Lec 20: Additive models

Weiping Zhang

2018.12.12

Motivation and definition

Backfitting

Additive Partially Linear Models

Case study

Introduction

- Additive models are very useful for approximating the high-dimensional regression mean functions. They and their extensions have become one of the most widely used nonparametric techniques since the excellent monograph by Hastie and Tibshirani (1990) and the companion software as described in Chambers and Hastie (1991). For a recent survey on additive models, see Horowitz (2014).
- In the regression framework, a simple additive model is defined by

  $$Y = \beta_0 + f_1(x_1) + \cdots + f_p(x_p) + \epsilon \tag{1}$$

  where
  $E(\epsilon|X_1, \ldots, X_p) = 0, E(\epsilon^2|X_1, \ldots, X_p) = \sigma^2(X_1, \ldots, X_p)$.

Identifiability

- Note that the $\beta_0, f_1, \ldots, f_p$ are not identified without further restrictions. To prevent ambiguity, various identification conditions can be assumed. For example, one can assume that either

  $$Ef_j(X_j) = 0, j = 1, \ldots, p$$

  or

  $$Ef_j(0) = 0, j = 1, \ldots, p$$

  or

  $$\int f_j(v)dv = 0, j = 1, \ldots, p$$

  whichever is convenient for the estimation method on hand.

- We also assume that the f_j's are smooth functions so that they can be estimated as well as the one-dimensional nonparametric regression problem (Stone, 1985, 1986). Hence, the curse of dimensionality is avoided.

- Under the identification conditions that $Ef_j(X_j) = 0, j = 1, \ldots, p$ and $E(\epsilon | X_1, \ldots, X_p) = 0$, we have $EY = \beta_0$. So that the intercept β_0 can be estimated by the sample mean $\bar{Y} = \frac{1}{n} \sum_{i=1}^{n} Y_i$.
- Since $\bar{Y}$ converges to β_0 at the parametric $\sqrt{n}$-rate, which is faster than any nonparametric convergence rate, here we will simply work on the model without β_0 in (1) by assuming $EY = 0$.
- Additive models of the form (1) have been shown to be useful in practice. They naturally generalize the linear regression models and allow interpretation of marginal changes, i.e., the effect of one variable, say X_j on the conditional mean function $m(x) = E(Y | X_1, \ldots, X_p)$ holding everything else constant. They are also interesting from a theoretical perspective since they combine flexible nonparametric modeling of many variables with statistical precision that is typical for just one explanatory variable.

Notations

- Now let us define the **Holder class** of functions $H_d(k+\gamma, L)$, for an integer $k \geq 0, 0 < \gamma \leq 1$ and $L > 0$, to contain all k times differentiable functions $f : \mathbb{R}^d \mapsto \mathbb{R}$ such that

$$\left| \frac{\partial^k f(x)}{\partial x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}} - \frac{\partial^k f(z)}{\partial x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_d^{\alpha_d}} \right| \leq L \|x - z\|_2^\gamma,$$

  for all x, z, and $\alpha_1 + \cdots + \alpha_d = k$. Note that $H_d(1, L)$ is the space of all $L-$Lipschitz functions, and $H_d(k+1, L)$ is the space of all functions whose kth-order partial derivatives are $L-$Lipschitz

- As an aside, why did we study the Holder class $H_d(k+\gamma, L)$? Because the analysis for kernel smoothing can be done via Taylor expansions, and it becomes pretty apparent that things will work out if we can bound the (partial) derivatives. So, in essence, it makes our proofs easier!

- Define the *Sobolev class* of functions $\mathcal{W}_1(m, C)$, for an integer $m \geq 0$ and $C > 0$, to contain all m times differentiable functions $f : \mathbb{R} \mapsto \mathbb{R}$ such that

  $$\int (f^{(m)}(x))^2 dx \leq C^2$$

  (The Sobolev class $\mathcal{W}_1(m, C)$ in d dimensions can be defined similarly, where we sum over all partial derivatives of order m.)

- it is worth noting that the Sobolev $\mathcal{W}_1(m, C)$ and Holder $H_1(m, L)$ classes are equivalent in the following sense: given $\mathcal{W}_1(m, C)$ for a constant $C > 0$, there are $L_0, L_1 > 0$ such that

  $$H_1(m, L_0) \subseteq \mathcal{W}_1(m, C) \subseteq H_1(m, L_1)$$

  The first containment is easy to show; the second is far more subtle, and is a consequence of the Sobolev embedding theorem. (The same equivalences hold for the d-dimensional versions of the Sobolev and Holder spaces.)

