STAT 153 & 248 - Time Series
Lecture Eleven

Fall 2025, UC Berkeley

Aditya Guntuboyina

October 2, 2025

1 High-dimensional Linear Model from last class

In the last lecture, we introduced the following model for a given time series $y_1, \ldots, y_n$:

$$y_t = \beta_0 + \beta_1(t-1) + \beta_2 \text{ReLU}(t-2) + \cdots + \beta_{n-1}\text{ReLU}(t-(n-1)) + \epsilon_t \tag{1}$$

where, as always, $\epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$. Here $\text{ReLU}(t-c) = (t-c)_+$ equals 0 if $t \leq c$ and equals $(t-c)$ if $t > c$.

The unknown parameters in this model are $\beta_0, \beta_1, \ldots, \beta_{n-1}$ as well as σ. This is a high-dimensional model because the number of parameters (which is $n+1$) is larger than the sample size n.

2 Two alternative representations of (1)

There are two alternative ways of writing the model (1). The first one is

$$y_t = \mu_t + \epsilon_t \qquad \text{with } \epsilon_t \overset{\text{i.i.d}}{\sim} N(0, \sigma^2) \tag{2}$$

and

$$\mu_t = \beta_0 + \beta_1(t-1) + \beta_2 \text{ReLU}(t-2) + \cdots + \beta_{n-1}\text{ReLU}(t-(n-1)). \tag{3}$$

The β's can be written in terms of μ_t as follows: $\beta_0 = \mu_1$, $\beta_1 = \mu_2 - \mu_1$, and

$$\beta_t = (\mu_{t+1} - \mu_t) - (\mu_t - \mu_{t-1}) \qquad \text{for } t = 2, \ldots, n-1.$$

In (2), μ_t can be interpreted as the underlying trend present in the data.

The second way of writing (1) is in regression form:

$$y = X\beta + \epsilon$$

where

$$X = \begin{pmatrix} 1 & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ 1 & 1 & 0 & \cdot & \cdot & \cdot & 0 \\ 1 & 2 & 1 & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & n-1 & n-2 & \cdot & \cdot & \cdot & 1 \end{pmatrix} \text{ and } \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdot \\ \cdot \\ \cdot \\ \beta_{n-1} \end{pmatrix}.$$

3 (Unregularized) Least Squares

As mentioned in the last lecture, the unregularized least squares estimates of $\beta_0, \ldots, \beta_{n-1}$ are obtained by minimizing the RSS:

$$\sum_{t=1}^{n} \left(y_t - \beta_0 - \beta_1(t-1) - \beta_2 \text{ReLU}(t-2) - \cdots - \beta_{n-1}\text{ReLU}(t-(n-1))\right)^2$$

over all $\beta_0, \ldots, \beta_{n-1}$. This gives

$$\hat{\beta}_0 = y_1 \quad \hat{\beta}_1 = y_2 - y_1 \quad \hat{\beta}_t = (y_{t+1} - y_t) - (y_t - y_{t-1})$$

for $t = 2, \ldots, n-1$. This will lead to the estimated trend function $\hat{\mu}_t = y_t$ for all t, and the smallest RSS value is zero.

These unregularized estimates will overfit the data, and will not produce a trend estimate that is simpler than the observed data.

4 Regularized Estimates

To produce useful estimates in cases where the MLE overfits, one employs the idea of regularization. We will discuss two ways of doing this: Ridge regularization and LASSO regularization.

The Ridge estimate of β will be denoted by $\hat{\beta}_{\text{ridge}}(\lambda)$ and is given by the minimizer of:

$$\begin{aligned}&\sum_{t=1}^{n} \left(y_t - \beta_0 - \beta_1(t-1) - \beta_2 \text{ReLU}(t-2) - \cdots - \beta_{n-1}\text{ReLU}(t-(n-1))\right)^2 \\ &+ \lambda\left(\beta_2^2 + \beta_3^2 + \cdots + \beta_{n-1}^2\right).\end{aligned} \tag{4}$$

In other words, $\hat{\beta}_{\text{ridge}}(\lambda)$ minimizes a new criterion function that is obtained by adding the penalty term $\lambda(\sum_{j=2}^{n-1} \beta_j^2)$ to the least squares criterion.

