\}^2 \quad \text{where} \\ \hat{\rho}_{Lip}(m) &= \sup_{\substack{i,j \in 1 \ldots n \\ X_i \ne X_j}} \frac{|m(X_i) - m(X_j)|}{|X_i - X_j|}. \end{aligned} \tag{3}$$

This is something we can handle. This depends on the values of $m(x)$ only at the observed data points $x \in \{X_1 \ldots X_n\}$, so we can implement it as an optimization over a vector $\vec{m} \in \mathbb{R}^n$ with the interpretation that $\vec{m}_i = m(X_i)$. The constraint $\hat{\rho}_{Lip}(m) \le B$ can be expressed as a set of constraints on $\vec{m}_i - \vec{m}_j$ for pairs i, j.

Exercise 4 *Rewrite this problem as a constrained optimization over the vector $\vec{m}$. Try to do it so what you've written translates straightforwardly into CVXR code.*

Tip. CVXR seems to be having some trouble with this one if we use division in our constraint, so don't. To write your constraint without division, observe that the following set of constraints are equivalent: (i) $\max_{i \le n} u_i/v_i \le B$, (ii) $u_i/v_i \le B$ for all $i \in 1 \ldots n$, and (iii) $u_i \le Bv_i$ for all $i \in 1 \ldots n$.

Exercise 5 *Implement that optimization in* **R**. *That is, write an* **R** *function* **lipreg** *analogous to* **tvreg** *from the bounded variation lab that solves* (3). *Then, from the six distributions described below, sample $n = 100$ observations $(X_1, Y_1) \ldots (X_n, Y_n)$ and use your code to calculate predictions $\hat{\mu}(X_1) \ldots \hat{\mu}(X_1)$ based on the solution to* (3) *with variation bound $B = 1$. Each time, plot your predictions on top of the data, i.e., make a single scatter plot showing both your predictions $(X_i, \hat{\mu}(X_i))$ and your observations (X_i, Y_i). Turn in those six plots as your solution to this exercise.*

We'll sample observations around six curves.

1. A step, $\mu(x) = 1(x >= .5)$.
2. A line, $\mu(x) = x$.

3. A vee, $\mu(x) = (x - .5)1(x \ge .5)$.
4. A sine, $\mu(x) = \sin(\pi x)$.
5. A damped rapidly oscillating curve, $\mu(x) = x \sin(1/x)$.
6. A more-damped rapidly-oscillating curve, $\mu(x) = x^{3/2} \sin(1/x)$.

For each, we'll work with independent and identically distributed observations $(X_1, Y_1) \ldots (X_n, Y_n)$ where X_i is drawn from the uniform distribution on $[0, 1]$ and $Y_i = \mu(X_i) + \varepsilon_i$ for ε_i drawn from the normal distribution with mean zero and standard deviation $\sigma = 1/10$.

Tip. The `R` built-in function `expand.grid` may be useful.

Tip. The **tvreg** code from the bounded variation lab is the solution to **Exercise 3.2** there.

Exercise 6 *Revisit the curves $\hat{\mu}$ you fit in the last exercise. For each, answer these questions.*

1. *Does it fit the data?*
2. *If not, what — if anything — could we do to fit the data better?*

Then, if there is something you can do, do it and include the resulting plot.

2.3 Filling in the gaps

At this point, you have an estimator $\hat{\mu}$ that minimizes squared error among the functions m satisfying $\hat{\rho}_{Lip}(m) \le B$, at least in the sense that you can evaluate $\hat{\mu}(X_i)$ for $i \in 1 \ldots n$. This lets us plot some isolated points. But we want a complete curve $\hat{\mu}(x)$ for $x \in [0, 1]$ that satisfies $\rho_{Lip}(\hat{\mu}) \le B$, and we want it to be the best-fitting such curve, i.e., we want the solution to (2).

To do this, we'll fill in the gaps linearly. That is, having sorted X_i into increasing order, we will define $\hat{\mu}(x)$ everywhere on $[X_1, X_n]$ by drawing line segments between successive points $\{X_i, \hat{\mu}(X_i)\}$ and $\{X_{i+1}, \hat{\mu}(X_{i+1})\}$, and extend the leftmost and rightmost segment to fill the intervals $[0, X_1]$ and $[X_n, 1]$. This gives us a piecewise-linear solution to (3). First, we'll implement it. Or borrow an implementation from somewhere else. Then we'll verify that it is, in fact, a solution to (2).

