STAT 425: Introduction to Nonparametric Statistics **Winter 2018**

Lecture 11: Regression: Penalized Approach

Instructor: Yen-Chi Chen

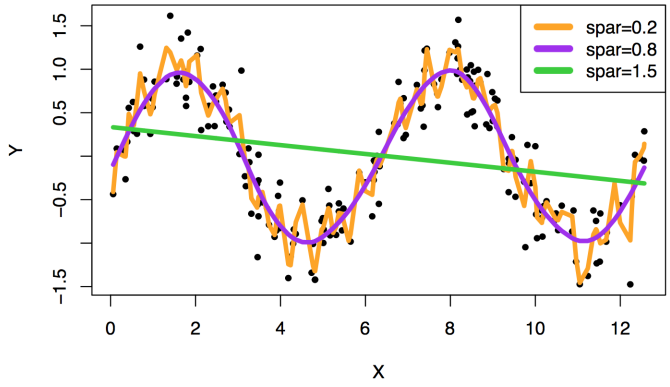
11.1 Penalized Methods: Introduction

In the regression tree, we talk about the case that we want to select the number of leaves M based on the following criterion:

$$C_{\lambda,n}(M) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{m}(X_i))^2}_{\text{fitting to the data}} + \underbrace{\lambda M}_{\text{penalty on the complexity}}. \tag{11.1}$$

It turns out that this type of criterion is very general in regression analysis because we want to avoid the problem of **overfitting**.

The overfitting means that you fit a too complex model to the data so that although the fitted curve is close to most of the observations, the actual prediction is very bad. For instance, the following picture shows the fitted result using a smoothing/cubic spline (here the quantity spar is related to λ):



This data is generated from a sine function plus a small noise. When λ is too small (orange curve), we fit a very complicated model to the data, which does not capture the right structure. On the other hand, when λ is too large (green curve), we fit a too simple model (a straight line), which is also bad in predicting the actual outcome. When λ is too small, it is called **overfitting** (orange curve) whereas when λ is too large, it is called **underfitting** (green curve). In fact, overfitting is similar to undersmoothing and underfitting is similar to oversmoothing. It regression analysis, people prefer to use overfitting and underfitting to describe the outcome and in density estimation, people prefer to use undersmoothing and oversmoothing.

Finding a regression estimator using a criterion with a fitting to the data plus a penalty on the complexity is called a penalized regression. In the case of regression tree, let

$$\mathcal{M}_{\mathsf{Tree}} = \{\text{all possible regression trees}\}$$

be the collection of all possible regression trees. We can rewrite equation (11.1) as

$$\widehat{m}_{\mathsf{Tree}} = \underset{m\in\mathcal{M}_{\mathsf{Tree}}}{\mathsf{argmin}} \frac{1}{n}\sum_{i=1}^{n}(Y_i - m(X_i))^2 + \mathcal{P}_\lambda(m),$$

where $\mathcal{P}_\lambda(m) = \lambda \times$ number of regions in m. Thus, with the penalty on the number of regions, the regression tree is a penalized regression approach.

For any penalized regression approach, there is an abstract expression for them:

$$\widehat{m} = \underset{m\in\mathcal{M}}{\mathsf{argmin}} \frac{1}{n}\sum_{i=1}^{n}(Y_i - m(X_i))^2 + \mathcal{P}_\lambda(m), \tag{11.2}$$

where $\mathcal{M}$ is a collection of regression estimators and $\mathcal{P}_\lambda(m)$ is the amount of penalty imposed for a regression estimator $m \in \mathcal{M}$ and λ is a tuning parameter that determines the amount of penalty. The penalized regression always have a fitting part (e.g., $\frac{1}{n}\sum_{i=1}^{n}(Y_i - m(X_i))^2$) and a penalized part (also called regularized part) $\mathcal{P}_\lambda(m)$. The fitting part makes sure the model fits the data well while the penalized part guarantees that the model is not too complex. Thus, the penalized regression often leads to a simple model with a good fitting to the data.

11.2 Spline

Smoothing spline is a famous example in penalized regression methods. Here we consider the case of univariate regression (i.e., the covariate X is univariate or equivalently, $d = 1$) and focus on the region where the covariates belongs to $[0, 1]$. Namely, our data is $(X_1, Y_1), \cdots, (X_n, Y_n)$ with $X_i \in [0, 1] \subset \mathbb{R}$ for each i.