Here λ denotes a tuning parameter. Different choices of λ give rise to different ridge estimators $\hat{\beta}_{\text{ridge}}(\lambda)$. When $\lambda = 0$, the penalty term is not used in (4) so that $\hat{\beta}_{\text{ridge}}(\lambda)$ coincides with the unregularized least squares estimator. If λ is set to be very large, then the penalty term dominates the objective function (4) and then the first two components of $\hat{\beta}_{\text{ridge}}(\lambda)$ coincide with linear regression while the last $n-2$ components are simply set to zero.

The LASSO estimate of β will be denoted by $\hat{\beta}_{\text{lasso}}(\lambda)$ and is given by the minimizer of:

$$\begin{aligned}&\sum_{t=1}^{n} \left(y_t - \beta_0 - \beta_1(t-1) - \beta_2 \text{ReLU}(t-2) - \cdots - \beta_{n-1}\text{ReLU}(t-(n-1))\right)^2 \\ &+ \lambda\left(|\beta_2| + |\beta_3| + \cdots + |\beta_{n-1}|\right).\end{aligned} \tag{5}$$

In other words, $\hat{\beta}_{\text{lasso}}(\lambda)$ minimizes a new criterion function that is obtained by adding the penalty term $\lambda(\sum_{j=2}^{n-1} |\beta_j|)$ to the least squares criterion. As in the case of the ridge estimator, when $\lambda = 0$, the penalty term is not used in (5) so that $\hat{\beta}_{\text{ridge}}(\lambda)$ coincides with the unregularized least squares estimator. If λ is set to be very large, then the penalty term dominates the objective function (5) and then the first two components of $\hat{\beta}_{\text{ridge}}(\lambda)$ coincide with linear regression while the last $n-2$ components are simply set to zero.

The only difference between the ridge and lasso is in the penalty term: $\sum_j \beta_j^2$ vs $\sum_j |\beta_j|$. We will discuss computation and the differences between these estimators in the next lecture.

Note that, in usual implementations of ridge and lasso, the penalty is usually placed on all the coefficients (with the possible exception of the intercept). Here we are only placing it on $\beta_2, \ldots, \beta_{n-1}$. Because of this choice, these regularized estimates revert to linear regression when the tuning parameter is large.

5 Alternative Representations

Based on the alternative representations of Section 2, we can rewrite the optimization objectives (4) and (5) as

$$\sum_{t=1}^{n}(y_t - \mu_t)^2 + \lambda \sum_{t=2}^{n-1} \left((\mu_{t+1} - \mu_t) - (\mu_t - \mu_{t-1})\right)^2 \tag{6}$$

and

$$\sum_{t=1}^{n}(y_t - \mu_t)^2 + \lambda \sum_{t=2}^{n-1} \left|(\mu_{t+1} - \mu_t) - (\mu_t - \mu_{t-1})\right| \tag{7}$$

We denote the minimizer of (6) by $\hat{\mu}_t^{\text{ridge}}(\lambda)$ and the minimizer of (7) by $\hat{\mu}_t^{\text{lasso}}(\lambda)$. The relation between $\hat{\mu}_t^{\text{ridge}}(\lambda)$ and $\hat{\beta}_{\text{ridge}}(\lambda)$ is given by

$$\hat{\mu}_t^{\text{ridge}}(\lambda) = \hat{\beta}_0^{\text{ridge}}(\lambda) + \hat{\beta}_1^{\text{ridge}}(\lambda)(t-1) + \sum_{j=2}^{n-1} \hat{\beta}_j^{\text{ridge}}(\lambda)\text{ReLU}(t-j).$$

Similarly the relation between $\hat{\mu}_t^{\text{ridge}}(\lambda)$ and $\hat{\beta}_{\text{ridge}}(\lambda)$ is given by

$$\hat{\mu}_t^{\text{lasso}}(\lambda) = \hat{\beta}_0^{\text{lasso}}(\lambda) + \hat{\beta}_1^{\text{lasso}}(\lambda)(t-1) + \sum_{j=2}^{n-1} \hat{\beta}_j^{\text{lasso}}(\lambda)\text{ReLU}(t-j).$$

The estimator $\hat{\mu}_t^{\text{ridge}}(\lambda)$ is actually known by the name Hodrick-Prescott filter in the econometrics literature (see e.g., `https://en.wikipedia.org/wiki/Hodrick\OT1\textendashPrescott_filter`), and it is closely related to the cubic spline smoother (see e.g., `https://en.wikipedia.org/wiki/Smoothing_spline`).

