Function estimation on high dimensions

김지수 (Jisu KIM)

통계적 기계학습(Statistical Machine Learning), 2024 1st semester

The lecture note is a minor modification of the lecture notes from Prof. Yongdai Kim's "Statistical Machine Learning", and Prof Larry Wasserman and Ryan Tibshirani's "Statistical Machine Learning". Also, see Section 9.1, 9.2, and 11.2 from [3] and Section 7.7 from [4].

1 Review

1.1 Basic Model for Supervised Learning

- Input(입력) / Covariate(설명 변수) : $x \in \mathbb{R}^d$, so $x = (x_1, \dots, x_d)$.
- Output(출력) / Response(반응 변수): $y \in \mathcal{Y}$. If y is categorical, then supervised learning is "classification", and if y is continuous, then supervised learning is "regression".
- Model(모형) :
$$y \approx f(x).$$
If we include the error ϵ to the model, then it can be also written as
$$y = \phi(f(x), \epsilon).$$
For many cases, we assume additive noise, so
$$y = f(x) + \epsilon.$$
- Assumption(가정): f belongs to a family of functions $\mathcal{M}$. This is the assumption of a model: a model can be still used when the corresponding assumption is not satisfied in your data.
- Loss function(손실 함수): $\ell(y, a)$. A loss function measures the difference between estimated and true values for an instance of data.
- Training data(학습 자료): $\mathcal{T} = \{(y_i, x_i), i = 1, \dots, n\}$, where (y_i, x_i) is a sample from a probability distribution P_i. For many cases we assume i.i.d., or x_i's are fixed and y_i's are i.i.d..
- Goal(목적): we want to find f that minimizes the expected prediction error,
$$f^0 = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(Y,X)\sim P} \left[\ell(Y, f(X))\right].$$
Here, $\mathcal{F}$ can be different from $\mathcal{M}$; $\mathcal{F}$ can be smaller then $\mathcal{M}$.
- Prediction model(예측 모형): f^0 is unknown, so we estimate f^0 by $\hat{f}$ using data. For many cases we minimizes on the empirical prediction error, that is taking the expectation on the empirical distribution $P_n = \frac{1}{n}\sum_{i=1}^n \delta_{(Y_i, X_i)}$.
$$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{P_n} \left[\ell(Y, f(X))\right] = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \ell(Y_i, f(X_i)).$$
- Prediction(예측): if $\hat{f}$ is a predicted function, and x is a new input, then we predict unknown y by $\hat{f}(x)$.

1.2 Linear Regression

From the additive noise model

$$y = f(x) + \epsilon,\ f \in \mathcal{M},$$

Linear Regression Model (선형회귀모형) is that

$$\mathcal{M} = \mathcal{F} = \left\{ \beta_0 + \sum_{j=1}^{d} \beta_j x_j : \beta_j \in \mathbb{R} \right\}.$$

For estimating β, we use least squares: suppose the training data is $\{(y_i, x_{ij}) : 1 \leq i \leq n, 1 \leq j \leq p\}$. We use square loss

$$\ell(y, a) = (y - a)^2,$$

then the eimpirical loss becomes the residual sum of square (RSS) as

$$\begin{aligned} RSS(\beta) &= \sum_{i=1}^{n} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j \right)^2 . \end{aligned}$$

Let $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_d)$ be the nimimizor of RSS, then the predicted function is

$$\hat{f}(x) = \hat{\beta}_0 + \sum_{j=1}^{d} \hat{\beta}_j x_j.$$

1.3 Notation

- We will define an empirical norm $\| \cdot \|_n$ in terms of the training points x_i, $i = 1, \ldots, n$, acting on functions $f : \mathbb{R}^d \to \mathbb{R}$, by

  $$\|f\|_n^2 = \frac{1}{n} \sum_{i=1}^{n} f^2(x_i).$$

  This makes sense no matter if the inputs are fixed or random (but in the latter case, it is a random norm)

- When the inputs are considered random, we will write P_X for the distribution of X, and we will define the L_2 norm $\| \cdot \|_2$ in terms of P_X, acting on functions $f : \mathbb{R}^d \to \mathbb{R}$, by

  $$\|f\|_2^2 = \mathbb{E}[f^2(X)] = \int f^2(x)\, dP_X(x).$$

  So when you see $\| \cdot \|_2$ in use, it is a hint that the inputs are being treated as random