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 of kernel smoothing is $n^{-2\alpha/(2\alpha+d)}$, which has an exponentially bad dependence on the dimension d. This is usually called the curse of dimensionality (though the term apparently originated with Bellman (1962), 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

$$f(x) = f(x_{\cdot 1}, \ldots, x_{\cdot d}) \tag{2}$$

we restrict our attention to functions of the form

$$f(x) = f_1(x_{\cdot 1}) + \cdots + f_d(x_{\cdot d}) \tag{3}$$

(Here the notation $x_{\cdot j}$ denotes the jth component of $x \in R^d$).

- As each function f_j, $j = 1, \dots, d$ is univariate, fitting an estimate of the form (3) is certainly less ambitious than fitting one of the form (2). On the other hand, the scope of (3) is still big enough that we can capture interesting (marginal) behavior in high dimensions.
- The choice of modeler (3) 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 Stone (1985) 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 $\bar{f}^{add}$ well. Assuming each component function $\bar{f}_{0,j}^{add}$ $j = 1, \ldots, d$ lies in the Holder 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

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

  Hence each component of the best additive approximation $\bar{f}^{add}$ 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.

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

$$\hat{f} = Smooth(z, y).$$

  E.g., we might choose $Smooth(\cdot, \cdot)$ 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.

- Given the inputs $x_1, \dots, x_n \in \mathbb{R}^d$, once our univariate smoother has been chosen, we initialize $\hat{f}_1, \dots, \hat{f}_d$ (say, to all to zero) and cycle over the following steps for $j = 1, \dots, d$:
  1. define $r_i = y_i - \sum_{l \neq j} \hat{f}_l(x_{il}), i = 1, \dots, n$
  2. smooth $\hat{f}_j = Smooth(x_{\cdot j}, r)$
  3. center $\hat{f}_j = \hat{f}_j - \frac{1}{n} \sum_{i=1}^n \hat{f}_j(x_{\cdot j})$.

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

$$\hat{f} + \bar{y} + \hat{f}_1(x_{\cdot 1}) + \dots + \hat{f}_d(x_{\cdot d}),$$

where $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$. Hastie & Tibshirani (1990) 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

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

  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 $Smooth_j(\cdot,\cdot), j = 1, \ldots, d$ to come from entirely different methods or giving estimates with entirely different structures
- Another extension: to capture correlations, we can perform smoothing over (small) groups of variables instead of individual variables; e.g., if we thought that variables 1, 2 might have reasonable correlation, then we could lump the backfitting steps over variables 1, 2 together and perform (say) a 2-dimensional kernel smooth, giving an estimate of the form

$$\hat{f} = \bar{y} + \hat{f}_{12}(x_{\cdot 1}, x_{\cdot 2}) + \hat{f}_3(x_{\cdot 3}) + \cdots + \hat{f}_d(x_{\cdot d})$$

Error rates

- Error rates for additive models are both kind of what you'd expect and suprising. What you'd expect: if the underlying function f_0 is additive, and we place standard assumptions on its component functions, such as $f_{0,j} \in \mathcal{W}_1(m, C)$, $j = 1, \ldots, d$, for a constant $C > 0$, a somewhat straightforward argument building on univariate minimax theory gives us the lower bound

$$\inf_{\hat{f}} \sup_{f_0 \in \oplus_{j=1}^d \mathcal{W}_1(m,C)} E\|\hat{f} - f_0\|_2^2 \gtrsim dn^{-2m/(2m+1)}.$$

  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., $f_{0,1} \in \mathcal{W}_1(2, C_1)$ 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

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

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

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

where the rougher of the two components dictates both rates? Recent work by van de Geer & Muro (2015) shows that (provided x_1, x_2 are not too correlated) it is the first case (4) that occurs

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

$$\hat{f}_1 = Smooth_1\Big(x_{\cdot 2}, (y_i - \hat{f}_2(x_{i2}))_{i=1}^n\Big)$$

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

$$y_i - \hat{f}_2(x_{i2}) = f_{0,1}(x_{i1}) + \epsilon_i + (f_{0,2}(x_{i1}) - \hat{f}_2(x_{i2})), i = 1, \ldots, n.$$

- The terms denoted $e_{2i}, i = 1, \ldots, n$ 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 van de Geer & Muro (2015) 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.)

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 (3) of the form

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

  where $S \subset \{1, \ldots, d\}$ 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 Lin & Zhang (2006), Ravikumar et al. (2009), Raskutti et al. (2012).