Let $\mathcal{M}_2$ denotes the collection of all univariate functions with second derivative on $[0, 1]$. The cubic (smoothing) spline finds an estimator

$$\widehat{m} = \underset{m\in\mathcal{M}_2}{\mathsf{argmin}} \frac{1}{n}\sum_{i=1}^{n}(Y_i - m(X_i))^2 + \lambda\int_0^1 |m''(x)|^2 dx. \tag{11.3}$$

In the cubic spline the penalty function is $\lambda\int_0^1 |m''(x)|^2 dx$, which imposes restriction on the smoothness – the curve $m(x)$ cannot change too drastically otherwise the second derivatives will be large. Thus, the cubic spline leads to a smooth curve but fits to the data well.

Why the estimator $\widehat{m}$ is called a cubic spline? This is because it turns out that $\widehat{m}$ is a piecewise polynomial function (spline) with degree of 3. Namely, there exists knots $\tau_1 < \cdots < \tau_K$ such that for $x \in (\tau_k, \tau_{k+1})$,

$$\widehat{m}(x) = \gamma_{0,k} + \gamma_{1,k}x + \gamma_{2,k}x^2 + \gamma_{3,k}x^3,$$

for some $\gamma_{0,k}, \cdots, \gamma_{3,k}$ with restriction that $\widehat{m}(x)$ has continuous second derivatives at each knot. In the case of cubic spline, it turns out that the knots are just data points.

The representation of a cubic spline is often done using some basis function. Here we will introduce a simple basis called the truncated power basis. Let $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$ be the ordered statistics of $X_1, \cdots, X_n$. In the cubic spline, the knots are

$$\tau_1 = X_{(1)}, \tau_2 = X_{(2)}, \cdots, \tau_n = X_{(n)}.$$

The truncated power basis uses a collection of functions

$$h_1(x) = 1, h_2(x) = x, h_3(x) = x^2, h_4(x) = x^3,$$

and

$$h_j(x) = (x - \tau_{j-4})_+^3, \quad j = 5, 6, \cdots, n+4,$$

where $(x)_+ = \max\{x, 0\}$. Then the estimator $\widehat{m}$ can be written as

$$\widehat{m}(x) = \sum_{j=1}^{n+4} \widehat{\beta}_j h_j(x),$$

for some properly chosen $\widehat{\beta}_j$.

How do we compute $\widehat{\beta}_1, \cdots, \widehat{\beta}_{n+4}$? They should be chosen using equation (11.3). Here how we will compute it. Define an $n \times (n+4)$ matrix $\mathbb{H}$ such that

$$\mathbb{H}_{ij} = h_j(X_i)$$

and an $(n+4) \times (n+4)$ matrix Ω with

$$\Omega_{ij} = \int_0^1 h_i''(x) h_j''(x) dx.$$

In this case, we define $m(x) = \sum_{j=1}^{n+4} \beta_j h_j(x)$ so the criterion in the right-hand side of (11.3) becomes

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}(Y_i - m(X_i))^2 + \lambda \int_0^1 |m''(x)|^2 dx \\
&\qquad = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{n+4}\beta_j h_j(X_i)\right)^2 + \lambda \int_0^1 \left(\sum_{j=1}^{n+4}\beta_j h_j(x)\right)\left(\sum_{\ell=1}^{n+4}\beta_\ell h_\ell(x)\right) dx \\
&\qquad = \|\mathbb{Y} - \mathbb{H}\beta\|^2 + \lambda \beta^T \Omega \beta \\
&\qquad = R_n(\beta)
\end{aligned}$$

where $\mathbb{Y} = (Y_1, \cdots, Y_n)$ and $\beta = (\beta_1, \cdots, \beta_{n+4})$. Thus,

$$\widehat{\beta} = \mathsf{argmin}_\beta R_n(\beta) = (\mathbb{H}^T\mathbb{H} + \lambda\Omega)^{-1}\mathbb{H}^T\mathbb{Y}.$$

Given a point x, let $H(x) = (h_1(x), h_2(x), \cdots, h_{n+4}(x))$ be an $(n+4)$-dimensional vector. Then the predicted value $\widehat{m}(x)$ has a simple form:

$$\widehat{m}(x) = H^T(x)\widehat{\beta} = H^T(x)(\mathbb{H}^T\mathbb{H} + \lambda\Omega)^{-1}\mathbb{H}^T\mathbb{Y} = \sum_{i=1}^{n} \ell_i(x) Y_i,$$

where

$$\ell_i(x) = H^T(x)(\mathbb{H}^T\mathbb{H} + \lambda\Omega)^{-1}\mathbb{H}^T e_i,$$

with $e_i = (0, 0, \cdots, 0, \underbrace{1}_{i\text{-th coordinate}}, 0, \cdots, 0)$ is the unit vector in the i-th coordinate. Therefore, again the cubic spline is a linear smoother.