- A quantity of interest will be the (squared) error associated with an estimator $\hat{f}$ of f_0, which can be measured in either norm:

  $$\|\hat{f} - f_0\|_n^2 \quad \text{or} \quad \|\hat{f} - f_0\|_2^2.$$

  In either case, this is a random quantity (since $\hat{f}$ is itself random). We will study bounds in probability or in expectation. The expectation of the errors defined above, in terms of either norm (but more typically the L_2 norm) is most properly called the risk; but we will often be a bit loose in terms of our terminology and just call this the error

1.4 Hölder Spaces and Sobolev Spaces

The class of Lipschitz functions $H(1, L)$ on $T \subset \mathbb{R}$ is the set of functions $g : T \to \mathbb{R}$ such that

$$|g(y) - g(x)| \leq L|x - y| \text{ for all } x, y \in T.$$

A differentiable function is Lipschitz if and only if it has bounded derivative. Conversely a Lipschitz function is differentiable almost everywhere.

Let $T \subset \mathbb{R}$ and let β be an integer. The Hölder space $H(\beta, L)$ is the set of functions $g : T \to \mathbb{R}$ such that g is $\ell = \beta - 1$ times differentiable and satisfies

$$|g^{(\ell)}(y) - g^{(\ell)}(x)| \leq L|x - y| \text{ for all } x, y \in T.$$

(There is an extension to real valued β but we will not need that.) If $g \in H(\beta, L)$ and $\ell = \beta - 1$, then we can define the Taylor approximation of g at x by

$$\tilde{g}(y) = g(y) + (y - x)g'(x) + \cdots + \frac{(y - x)^\ell}{\ell!} g^{(\ell)}(x)$$

and then $|g(y) - \tilde{g}(y)| \leq |y - x|^\beta$.

The definition for higher dimensions is similar. Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$. Let β and L be positive numbers. Given a vector $s = (s_1, \ldots, s_d)$, define $|s| = s_1 + \cdots + s_d$, $s! = s_1! \cdots s_d!$, $x^s = x_1^{s_1} \cdots x_d^{s_d}$ and

$$D^s = \frac{\partial^{s_1 + \cdots + s_d}}{\partial x_1^{s_1} \cdots \partial x_d^{s_d}}.$$

Let β be a positive integer. Define the *Hölder class*

$$H_d(\beta, L) = \left\{ g : \ |D^s g(x) - D^s g(y)| \leq L\|x - y\|, \quad \text{for all } s \text{ such that } |s| = \beta - 1, \text{ and all } x, y \right\}. \tag{1}$$

For example, if $d = 1$ and $\beta = 2$ this means that

$$|g'(x) - g'(y)| \leq L\,|x - y|, \quad \text{for all } x, y.$$

The most common case is $\beta = 2$; roughly speaking, this means that the functions have bounded second derivatives.

Again, if $g \in H_d(\beta, L)$ then $g(x)$ is close to its Taylor series approximation:

$$|g(u) - g_{x,\beta}(u)| \leq L\|u - x\|^\beta \tag{2}$$

where

$$g_{x,\beta}(u) = \sum_{|s| \leq \beta} \frac{(u - x)^s}{s!} D^s g(x). \tag{3}$$

In the common case of $\beta = 2$, this means that

$$\left| p(u) - [p(x) + (x - u)^T \nabla p(x)] \right| \leq L||x - u||^2.$$

The Sobolev class $S_1(\beta, L)$ on $T \subset \mathbb{R}$ is the set of β times differentiable functions (technically, it only requires weak derivatives) $g : T \to \mathbb{R}$ such that

$$\int_T (g^{(\beta)}(x))^2 dx \leq L^2.$$

Again this extends naturally to $\mathbb{R}^d$. Also, there is an extension to non-integer β. It can be shown that if T is bounded, then $H_d(\beta, L) \subset S_d(\beta, L')$ for appropriate L and L'.

2 Introduction

For low dimensions, there are various methods of estimating functions. Examples are Spline and kernel methods.

These methods do not work well for high dimensions due to the curse of dimensionality. We will see how researchers have been trying to avoid the curse of dimensionality, by following examples:

- Generalized Additive Models(일반화가법모형)
- Projection Pursuit Regression(사영추적회귀)
- Multivariate Adaptive Regression Spline(MARS, 다변량 적응 회귀스플라인): not covered in this lecture. See Section 9.4 from [3].