Variance Estimation and Confidence Bands

- Let

  $$\sigma^2(x) = Var(Y|X = x)$$

  we can estimate $\sigma^2(x)$ as follows. Let $\hat{f}$ be an estimate of the regression function. Let $e_i = Y_i - \hat{f}(X_i)$. Now apply nonparametric regression again treating e_i^2 as the response. The resulting estimator $\hat{\sigma}^2(x)$ can be shown to be consistent under some regularity conditions.

- Ideally we would also like to find random functions $l_n(x)$ and $u_n(x)$ such that

  $$P(l_n(x) \leq f(x) \leq u_n(x) \ for \ all \ x) \to 1 - \alpha$$

  For the reasons we discussed earlier with density functions, this is essentially an impossible problem.

- We can, however, still get an informal (but useful) estimate the variability of $\hat{f}(x)$. Suppose that $\hat{f}(x) = \sum_i w_i(x) Y_i$. The conditional variance is $\sum_i w_i^2(x)\sigma^2(x)$ which can be estimated by $\sum_i w_i^2(x)\hat{\sigma}^2(x)$. An asymptotic, pointwise (biased) confidence band is $\hat{f} \pm z_{\alpha/2}\sqrt{\sum_i w_i^2(x)\hat{\sigma}^2(x)}$.
- A better idea is to bootstrap the quantity

$$\frac{\sqrt{n}\sup_x |\hat{f}(x) - E\hat{f}(x)|}{\hat{\sigma}(x)}$$

to get a bootstrap quantile t_n. Then

$$\left[\hat{f}(x) - \frac{t_n \hat{\sigma}(x)}{\sqrt{n}}, \hat{f}(x) + \frac{t_n \hat{\sigma}(x)}{\sqrt{n}}\right]$$

is a bootstrap variability band.

Generalized Additive Models: Logistic Regression

- In linear modelling of binary data, the most popular approach is logistic regression which models the logit of the response probability with a linear form

$$logit P(Y=1|X) = X'\beta$$

- We can generalize the above model by replacing the linear predictor with an additive one

$$logit P(Y=1|X) = \beta_0 + \sum_{j=1}^{p} f_j(X_j). \tag{6}$$

- To estimate the model (6), we can gain some insight from the linear logistic regression methodology. Maximum likelihood is the most popular method for estimating the linear logistic model. For the present problem the log-likelihood has the form

$$l(\beta) = \sum_{i=1}^{n} \{Y_i log P(X_i) + (1 - Y_i) log(1 - P(X_i))\}, \qquad (7)$$

where $P(X_i) = exp(X_i'\beta)/(1 + exp(X_i'\beta))$. The score equations

$$\frac{\partial l(\beta)}{\partial \beta} = \sum_{i=1}^{n} X_i [Y_i - P(X_i)] = 0$$

are nonlinear in the parameters β and consequently one has to find the solution iteratively.

- The Newton-Raphson iterative method can be expressed in an appealing form. Given the current estimate $\hat{\beta}$, we can estimate the probabilities $P(X_i)$ by $p_i = exp(X_i'\hat{\beta})/(1 + exp(X_i'\hat{\beta}))$. We form the linearized response

$$Z_i = X_i'\hat{\beta} + (Y_i - p_i)/(p_i(1 - p_i)),$$

  where the quantity Z_i represents the first-order Taylor's series approximation to $logit(P(X_i))$ about the current estimate p_i.

- Denote $u_i = (Y_i - p_i)/(p_i(1 - p_i))$. If $\hat{\beta}$ and hence p_i are fixed, the variance of Z_i is $1/(p_i(1 - p_i))$, and hence we choose the weights $w_i = p_i(1 - p_i)$. Alternatively, we can verify that

$$E[u_i|X_i] = 0, \quad E[u_i^2|X_i] = \frac{1}{P(X_i)(1 - P(X_i))}$$

  in the extreme case where $p_i = P(X_i)$.

- So when p_i approximate $P(X_i)$, we expect that

  $$E[u_i|X_i] \approx 0, \quad E[u_i^2|X_i] \approx \frac{1}{P(X_i)(1-P(X_i))} \approx \frac{1}{p_i(1-p_i)}$$

  Consequently, a new $\hat{\beta}$ can be obtained by weighted linear regression of Z_i on X_i with weights $w_i = p_i(1-p_i)$. This is repeated until $\hat{\beta}$ converges.

