Additive Models

Advanced Methods for Data Analysis (36-402/36-608)

Spring 2014

1 Nonparametric smoothing in multiple dimensions

1.1 Nonparametric review in one dimension

- Assume for know that $X \in \mathbb{R}$. A model of the form

  $$Y = r(X) + \varepsilon,$$

  where we don't make any assumptions about the form of the true underlying regression function $r(x) = \mathbb{E}(Y|X = x)$, is called a *nonparametric* regression model. Contrast this with a *parametric* regression model, e.g., linear regression, in which we assume that $r(X) = \beta_0 + \beta X$, so that the model becomes

  $$Y = \beta_0 + \beta X + \varepsilon.$$

  In this case, we can simply employ linear regression to estimate β_0, β; but in the first case, when we don't assume a particular (i.e., parametric form) for r, we turn to regression smoothers that are much more flexible to adapt to unknown trends, like k-nearest-neighbors, kernel regression, or smoothing splines

- When thinking about parametric versus nonparametric, it's important to remember the *bias-variance tradeoff*. Generally speaking, a parametric estimator (e.g., linear regression) will have a lower variance than a nonparametric one (e.g., k-nearest-neighbors, kernel regression, or smoothing splines), because it is more restrictive. Meanwhile, the bias depends on the true underlying model. Nonparametric estimators are generally flexible enough that they will have a low bias for a wide range of underlying regression functions, but a parametric estimator (such as linear regression) will only have a low bias if the parametric assumption is approximately correct (i.e., the true model is approximately linear), and can otherwise suffer from high bias

- As we know, expected test error is composed of bias and variance, so both of these quantities are important for predictive performance. In univariate smoothing, i.e., when $X \in \mathbb{R}$, it can often be the case that our considerations for the bias dominate those for the variance, and so we favor nonparametric methods for fitting[1]

1.2 Multiple dimensions and the curse of dimensionality

- When $X \in \mathbb{R}^p$, i.e., we have p predictors instead of 1, extending the linear model is straightforward; as you well know, writing $X = (X_1, \ldots X_p)$, the multi-dimensional linear model is

  $$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon, \tag{1}$$

[1]Note: this is a very rough reflection of a general trend, and should not be interpreted as universal in any sense!

and estimation proceeds by linear regression, just as in the univariate case. But, given the multi-dimensional nonparametric model

$$Y = r(X_1, \dots X_p) + \varepsilon, \tag{2}$$

how do we construct fully nonparametric estimates for r? Actually, this is possible for each of the methods we discussed: k-nearest-neighbors, kernel regression, and smoothing splines. The k-nearest-neighbors and kernel estimates are really just local averaging procedures, and naturally extend to the setting of p-dimensional predictor variables, as we discussed in the lecture notes. Smoothing splines do as well, but their extension is not nearly as obvious, and is called *thin-plate splines*, something that we will not cover

- In multiple dimensions, the variance of nonparametric estimators becomes a real problem. Nonparametric methods typically suffer from variance that scales exponentially with the number of predictors p; remember, this means that the test error for such methods also scales exponentially with p, which is an awful trend! This is called the *curse of dimensionality*[2]

- On the other hand, parametric methods like linear regression typically have a variance that grows merely linearly with p; but their bias can degrade quickly with increasing p, too. This begs the question: is there some middle ground?

2 Additive models

2.1 The additive compromise

- Enter additive models, a framework that lies somewhere in between the fully parametric and nonparametric settings, (1) and (2). Starting with the linear model in (1), we could simply replace each linear term $X_i\beta_i$ with a general, nonlinear one $r_i(X_i)$, yielding the *additive model*

$$Y = \beta_0 + r_1(X_1) + \ldots + r_p(X_p) + \varepsilon. \tag{3}$$

  This is in a sense simpler than the fully nonparametric model (2), because of the restriction that r decompose into a sum of univariate regression functions over the variables

- Without any restrictions on the functions $r_1, \dots r_p$, the model in (3) is not identifiable, so we usually assume without a loss of generality that

$$\mathbb{E}(Y) = \beta_0, \quad \mathbb{E}\big(r_j(X_j)\big) = 0, \quad j = 1, \dots p.$$