3 Generalized Additive Models(일반화가법모형)

3.1 Motivation and definition

Computational efficiency and statistical efficiency are both very real concerns as the dimension d grows large, in nonparametric regression. If you're trying to fit a kernel, thin-plate spline, or RKHS estimate in > 20 dimensions, without any other kind of structural constraints, then you'll probably be in trouble (unless you have a very fast computer and tons of data).

Recall that the minimax rate over the Hölder class $H_d(\alpha, L)$ is

$$\inf_{\hat{f}} \sup_{f_0 \in H_d(\alpha,L)} \mathbb{E}\|\hat{f} - f_0\|_2^2 \gtrsim n^{-2\alpha/(2\alpha+d)}, \tag{4}$$

which has an exponentially bad dependence on the dimension d. This is usually called the curse of dimensionality (though the term apparently originated with [1], who encountered an analogous issue but in a separate context—dynamic programming).

What can we do? One answer is to change what we're looking for, and fit estimates with less flexibility in high dimensions. Think of a linear model in d variables: there is a big difference between this and a fully nonparametric model in d variables. Is there some middle man that we can consider, that would make sense?

Additive models play the role of this middle man. Instead of considering a full d-dimensional function of the form

$$f(x) = f(x(1), \dots, x(d)) \tag{5}$$

we restrict our attention to functions of the form

$$f(x) = f_1(x(1)) + \dots + f_d(x(d)). \tag{6}$$

As each function f_j, $j = 1, \dots, d$ is univariate, fitting an estimate of the form (6) is certainly less ambitious than fitting one of the form (5). On the other hand, the scope of (6) is still big enough that we can capture interesting (marginal) behavior in high dimensions.

There is a theoretical justification for doing this. It had been known since the 1950s (via the Kolmogorov–Arnold representation theorem) that any multivariate continuous function $f : [0,1]^d \to \mathbb{R}$ could be represented as sums and compositions of univariate functions,

$$f(x) = \sum_{i=1}^{2d} \Phi_i \left(\sum_{j=1}^{d} \phi_{i,j}(x(j)) \right).$$

Unfortunately, though the Kolmogorov–Arnold representation theorem asserts the existence of a function of this form, it gives no mechanism whereby one could be constructed. Certain constructive proofs exist, but they tend to require highly complicated (i.e. fractal) functions, and thus are not suitable for modeling approaches. Therefore, the *generalized additive model* drops the outer sum, and demands instead that the function belong to a simpler class,

$$f(x) = \Phi \left(\sum_{j=1}^{d} \phi_{i,j}(x(j)) \right),$$

where Φ is a smooth monotonic function. Writing g for the inverse of Φ, this is traditionally written as

$$g(f(x)) = \sum_{j=1}^{d} f_j(x(j)).$$

When this function is approximating the expectation of some observed quantity, it could be written as

$$g(\mathbb{E}[Y]) = \beta_0 + f_1(x(1)) + \cdots + f_d(x(d)),$$

which is the standard formulation of a generalized additive model. We consider the case $\beta_0 = 0$ and $g(y) = y$ for convenience.

There is also a link to naive-Bayes classification (not covered in this class).

The choice of estimator of the form (6) need not be regarded as an assumption we make about the true function f^0, just like we don't always assume that the true model is linear when using linear regression. In many cases, we fit an additive model because we think it may provide a useful approximation to the truth, and is able to scale well with the number of dimensions d.

A classic result by [8] encapsulates this idea precisely. He shows that, while it may be difficult to estimate an arbitrary regression function f^0 in multiple dimensions, we can still estimate its *best additive approximation* $\bar{f}^{\text{add}}$ well. Assuming each component function $\bar{f}_j^{\text{add}}$, $j = 1, \ldots, d$ lies in the Hölder class $H_1(\alpha, L)$, for constant $L > 0$, and we can use an additive model, with each component $\hat{f}_j$, $j = 1, \ldots, d$ estimated using an appropriate kth degree spline, to give

$$\mathbb{E}\|\hat{f}_j - \bar{f}_j^{\text{add}}\|_2^2 \lesssim n^{-2\alpha/(2\alpha+1)}, \quad j = 1, \ldots, d.$$

Hence each component of the best additive approximation $\bar{f}^{\text{add}}$ to f^0 can be estimated at the optimal univariate rate. Loosely speaking, though we cannot hope to recover f^0 arbitrarily, we can recover its major structure along the coordinate axes.