- **Algorithm** The above iterative algorithm lends itself ideally to the generalized additive model in (6). Define

  $$Z_i = \tilde{\beta} + \sum_{j=1}^{p} \tilde{f}_j(X_{ij}) + (Y_i - p_i)/(p_i(1-p_i))$$

  where $(\tilde{\beta}, \tilde{f}_j)$ are the current estimates for the additive model components

- and

$$p_i = \frac{exp(\tilde{\beta} + \sum_{j=1}^{p} \tilde{f}_j(X_{ij}))}{1 + exp(\tilde{\beta} + \sum_{j=1}^{p} \tilde{f}_j(X_{ij}))}$$

- Define the weights

$$w_i = p_i(1 - p_i)$$

  The new estimates of β_0 and $f_j(j = 1, \ldots, p)$ are computed by fitting a weighted additive model to Z_i.

- Of course, this additive model fitting procedure is iterative as well. Fortunately, the functions from the previous step are good starting values for the next step. This procedure is called the *local-scoring algorithm* in the literature. The new estimates from each local scoring step are monitored and the iterations are stopped when their relative change is negligible.

Additive Partially Linear Models

- A typical additive partially linear model is of the form

$$Y_i = X_i' \beta_0 + g_1(Z_{i1}) + \cdots + g_L(Z_{iL}) + u_i, \tag{8}$$

  where $E[u_i|X_i, Z_{i1}, \dots, Z_{iL}] = 0$, X_i is a $p \times 1$ vector of random variables that does not contain a constant term, β_0 is a $p \times 1$ vector of unknown parameter; Z_{il} is of dimension q_l ($q_l \geq 1, l = 1, \dots, L$); $g_l(\cdot), l = 1, \dots, L$, are unknown smooth functions.

- We introduce two methods to estimate additive partially linear models. One is the series method of Li (2000, Intl Economic Review)) and the other is the kernel method of Fan and Li (2003, Statistica Sinica).

Li's series method

- Denote $g(z) = \sum_{l=1}^{L} g_l(z_l)$, and a series estimate of $g(z)$ by

$$\hat{g}(z) = p^K(z)'\hat{\alpha}$$

  and suppose $\hat{g}$ can approximate g arbitrarily well in the mean squared error sense.

- Define $P = (p^K(Z_1), \ldots, p^K(Z_n))'$ and $M = P(P'P)^- P'$. Let $\tilde{A} = MA$ for any n row matrix A. If we premultiply both sides of (8) by M, then we have

$$\tilde{Y} = \tilde{X}\beta_0 + \tilde{g} + \tilde{u} \tag{9}$$

  Subtracting (9) from (8) gives

$$Y - \tilde{Y} = (X - \tilde{X})\beta_0 + (g - \tilde{g}) + (u - \tilde{u})$$

- So we can estimate β_0 by regressing $Y - \tilde{Y}$ on $X - \tilde{X}$ to obtain

$$\hat{\beta} = [(X - \tilde{X})'(X - \tilde{X})]^{-}(X - \tilde{X})'(Y - \tilde{Y})$$

- After obtaining $\hat{\beta}$ we can estimate $g(z)$ by $\hat{g}(z) = p^K(z)'\hat{\alpha}$, where

$$\hat{\alpha} = (P'P)^{-}P'(Y - X\hat{\beta}).$$

- **Theorem** Under some regularity conditions, we have

$$\sqrt{n}(\hat{\beta} - \beta_0) \rightsquigarrow N(0, \Phi^{-1}\Psi\Phi^{-1})$$

Fan and Li's Kernel Method

- Like Fan and Li(2003), we consider the additive partially linear model

$$Y_i = \beta_0 + X_i'\beta + g_1(Z_{i1}) + \cdots + g_q(Z_{iq}) + u_i, \qquad (10)$$

  where $E[u_i|X_i, Z_{i1}, \ldots, Z_{iq}] = 0$, X_i is a $p \times 1$ vector of random variables that does not contain a constant term, β_0 is a scalar parameter, β is a $p \times 1$ vector of unknown parameter; Z_{il}'s are univariate continuous random variables, and $g_l(\cdot), l = 1, \ldots, q$, are unknown smooth functions.

- Let $Z_{i,-l} = (Z_{i1}, \ldots, Z_{i,l-1}, Z_{i,l+1}, \ldots, Z_{iq})$ where Z_{il} is removed from $Z_i = (Z_{i1}, \ldots, Z_{iq})$.