Note that when the sample size n is large, the spline estimator behaves like a kernel regression in the sense that

$$\ell_i(x) \approx \frac{1}{p(X_i)h(X_i)} K\left(\frac{X_i - x}{h(X_i)}\right)$$

and

$$h(x) = \left(\frac{\lambda}{np(x)}\right)^{1/4}, \quad K(x) = \frac{1}{2}\exp\left(-\frac{|x|}{\sqrt{2}}\right)\sin\left(\frac{|x|}{\sqrt{2}} + \frac{\pi}{4}\right).$$

Remark.

- **Regression spline.** In the case where we use the spline basis to do regression but without a penalty and use fewer number of knots (and we allow the knots to be at non data points), the resulting estimator is called a regression spline. Namely, a regression spline is an estimator of the form $\widehat{m}(x) = \sum_{j=1}^{M} \widehat{\beta}_j h_j(x)$, where $\widehat{\beta}_1, \cdots, \widehat{\beta}_M$ are determined by minimizing

  $$\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \sum_{j=1}^{M}\beta_j h_j(X_i)\right)^2.$$

  Using our notations, the regression spline can be written as

  $$\widehat{m}(x) = H^T(x)\widehat{\beta} = H^T(x)(\mathbb{H}^T\mathbb{H})^{-1}\mathbb{H}^T\mathbb{Y}.$$

- **B-spline basis.** There are other basis that can be used in constructing a spline estimator. One of the most famous basis is the B-spline basis. This basis is defined through a recursive way so we will not go to the details here. If you are interested in, you can check https://cran.r-project.org/web/packages/crs/vignettes/spline_primer.pdf. The advantage of using a B-spline basis is the computation.

- **M-th order spline.** There are higher order spline. If we modify the optimization criterion to

  $$\frac{1}{n}\sum_{i=1}^{n}(Y_i - m(X_i))^2 + \lambda\int_0^1 |m^{(\beta)}(x)|^2 dx,$$

  where $m^{(\beta)}$ denotes the β-th derivative, then the estimator is called a $(\beta+1)$-th order spline. As you may expect, we can construct a truncated power basis using polynomials up to the order of $\beta+1$. Namely, we will use $1, x, x^2, \cdots, x^{\beta+1}$ and knots to construct the basis.

11.3 Penalized Parametric Regression

We consider a multivariate linear regression problem where there are d covariates, i.e., $X_i \in \mathbb{R}^d$. In linear regression, we assume that the relationship between the response and the covariates can be written as

$$Y_i = \beta^T X_i + \epsilon_i,$$

where ϵ_i is a mean 0 noise random variable. Note that here we assume the intercept is 0.

In many cases, we may have many covariates so d is large. However, we believe that some of these covariates are useless covariates – the slope of these covariates are 0. Only a few covariates that have the actual linear relation with the response. Even we know this is true, if we naively apply the least square approach to find β, we often have all fitted coefficients being non-zero and some of them could even be quiet significant just due to randomness of the data. Note that the least square estimator finds the fitted parameter as

$$\widehat{\beta}_{\mathsf{LSE}} = \underset{\beta}{\mathsf{argmin}}\frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^T X_i)^2.$$

The idea of penalization/regularization can help in this case. There are two comment penalized parametric regression model: (i) the ridge regression model, and (ii) LASSO (least absolute shrinkage and selection operator).

Ridge regression. The ridge regression added a penalty called the L_2 penalty in the minimization criterion. Namely, the ridge regression finds the fitted parameter as

$$\widehat{\beta}_{\mathsf{Ridge}} = \underset{\beta}{\mathsf{argmin}} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^T X_i)^2 + \lambda\|\beta\|_2^2,$$

where $\|\beta\|_2^2 = \sum_{j=1}^{d}\beta_j^2$ is the square 2-norm of the vector β. The penalty $\lambda\|\beta\|_2^2$ is called the L_2 penalty because it is based on the L_2 norm of the parameter.

It turns out that the ridge regression has a closed-form solution that is similar to the least square estimator and the spline:

$$\widehat{\beta}_{\mathsf{Ridge}} = \left(\mathbb{X}^T\mathbb{X} + n\lambda\mathbb{I}_d\right)^{-1}\mathbb{X}^T\mathbb{Y},$$

where $\mathbb{X}$ is the $n \times d$ data matrix and $\mathbb{I}_d$ is the $d \times d$ identity matrix.