3.2 Backfitting

Estimation with additive models is actually very simple; we can just choose our favorite univariate smoother (i.e., nonparametric estimator), and cycle through estimating each function f_j, $j = 1, \ldots, d$ individually (like a block coordinate descent algorithm). Denote the result of running our chosen univariate smoother to regress $Y = (Y_1, \ldots, Y_n) \in \mathbb{R}^n$ over the input points $Z = (Z_1, \ldots, Z_n) \in \mathbb{R}^n$ as

$$\hat{f} = \text{Smooth}(Z, Y).$$

E.g., we might choose $\text{Smooth}(\cdot, \cdot)$ to be a cubic smoothing spline with some fixed value of the tuning parameter λ, or even with the tuning parameter selected by generalized cross-validation

Once our univariate smoother has been chosen, we initialize $\hat{f}_1, \ldots, \hat{f}_d$ (say, to all to zero) and cycle over the following steps for $j = 1, \ldots, d, 1, \ldots, d, \ldots$:

1. define $r_i = Y_i - \sum_{\ell \neq j} \hat{f}_l(X_i(l))$, $i = 1, \ldots, n$;
2. smooth $\hat{f}_j = \text{Smooth}(x(j), r)$;
3. center $\hat{f}_j = \hat{f}_j - \frac{1}{n}\sum_{i=1}^{n} \hat{f}_j(X_i(j))$.

This algorithm is known as *backfitting*. In last step above, we are removing the mean from each fitted function $\hat{f}_j$, $j = 1, \ldots, d$, otherwise the model would not be identifiable. Our final estimate therefore takes the form

$$\hat{f}(x) = \bar{Y} + \hat{f}_1(x(1)) + \cdots + \hat{f}(x(d))$$

where $\bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$. [2] provide a very nice exposition on the some of the more practical aspects of backfitting and additive models.

In many cases, backfitting is equivalent to blockwise coordinate descent performed on a joint optimization criterion that determines the total additive estimate. E.g., for the additive cubic smoothing spline optimization problem,

$$\hat{f}_1, \ldots, \hat{f}_d = \text{argmin}_{f_1, \ldots, f_d} \sum_{i=1}^{n} \Big(Y_i - \sum_{j=1}^{d} f_j(x_{ij}) \Big)^2 + \sum_{j=1}^{d} \lambda_j \int_0^1 f_j''(t)^2 \, dt,$$

backfitting is exactly blockwise coordinate descent (after we reparametrize the above to be in finite-dimensional form, using a natural cubic spline basis).

The beauty of backfitting is that it allows us to think *algorithmically*, and plug in whatever we want for the univariate smoothers. This allows for several extensions. One extension: we don't need to use the same univariate smoother for each dimension, rather, we could mix and match, choosing $\mathrm{Smooth}_j(\cdot,\cdot)$, $j = 1,\dots,d$ to come from entirely different methods or giving estimates with entirely different structures.

Another extension: to capture interactions, we can perform smoothing over (small) groups of variables instead of individual variables. For example we could fit a model of the form

$$f(x) = \sum_j f_j(x(j)) + \sum_{j<k} f_{jk}(x(j), x(k)).$$

3.3 Error rates

Error rates for additive models are both kind of what you'd expect and surprising.

What you'd expect: if the underlying function f^0 is additive, and we place standard assumptions on its component functions, such as $f_{0,j} \in S_1(m, C)$, $j = 1,\dots,d$, for a constant $C > 0$, a somewhat straightforward argument building on univariate minimax theory gives us the lower bound

$$\inf_{\hat{f}} \sup_{f^0 \in \oplus_{j=1}^d S_1(m,C)} \mathbb{E}\|\hat{f} - f^0\|_2^2 \gtrsim dn^{-2m/(2m+1)}.$$

This is simply d times the univariate minimax rate. (Note that we have been careful to track the role of d here, i.e., it is not being treated like a constant.) Also, standard methods like backfitting with univariate smoothing splines of polynomial order $k = 2m - 1$, will also match this upper bound in error rate (though the proof to get the sharp linear dependence on d is a bit trickier).