The estimator $\hat{\mu}_t^{\text{lasso}}(\lambda)$ is known by the name ℓ_1 trend filter (see `https://stanford.edu/~boyd/papers/l1_trend_filter.html`).

Both the objective functions (6) and (7) ensure good fit to the data (because of the term $\sum_{t=1}^{n}(y_t - \mu_t)^2$) while also ensuring that neighboring slopes $\mu_{t+1} - \mu_t$ and $\mu_t - \mu_{t-1}$ are close to each other (this is because of the terms $\lambda \sum_{t=2}^{n-1} \left((\mu_{t+1} - \mu_t) - (\mu_t - \mu_{t-1})\right)^2$ and $\lambda \sum_{t=2}^{n-1} \left|(\mu_{t+1} - \mu_t) - (\mu_t - \mu_{t-1})\right|$). Closeness of neighboring slopes $\mu_{t+1} - \mu_t$ and $\mu_t - \mu_{t-1}$ gives a smooth appearance to $\{\mu_t\}$. These can therefore be seen as methods for trying to fit a smooth trend function μ_t to the observed time series y_t.

6 Ridge vs LASSO

The LASSO estimator $\hat{\beta}_{\text{lasso}}(\lambda)$ is usually sparse which means that most of $\hat{\beta}_2^{\text{lasso}}(\lambda), \ldots, \hat{\beta}_{n-1}^{\text{lasso}}(\lambda)$ are exactly (up to numerical precision) equal to zero. This implies that $\hat{\mu}_t^{\text{lasso}}(\lambda)$ is piecewise

linear. On the other hand, $\hat{\beta}_{\text{ridge}}(\lambda)$ will not be sparse in that all the terms $\hat{\beta}_2^{\text{ridge}}(\lambda), \ldots, \hat{\beta}_{n-1}^{\text{ridge}}(\lambda)$ will be nonzero (even though they may be small). This gives a smooth appearance to $\hat{\beta}_{\text{ridge}}(\lambda)$.

Some insight into the tendency of the LASSO regularization to yield exact zeroes in contrast to ridge regularization can be gained from the following two simple facts.

Fact 6.1 (Simple Ridge)**.** *Suppose y is a real number and $\lambda > 0$. Then the minimizer of*

$$f(\beta) = (y - \beta)^2 + \lambda\beta^2$$

is given by

$$\hat{\beta} = \frac{y}{1+\lambda}.$$

Proof. We just need to differentiate f and set the derivative to zero:

$$f'(\beta) = 2(\beta - y) + 2\lambda\beta = 0 \implies \beta = \frac{y}{1+\lambda}.$$

□

Fact 6.2 (Simple LASSO)**.** *Suppose y is a real number and $\lambda > 0$. Then the minimizer of*

$$f(\beta) = (y - \beta)^2 + \lambda|\beta|$$

is given by

$$\hat{\beta} = \begin{cases} y - \lambda/2 & \text{if } y > \lambda/2 \\ y + \lambda/2 & \text{if } y < -\lambda/2 \\ 0 & \text{if } -\lambda/2 \leq y \leq \lambda/2. \end{cases}$$

Proof. The derivative of f is given by:

$$f'(\beta) = \begin{cases} 2(\beta - y) + \lambda & \text{if } \beta > 0 \\ 2(\beta - y) - \lambda & \text{if } \beta < 0. \end{cases}$$

At $\beta = 0$, the function $|\beta|$ is not differentiable. We now need to set the derivative to zero. Setting to zero the expression for $f'(\beta)$ for $\beta > 0$, we get

$$2(\beta - y) + \lambda = 0 \implies \beta = y - \frac{\lambda}{2}.$$

Since this expression for $f'(\beta)$ is only valid when $\beta > 0$, we need to assume that $y > \lambda/2$.