- Estimation in an additive model is actually very simple: the beauty of it is that we can just rely on univariate smoothing, which we already know a lot of about! More on this in the next section

- Additive estimates tend to balance the strengths of the fully nonparametric and parametric estimates. I.e., additive estimates tend have a lower variance than fully nonparametric ones, and can have a lower bias than parametric ones

- The main downside: by restricting the estimate to be additive, we miss potential interactions between variables. However, like in linear regression, we can manually add interactions terms like $r_{ij}(X_i, X_j)$ and $r_{ijk}(X_i, X_j, X_k)$, etc. to the model if we desire or deem it appropriate

[2]Strictly speaking, the curse of dimensionality describes a slightly different point on a fundamental lower bound for estimation in fully nonparametric settings, for growing p, but this is highly related nonetheless

2.2 Backfitting

- Given pairs $(x_i, y_i) \in \mathbb{R}^p \times \mathbb{R}$, $i = 1, \ldots n$, with each $x_i = (x_{i1}, \ldots x_{ip}) \in \mathbb{R}^p$, the additive model becomes
$$y_i = \beta_0 + r_1(x_{i1}) + \ldots + r_p(x_{ip}) + \epsilon_i, \quad i = 1, \ldots n, \tag{4}$$
subject to the same identifiability assumptions $\mathbb{E}(y_i) = \beta_0$, and $\mathbb{E}(r_j(x_{ij})) = 0$ for $j = 1, \ldots p$.

- Computing an additive estimate from the model (4) is now done by a simple procedure called *backfitting*. We first let $\hat{\beta}_0 = \bar{y} = \frac{1}{n}\sum_{i=1}^n y_i$. The idea is for the rest is just to cycle through estimating each of $r_1, \ldots r_p$ one at a time, by univariate smoothing, and repeat this until convergence

  To be more concrete: write $S(z, y)$ to denote a univariate smoother constructed from inputs $z = (z_1, \ldots z_n)$ and outputs $y = (y_1 \ldots y_n)$. This could be, e.g., a linear regression estimate from the pairs $(z_i, y_i) \in \mathbb{R} \times \mathbb{R}$, $i = 1, \ldots n$, or it could be a kernel regression or a smoothing spline estimate. Whatever the choice of smoother, it returns an estimated regression function (this is a function of the input variable). Then backfitting repeats the following loop until convergence:

  - For $j = 1, \ldots p$:
    - Build the jth partial residual,
    $$y^{(j)} = y - \hat{\beta}_0 - \sum_{\ell \neq j} \hat{r}_\ell(x_{\cdot\ell}).$$
    - Update $\hat{r}_j$ by smoothing the jth partial residual on the jth variable,
    $$\hat{r}_j = S\Big(x_{\cdot j}, y^{(j)}\Big).$$
    - Center $\hat{r}_j$,
    $$\hat{r}_j = \hat{r}_j - \frac{1}{n}\sum_{i=1}^n \hat{r}_j(x_{ij}).$$

  In the above, $x_{\cdot j} = (x_{1j}, \ldots x_{nj}) \in \mathbb{R}^n$ denotes the n measurements of the jth input variable. We stop repeating this loop when the estimated functions $\hat{r}_j$, $j = 1, \ldots p$ don't change much from one cycle to the next

- The intuition for backfitting just comes from rearranging (4). Supposing that we fixed all of the underlying regression functions except the jth one (and the intercept) at the estimates $\hat{r}_\ell$, $\ell \neq j$ (and $\hat{\beta}_0$), the model becomes
$$y_i - \hat{\beta}_0 - \sum_{\ell \neq j} \hat{r}_j(x_{ij}) = r_j(x_{ij}) + \epsilon_i, \quad i = 1, \ldots n.$$
To estimate r_j, therefore, we can just treated the left-hand side above as the outcome, and regress this outcome on $x_{\cdot j}$, which is exactly what we do in each iteration of backfitting. (The post-centering step is just done to preserve the zero mean condition for model identifiability)

- Note: there's no reason to use the same univariate smoother in every iteration of backfitting; we can, if we think it is appropriate, use different types of smoothers for different variables. The default is probably to use smoothing splines for each variable, where we either specify the degrees of freedom of the fit ahead of time, or choose it by (generalized) cross-validation in each regression