- We can rewrite (10) as

$$Y_i = \beta_0 + X_i'\beta + g_l(Z_{il}) + G_{-l}(Z_{i,-l}) + u_i. \qquad (11)$$

- Fan, Härdle, and Mammen (1998) consider the case where X_i is a $p \times 1$ vector of discrete variables and suggest two ways of estimating model (11). In neither method did they make full use of the information that X_i enters the regression function linearly. Motivated by this observation, Fan and Li (2003) consider a two-stage estimation procedure which applies to the case where X_i contains both discrete and continuous elements and makes full use of the information that X_i enters the regression function linearly.
- For $l = 1, \ldots, q$, define

$$\begin{aligned}\xi(z_l, z_{-l}) =& E[Y_i | Z_{il} = z_l, Z_{i,-l} = z_{-l}], \xi_l(z_l) = E[\xi(z_l, Z_{i,-l})], \\ \eta(z_l, z_{-l}) =& E[X_i | Z_{il} = z_l, Z_{i,-l} = z_{-l}], \eta_l(z_l) = E[\eta(z_l, Z_{i,-l})],\end{aligned}$$

- Denote $\xi_{il} = \xi_l(Z_{il})$ and $\eta_{il} = \eta_l(Z_{il})$. Then taking conditional expectations on both sides of (11) gives

$$\xi(z_l, z_{-l}) = \beta_0 + \eta(z_l, z_{-l})'\beta + g_l(z_l) + G_{-l}(z_{-l}). \tag{12}$$

- where we have used the identification condition that $E[G_{-l}(Z_{i,-l})] = 0$. Replacing z_l in (12) by Z_{il} and then summing both sides of (12) gives

$$\sum_{l=1}^{q} \xi_{il} = q\beta_0 + \sum_{l=1}^{q} \eta_{il}'\beta + \sum_{l=1}^{q} g_l(Z_{il}). \tag{13}$$

- Substracting (13) from (10), we get

$$Y_i - \sum_{l=1}^{q} \xi_{il} = (1-q)\beta_0 + (X_i - \sum_{l=1}^{q} \eta_{il})'\beta + u_i. \tag{14}$$

- Let $\mathcal{Y}_i = Y_i - \sum_{l=1}^{q} \xi_{il}$ and $\mathcal{X}_i = (1, (X_i - \sum_{l=1}^{q} \eta_{il})')$. Then in vector notation we can write (14) as

$$\mathcal{Y} = \mathcal{X}\delta + U \tag{15}$$

where $\delta = (\alpha_0, \beta')'$ with $\alpha_0 = (1-q)\beta_0$.

- We can apply OLS regression to (15) to obtain

$$\bar{\delta} = \left(\begin{array}{c} \bar{\alpha}_0 \\ \bar{\beta} \end{array} \right) = (\mathcal{X}'\mathcal{X})^{-1}\mathcal{X}'\mathcal{Y} = \delta + (\mathcal{X}'\mathcal{X})^{-1}\mathcal{X}'U \tag{16}$$

Under standard conditions, we can show that $\bar{\delta}$ converges to δ at the parametric $\sqrt{n}$-rate.

- Nevertheless, $\bar{\delta}$ is an infeasible estimator because it depends on the unknown quantities $\sum_{l=1}^{q} \xi_{il}$ and $\sum_{l=1}^{q} \eta_{il}$. To obtain a feasible estimator of δ, we need to replace these unknown quantities by their consistent estimates. A consistent estimator of $\xi_{il} = \xi_l(Z_{il})$ is given by

$$\begin{aligned}\hat{Y}_{il} &= \frac{1}{n} \sum_{j=1}^{n} \frac{\sum_{k=1}^{n} Y_k \mathcal{K}_{h_l}(Z_{ik} - Z_{il}) L_{h_{-l}}(Z_{k,-l} - Z_{j,-l})}{\sum_{s=1}^{n} \mathcal{K}_{h_l}(Z_{is} - Z_{il}) L_{h_{-l}}(Z_{s,-l} - Z_{j,-l})} \\ &= \sum_{k=1}^{n} Y_k W_{il,k}\end{aligned}$$

where the definition for $W_{il,k}$ is clear, $\mathcal{K}$ and L are production kernels. Fan and Li (2003) use the leave-one-out method to obtain $W_{il,k}$ which can only simplify the proofs but does not change the asymptotic results.