Let $\widehat{\beta}_{\mathsf{LS}} = \left(\mathbb{X}^T\mathbb{X}\right)^{-1}\mathbb{X}^T\mathbb{Y}$ be the ordinary least square estimator (no penalty, the classical approach). The ridge regression has a very similar coefficients as the least square estimator but just the coefficients are moved toward 0 because in the matrix inverse, there is an extra $n\lambda\mathbb{I}_d$ term. We will say that the ridge regression shrinks the estimator $\widehat{\beta}_{\mathsf{Ridge}}$ toward 0.

LASSO. LASSO (least absolute shrinkage and selection operator) is one of the most famous penalized parametric regression model. It has revolutionized the modern statistical research because of its attractive properties. LASSO finds the regression parameters/coefficients using

$$\widehat{\beta}_{\mathsf{LASSO}} = \underset{\beta}{\mathsf{argmin}}\ \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^T X_i)^2 + \lambda\|\beta\|_1,$$

where $\|\beta\|_1 = \sum_{j=1}^{d}|\beta_j|$ is the 1-norm of the vector β. The penalty $\lambda\|\beta\|_1$ is called the L_1 penalty.

If we normalized the covariates so that $\mathbb{X}^T\mathbb{X} = \mathbb{I}_d$, the LASSO estimates can be written as

$$\widehat{\beta}_{\mathsf{LASSO,j}} = \widehat{\beta}_{\mathsf{LS,j}} \times \max\left\{0, 1 - \frac{n\lambda}{|\widehat{\beta}_{\mathsf{LS,j}}|}\right\}$$

for $j = 1, \cdots, d$. Namely, the coefficients from LASSO are those coefficients from the least square method shrinking toward 0 and for those parameters whose value are below $n\lambda$, they will be shrink to 0.

When λ is large or the signal is small, many coefficients will be 0. This is called **sparsity** in statistics (only a few non-zero coefficients). Thus, we will say that the LASSO outputs a **sparse** estimate. Those $\widehat{\beta}_j$ will be 0 if it does not provide much improvement on predicting Y. So it naturally leads to an estimator with an automatic **variable selection** property. The value of λ will affect the estimates $\widehat{\beta}$. Larger λ encourages a sparser $\widehat{\beta}$ (namely, more coefficients are 0) whereas smaller λ leads to a less sparse $\widehat{\beta}$.

Although ridge regression also shrinks the coefficients toward 0, it does not yield a sparse estimator. The coefficients are just smaller but generally non-zero. On the other hand, LASSO not only shrinks the values of coefficients but also set them to be 0 if the effect is very weak. Actually, this is a property of the L_1 penalty – it tends to yield a sparse estimator – an estimator with many 0's.

Remark.

- L_0 **penalty.** There is something called the L_0 penalty. For a vector β, its L_0 norm is

$$\|\beta\| = \text{ number of non-zero elements.}$$

We can also use the L_0 penalty in regression:

$$\widehat{\beta}_{\mathsf{Best}} = \underset{\beta}{\mathsf{argmin}} \frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^T X_i)^2 + \lambda\|\beta\|_0.$$

The resulting coefficients are related to the so-called *best subset estimators*.

However, a problem of the L_0 penalty is that finding the minimum of $\frac{1}{n}\sum_{i=1}^{n}(Y_i - \beta^T X_i)^2 + \lambda\|\beta\|_0$ is difficult. It is a non-convex problem and is an NP-hard problem (you can just view these two statements as 'computationally very very very difficult'). Thus, in many situations we will replace the L_0 penalty by an L_1 penalty because solving an L_1 penalty problem is still a convex problem, so computationally it is not very challenging. The process of replacing L_0 penalty (or other non-convex problem) by L_1 penalty (or other convex problem) is called *convex relaxation*. A common trick in machine learning and optimization.

- **High-dimensional problem.** Both ridge regression and LASSO are common tools in high-dimensional data analysis. The so-called high-dimensional problem is where the number of variables d is much larger than the number of observation n. In this case, the usual least square estimator does not work because we have more variables (d) than the constraints n. A good news is: both ridge regression and LASSO work. In particular, LASSO is very powerful in this scenario because it leads to a sparse estimator (many coefficients are 0). The sparsity is a common belief in high-dimensional statistics because we anticipate only a few covariates are actually related to the response and most covariates are useless. Note that the high-dimensional problems are very common in genetics (there are many genes per individuals but often we have little patient in our study), neuroscience (the fMRI machine produces many voxels per person at a given time), and many other scientific domains.