Surprising: an additive model with different levels of smoothness among its component functions behaves in an interesting manner. Just in $d = 2$ dimensions, let us consider $f_0(x) = f_{0,1}(x_{\cdot 1}) + f_{0,2}(x_{\cdot 2})$, where $f_{0,1}$ is a lot smoother than $f_{0,2}$, e.g., $f_{0,1} \in W_1(2, C_1)$ and $f_{0,2} \in F(0, C_2)$, so

$$\int_0^1 f_{0,1}''(t)^2 \, dt \le C_1 \quad \text{and} \quad TV(f_{0,2}) \le C_2,$$

for constants $C_1, C_2 > 0$. Suppose also that we used an additive model to estimate f_0, with (say) a 3rd-order smoothing spline for the first component smoother, and a 0th-order locally adaptive regression spline for the second component smoother. Now, assuming each smoother was appropriately tuned, should we expect that

$$\|\hat{f}_1 - f_{0,1}\|_n^2 \lesssim n^{-4/5} \quad \text{and} \quad \|\hat{f}_2 - f_{0,2}\|_n^2 \lesssim n^{-2/3}, \tag{7}$$

each having the error rate associated with their corresponding univariate problem, or

$$\|\hat{f}_1 - f_{0,1}\|_n^2 \lesssim n^{-2/3} \quad \text{and} \quad \|\hat{f}_2 - f_{0,2}\|_n^2 \lesssim n^{-2/3}, \tag{8}$$

where the rougher of the two components dictates both rates? Recent work by [?] shows that (provided $x_{\cdot 1}, x_{\cdot 2}$ are not too correlated) it is the first case (7) that occurs

This is somewhat surprising, because if you think about it from the perspective of backfitting, at convergence, we have

$$\hat{f}_1 = \mathrm{Smooth}_1\Big(x_{\cdot 2}, \big(y_i - \hat{f}_2(x_{i2})\big)_{i=1}^n\Big),$$

a cubic smoothing spline fit to the effective responses $r_i = y_i - \hat{f}_2(x_{i2})$, $i = 1,\dots,n$. If we were actually fitting a smoothing spline to $f_{0,1}(x_{i1}) + \epsilon_i$, $i = 1,\dots,n$, then we'd see a $n^{-4/5}$ error rate. But we're not; instead we're fitting a cubic smoothing spline to

$$y_i - \hat{f}_2(x_{i2}) = f_{0,1}(x_{i1}) + \epsilon_i + \underbrace{\big(f_{0,2}(x_{i2}) - \hat{f}_2(x_{i2})\big)}_{e_{2i}}, \quad i = 1,\dots,n.$$

The terms denoted e_{2i}, $i = 1,\dots,n$ above are the errors in estimating the second component function at the input points. In the best case, we should hope for $\|e_2\|_2^2/n \asymp n^{-2/3}$. Doesn't this $n^{-2/3}$ pertubation mess up our estimation of $f_{0,1}$? Surprisingly, it does not.

It is worth noting that the proof given by [?] is very intricate, and does not obviously extend beyond $d = 2$ components. (Also, it is worth noting that this result relates to older, classic results from semiparametric estimation.)

3.4 Sparse additive models

Recently, *sparse additive models* have received a good deal of attention. In truly high dimensions, we might believe that only a small subset of the variables play a useful role in modeling the regression function, so might posit a modification of (6) of the form

$$f(x) = \sum_{j \in S} f_j(x(j))$$

where $S \subseteq \{1, \dots, d\}$ is an unknown subset of the full set of dimensions.

This is a natural idea, and to estimate a sparse additive model, we can use methods that are like nonparametric analogies of the lasso (more accurately, the group lasso). This is a research topic still very much in development; some recent works are [5, 7, 6].

3.5 Pros and Cons of Generalized Additive Models

- GAMs allow us to fit a non-linear f_j to each X_j, so that we can automatically model non-linear relationships that standard linear regression will miss. This means that we do not need to manually try out many different transformations on each variable individually.
- The non-linear fits can potentially make more accurate predictions for the response Y.
- Because the model is additive, we can examine the effect of each X_j on Y individually while holding all of the other variables fixed.
- The smoothness of the function f_j for the variable X_j can be summarized via degrees of freedom.
- The main limitation of GAMs is that the model is restricted to be additive. With many variables, important interactions can be missed. However, as with linear regression, we can manually add interaction terms to the GAM model by including additional predictors of the form $X_j \times X_k$. In addition we can add low-dimensional interaction functions of the form $f_{jk}(X_j, X_k)$ into the model; such terms can be fit using two-dimensional smoothers such as local regression, or two-dimensional splines.