- Similarly, a consistent estimator of $\eta_{il} = \eta(Z_{il})$ is given by

$$\hat{X}_{il} = \sum_{k=1}^{n} X_k W_{il,k}. \tag{17}$$

- Let $\hat{\mathcal{Y}}_i = Y_i - \sum_{l=1}^{q} \hat{Y}_{il}$ and $\hat{\mathcal{X}}_i = (1, (X_i - \sum_{l=1}^{q} \hat{X}_{il})$.
- Fan and Li (2003) estimate δ by

$$\hat{\delta} = \left(\begin{array}{c} \hat{\alpha}_0 \\ \hat{\beta} \end{array} \right) = (\hat{\mathcal{X}}'\hat{\mathcal{X}})^{-1}\hat{\mathcal{X}}'\hat{\mathcal{Y}} = \Big(\sum_{i=1}^{n} \hat{\mathcal{X}}_i'\hat{\mathcal{X}}_i'\mathbf{1}_i \Big)^{-1} \sum_{i=1}^{n} \hat{\mathcal{Y}}_i \mathbf{1}_i \tag{18}$$

where $\mathbf{1}_i = \mathbf{1}(Z_i \in \prod_{l=1}^{q}[c_l + b_n, d_l - b_n])$ with $b_n = ch^{\epsilon}$, $c > 0$, $0 < \epsilon < 1$, $h = \max\{h_l, h_{-l}\}$.

- **Theorem** Under some regularity conditions, we have

$$\sqrt{n}(\hat{\beta} - \beta) \rightsquigarrow N(0, \Phi^{-1}\Psi\Phi^{-1})$$

- Given the $\sqrt{n}$-consistent estimator $\hat{\beta}$, the intercept term β_0 can be $\sqrt{n}$-consistently estimated by

$$\hat{\beta}_0 = \bar{Y} - \bar{X}'\hat{\beta}.$$

Case study

- As an example, we'll visit the California house price data.

```
calif = read.table("cadata.dat",header=TRUE)
```

- Fitting a linear model is very fast. Here are the summary statistics:

```
    Call:
lm(formula = log(MedianHouseValue) ~ ., data = calif)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.5180 -0.2038  0.0016  0.1949  3.4641 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)    -1.180e+01  3.059e-01 -38.570  < 2e-16
MedianIncome    1.782e-01  1.639e-03 108.753  < 2e-16
MedianHouseAge  3.261e-03  2.111e-04  15.446  < 2e-16
TotalRooms     -3.186e-05  3.855e-06  -8.265  < 2e-16
TotalBedrooms   4.798e-04  3.375e-05  14.215  < 2e-16
Population     -1.725e-04  5.277e-06 -32.687  < 2e-16
Households      2.493e-04  3.675e-05   6.783 1.21e-11
Latitude       -2.801e-01  3.293e-03 -85.078  < 2e-16
Longitude      -2.762e-01  3.487e-03 -79.212  < 2e-16

Residual standard error: 0.34 on 20631 degrees of freedom
Multiple R-squared:  0.6432,    Adjusted R-squared:  0.643 
F-statistic:  4648 on 8 and 20631 DF,  p-value: < 2.2e-16
```

- The following Figure 1 plots the predicted prices, ± 2 standard errors, against the actual prices. The predictions are not all that accurate — the RMS residual is 0.340 on the log scale (i.e., 41%), and only 3.3% of the actual prices fall within the prediction bands. On the other hand, they are quite precise, with an RMS standard error of 0.0071 (i.e., 0.71%). This linear model is pretty thoroughly converged.

```
predictions = predict(linfit,se.fit=TRUE)
plot(calif$MedianHouseValue,exp(predictions$fit),cex=0.1,
    xlab="Actual price",ylab="Predicted")
segments(calif$MedianHouseValue,exp(predictions$fit-2*predictions$se.fit),
    calif$MedianHouseValue,exp(predictions$fit+2*predictions$se.fit),
    col="grey")
abline(a=0,b=1,lty=2)
```

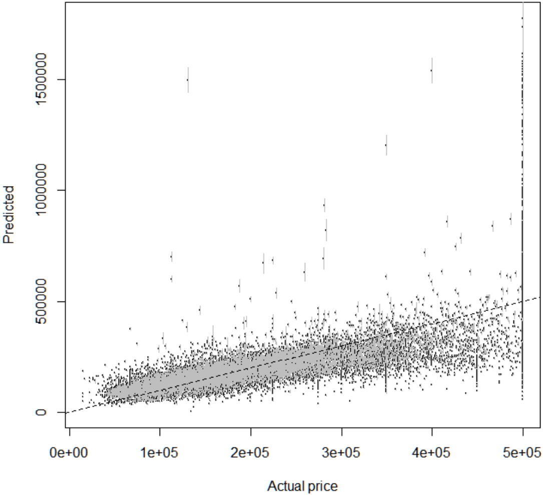


Figure 1: Actual median house values (horizontal axis) versus those predicted by the linear model (black dots), plus or minus two standard errors (grey bars). The dashed line shows where actual and predicted prices would be equal. Note that I've exponentiated the predictions so that they're comparable to the original values.