2.3.1 Implementation

Exercise 7 *Write out a formula for the piecewise-linear curve $\hat{\mu}(x)$ in terms of $\hat{\mu}(X_1) \ldots \hat{\mu}(X_n)$. Then implement it and add the curve $\hat{\mu}(x)$ for $x \in [0, 1]$ to your plots from the last exercise.*

Tip. For coding a piecewise linear function, try to modify the predict.piecewise.constant function from bounded variation lab **Section 2 Code**.

2.3.2 Verification

Exercise 8 *Consider any pair $x < x'$. Show that for any piecewise-linear function m with breaks at $X_1 \ldots X_n$, the secant slope $\{m(x')-m(x)\}/(x'-x)$ between these points is a* weighted average *of the slopes $\{m(X_{j+1})-m(X_j)\}/(X_{j+1}-X_j)$ of the segments that lie between them. Explain why this implies that our piecewise-linear solution $\hat{\mu}$ satisfies $\rho_{Lip}(\hat{\mu}) = \hat{\rho}_{Lip}(\hat{\mu})$ and why this implies that $\hat{\mu}$ solves* (2)*.*

Tip. See the beginning of the Bounded Variation Lab's *Implementation* section for a broadly similar argument.

2.4 Optimized Fitting

We can speed up our fitting code by simplifying our set of constraints by hand. In particular, I claim that you get the same solution if you impose the constraint $|m(X_i) - m(X_j)|/|X_i - X_j| \leq B$ for adjacent points. That is, if the points $X_1 \ldots X_n$ are sorted in increasing order, it's equivalent to impose the constraint for the pairs $(i, j = i + 1)$.

Exercise 9 *Prove it! Then implement it and check that your solution agrees with the one you got before using the all-pairs constraint. Include the proof as your solution. No need to turn in code, but you'll want this faster implementation later.*

Tip. The proof should be easy. Use ideas from the last exercise.

3 Rates of Convergence

Now we've got three nonparametric regression models: monotone curves, bounded variation curves, and lipschitz curves. To keep things simple, we'll be working with data sampled around one curve: $\mu(x) = 0$. That is, we'll work with independent and identically distributed observations $(X_1, Y_1) \ldots (X_n, Y_n)$ where X_i is drawn from uniform distribution on $[0, 1]$ and Y_i drawn independently from the normal distribution with mean zero and standard deviation $\sigma = .5$.

Tip. We looked at rates of convergence for the R Learner at the end of the R Learner Lab, so you may want to base your code on what's there. See the section *Comparing to the Oracle*. But we didn't do any replication and averaging over samples there, so you'll have to add that here.

Exercise 10 *Draw a sample of size* $N = 400$ *from this distribution. To get samples of sizes* $n = \{25, 50, 100, 200, 400\}$, *use the first* 25, 50, *etc. observations.*

At all of these sample sizes, fit an increasing curve and a bounded variation and lipschitz curves with budgets $B = 1$. *Calculate sample MSE* $\|\hat{\mu} - \mu\|^2_{L_2(Pn)}$ *and population MSE* $\|\hat{\mu} - \mu\|^2_{L_2(P)}$ *for each. Repeat this ten times and average the results to get estimates of expected sample MSE and expected population MSE at each sample size* n. *Include plots of these as a function of* n *as your solution.*

Let's try to summarize these plots by rates of convergence.

Exercise 11 *For each of your four regression models, use log-log regression to fit a curve of the form* αn^{-r} *to your estimates of sample and population MSE. Plot the resulting predictions of MSE,* $\hat{\alpha} n^{-\hat{r}}$, *on top of your actual MSE curves from the the previous exercise to check their accuracy. Include these plots and report these rates of convergence* r *as your solution. Briefly comment on what you see, too.*

Tip. We often talk about rates of convergence in terms of $RMSE$ ($\sqrt{MSE}$) instead of MSE. If $MSE \approx \alpha n^{-r}$, then $RMSE \approx \sqrt{\alpha} n^{-r/2}$, i.e., we should halve the exponent we get in this regression if, as usual, we want to talk about the typical size of $\|\hat{\mu} - \mu\|_{L_2(Pn)}$ and $\|\hat{\mu} - \mu\|_{L_2(P)}$ instead of their squares.