4 Projection Pursuit Regression(PPR, 사영추적회귀)

Projection Pursuit Regression(PPR, 사영추적회귀) is introduced by Friedman and Stuetzle (1981). The model is:

$$Y = \beta_0 + \sum_{m=1}^{M} \beta_m \phi_m(a_m^\top x) + \epsilon.$$

Sometimes we assume $\mathbb{E}(\phi_m(a_m^\top x)) = 0, \mathbb{E}(\phi_m^2(a_m^\top x)) = 1$ for $m = 1, \dots, M$. Parameters are: $(\beta_0, \dots, \beta_M; \phi_1, \dots, \phi_M; a_1, \dots, a_M)$. This is an additive model, but in the derived features $a_m^\top x$ rather than the inputs themselves. For the name "Projection Pursuit", "Projection" indicates that x is projected onto the direction vectors $a_1, \dots, a_M$ to get the lengths $a_m^\top x$ of the projection. "Pursuit" indicates that an optimization technique is used to find "good" direction vectors $a_1, \dots, a_M$.

PPR can include interaction terms: Suppose $\mathbb{E}(Y|x) = x_1 x_2$. Then we can represent the regression function by $\beta_1\phi_1(a_1^\top x) + \beta_2\phi_2(a_2^\top x)$ where

- $\beta_1 = \beta_2 = 1/4$
- $a_1 = (1,1)^\top, a_2 = (1,-1)^\top$
- $\phi_1(t) = t^2, \phi_2(t) = -t^2$.

The estimation is done by Foward-Backward method.

1. Choose $M_{\min} > 0$ and choose $M > M_{\min}$.
2. Forward stepwise procedure
   (a) For $k = 1, M$
      i. Choose a_k arbitrary.

ii. Let $r_i = y_i - \sum_{l=1}^{k-1} \hat{\beta}_l \hat{\phi}_l(a_l^t x_i)$.

iii. Construct $\hat{\phi}_k$ with r_i as response and $a_k^\top x$ as input.

iv. Let $\hat{\phi}_k(t) = \hat{\phi}_k(t)/\|\hat{\phi}_k(t)\|, \hat{\beta}_k = \|\hat{\phi}_k(t)\|$ where $\|\hat{\phi}_k(t)\|^2 = \sum_{i=1}^n \hat{\phi}_k^2(a_k^t x_i)/n$.

v. Adjust a_k by

$$\hat{a} = \text{argmin} \sum_{i=1}^{n} (r_i - \hat{\beta}_k \hat{\phi}_k(a^T x_i))^2.$$

vi. Repeat $(c) - (e)$ until converge.

(b) End For

3. Backward stepwise procedure

(a) Among M terms $\hat{\phi}_1, \ldots, \hat{\phi}_M$, delete least significant terms (based on $|\hat{\beta}_k|$) one by one until $M_{\min}$ many terms remain.

4. Model Selection

(a) After Forward stepwise procedure and Backward stepwise procedure, we have $M - M_{\min} + 1$ many models. We choose one optimal model by use of various model selection techniques such as AIC, BIC etc.

References

[1] Richard Bellman. *Adaptive Control Processes.* Princeton University Press, 1962.

[2] Trevor Hastie and Robert Tibshirani. *Generalized Additive Models.* Chapman and Hall, 1990.

[3] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning.* Springer Series in Statistics. Springer, New York, second edition, 2009. Data mining, inference, and prediction.

[4] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning—with applications in R.* Springer Texts in Statistics. Springer, New York, [2021] ©2021. Second edition [of 3100153].

[5] Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *Annals of Statistics*, 34(5):2272—2297, 2006.

[6] Garvesh Raskutti, Martin Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *Journal of Machine Learning Research*, 13:389–427, 2012.

[7] Pradeep Ravikumar, Han Liu, John Lafferty, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society: Series B*, 75(1):1009–1030, 2009.

[8] Charles Stone. Additive regression models and other nonparametric models. *Annals of Statistics*, 13(2):689–705, 1985.