- Next, we'll fit an additive model, using the gam function from the mgcv package; this automatically sets the bandwidths using a fast approximation to leave-one-out CV called generalized cross-validation, or GCV.

```
> require(mgcv)
> system.time(addfit <- gam(log(MedianHouseValue) ~ s(MedianIncome)
+ + s(MedianHouseAge) + s(TotalRooms)
+ + s(TotalBedrooms) + s(Population) + s(Households)
+ + s(Latitude) + s(Longitude), data=calif))
   user  system elapsed
   5.03    0.25    5.29
```

- The *s()* terms in the gam formula indicate which terms are to be smoothed — if we wanted particular parametric forms for some variables, we could do that as well. The smoothing here is done by splines, and there are lots of options for controlling the splines, if you know what you're doing.

- Figure 2 compares the predicted to the actual responses. The RMS error has improved (0.29 on the log scale, or 33%, with 9.5% of observations falling with ± 2 standard errors of their fitted values), at only a fairly modest cost in precision (the RMS standard error of prediction is 0.016, or 1.6%). Figure 3 shows the partial response functions.
- It seems silly to have latitude and longitude make separate additive contributions here; presumably they interact. We can just smooth them together

```
addfit2 <- gam(log(MedianHouseValue) ~ s(MedianIncome) + s(MedianHouseAge)
 + s(TotalRooms) +s(TotalBedrooms) + s(Population) + s(Households)
 + s(Longitude,Latitude), data=calif)
```

  This gives an RMS error of $\pm 31\%$ (with 11% coverage), with no decrease in the precision of the predictions (at least to two figures).

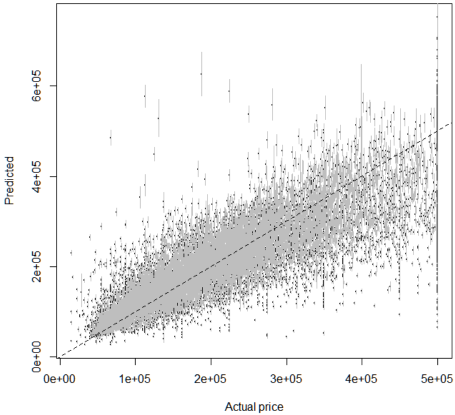
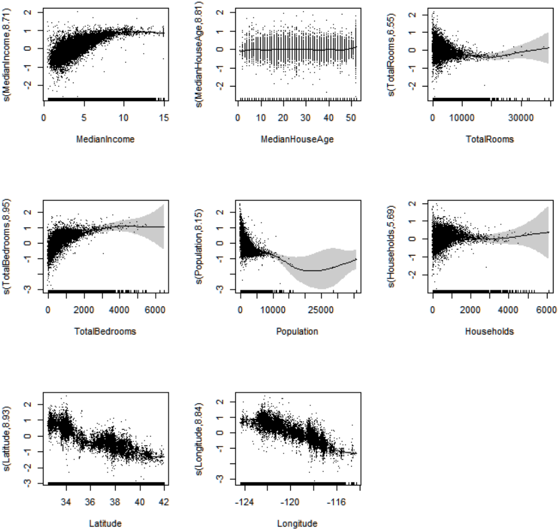


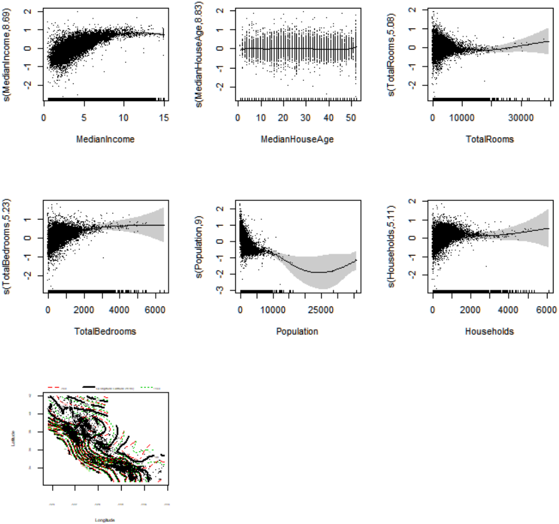
Figure 2: Actual versus predicted prices for the additive model, as in Figure 1.

```
predictions = predict(addfit,se.fit=TRUE)
plot(calif$MedianHouseValue,exp(predictions$fit),cex=0.1,
xlab="Actual price",ylab="Predicted")
segments(calif$MedianHouseValue,exp(predictions$fit-2*predictions$se.fit),
calif$MedianHouseValue,exp(predictions$fit+2*predictions$se.fit),
col="grey")
abline(a=0,b=1,lty=2)
```



```
plot(addfit,scale=0,se=2,shade=TRUE,resid=TRUE,pages=1)
```

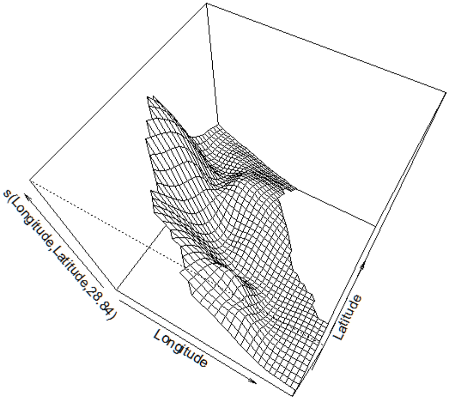
Figure 3: The estimated partial response functions for the additive model, with a shaded region showing ± 2 standard errors, and dots for the actual partial residuals. The tick marks along the horizontal axis show the observed values of the input variables (a rug plot); note that the error bars are wider where there are fewer observations. Setting pages=0 (the default) would produce eight separate plots, with the user prompted to cycle through them. Setting scale=0 gives each plot its own vertical scale; the default is to force them to share the same one. Finally, note that here the vertical scale is logarithmic.



```
plot(addfit2,scale=0,se=2,shade=TRUE,resid=TRUE,pages=1)
```

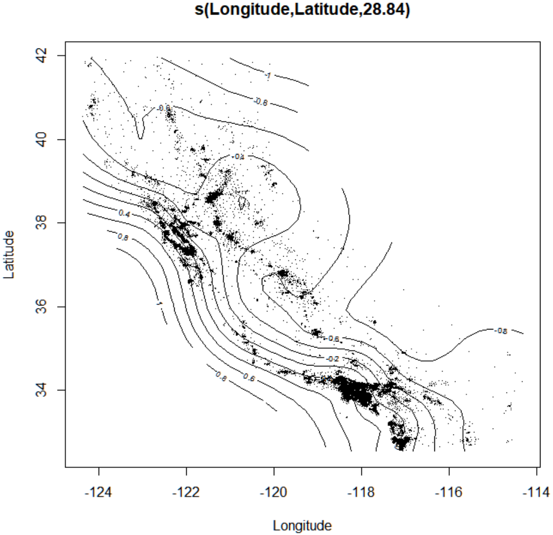
Figure 4: Partial response functions and partial residuals for addfit2, as in Figure 3. See subsequent figures for the joint smoothing of longitude and latitude, which here is an illegible mess.

- Figures 5 and 6 show two different views of the joint smoothing of longitude and latitude. In the perspective plot, it's quite clear that price increases specifically towards the coast, and even more specifically towards the great coastal cities. In the contour plot, one sees more clearly an inward bulge of a negative, but not too very negative, contour line (between -122 and -120 longitude) which embraces Napa, Sacramento, and some related areas, which are comparatively more developed and more expensive than the rest of central California, and so more expensive than one would expect based on their distance from the coast and San Francisco.



```
plot(addfit2,select=7,phi=60,pers=TRUE)
```

Figure 5: The result of the joint smoothing of longitude and latitude.



```
plot(addfit2,select=7,se=FALSE)
```

Figure 6: The result of the joint smoothing of longitude and latitude. Setting se=TRUE, the default, adds standard errors for the contour lines in multiple colors. Again, note that these are log units.

- The fact that the prediction intervals have such bad coverage is partly due to their being based on Gaussian approximations. Still, ± 2 standard errors should cover at least 25% of observations, which is manifestly failing here.
- This suggests substantial remaining bias. One of the standard strategies for trying to reduce such bias is to allow more interactions. We will see automatic ways of doing this, in later lectures, where we can still get some sort of interpretation of results.
- We could, of course, just use a completely unrestricted nonparametric regression — going to the opposite extreme from the linear model. I'll use npreg from the np package to fit a Nadaraya-Watson regression, using its built-in function npregbw to pick the bandwidths.

```
library(np)
system.time(calif.bw <- npregbw(log(MedianHouseValue) ~., data=calif,type="ll"))
```