Similarly setting to zero the expression for $f'(\beta)$ when $\beta < 0$, we get

$$2(\beta - y) - \lambda = 0 \implies \beta = y + \frac{\lambda}{2}$$

which is valid when $y + \lambda/2 < 0$ or $y < -\lambda/2$.

The above calculations show that $\hat{\beta}$ equals $y - \lambda/2$ when $y > \lambda/2$, and that $\hat{\beta}$ equals $y + \lambda/2$ when $y < -\lambda/2$. In the intermediate range $-\lambda/2 \leq y \leq \lambda/2$, check that $f'(\beta) < 0$ for $\beta < 0$ and $f'(\beta) > 0$ for $\beta > 0$. This means that f is decreasing on $(-\infty, 0)$ and then increasing on $(0, \infty)$ which implies that the minimum of f has to be achieved at 0. □

From these facts, it is clear that when $y \neq 0$, the ridge minimizer will never be zero, while the lasso minimizer will equal exactly zero for all y-values in the range $[-\lambda/2, \lambda/2]$. The LASSO penalty therefore has a tendency to produce exact zeros unlike the ridge penalty.

7 Cross-validation for selecting λ

The behavior of $\hat{\beta}_{\text{ridge}}(\lambda)$ and $\hat{\beta}_{\text{lasso}}(\lambda)$ depend crucially on the choice of the tuning parameter λ. One can visually tune λ in order to obtain $\hat{\mu}_t^{\text{ridge}}(\lambda), \hat{\mu}_t^{\text{lasso}}(\lambda)$ that is simple (not too wiggly) and which fits the data well (for example, one can start with $\lambda = 1$ and either increase or decrease λ by factors of 10 until a visually appealing trend estimate is obtained). Another popular approach is to use cross-validation.

The basic idea behind cross validation is the following. First split the total set of time points $T = \{1, \ldots, n\}$ into two disjoint groups T_{train} and T_{test}. Generally T_{train} will be much larger than T_{test} (e.g., T_{train} will contain about 80% of the data and T_{test} will contain about 20% of the data). For this split, fit the model to the time indices in T_{train} and obtain $\hat{\beta}_{\text{train}}^{\text{ridge}}(\lambda)$ as the minimizer of

$$\begin{aligned} &\sum_{t \in T_{\text{train}}} \left(y_t - \beta_0 - \beta_1(t-1) - \beta_2 \text{ReLU}(t-2) - \cdots - \beta_{n-1}\text{ReLU}(t-(n-1))\right)^2 \\ &+ \lambda\left(\beta_2^2 + \beta_3^2 + \cdots + \beta_{n-1}^2\right) \end{aligned} \tag{8}$$

and $\hat{\beta}_{\text{train}}^{\text{lasso}}(\lambda)$ as the minimizer of

$$\begin{aligned} &\sum_{t \in T_{\text{train}}} \left(y_t - \beta_0 - \beta_1(t-1) - \beta_2 \text{ReLU}(t-2) - \cdots - \beta_{n-1}\text{ReLU}(t-(n-1))\right)^2 \\ &+ \lambda\left(|\beta_2| + |\beta_3| + \cdots + |\beta_{n-1}|\right) \end{aligned} \tag{9}$$

Using these estimates, predict the values of y_t for $t \in T_{\text{test}}$:

$$\hat{y}_t^{\text{ridge}}(\lambda) = \hat{\beta}_{\text{train},0}^{\text{ridge}}(\lambda) + \hat{\beta}_{\text{train},1}^{\text{ridge}}(\lambda)(t-1) + \hat{\beta}_{\text{train},2}^{\text{ridge}}(\lambda)\text{ReLU}(t-2) + \cdots + \hat{\beta}_{\text{train},n-1}^{\text{ridge}}(\lambda)\text{ReLU}(t-(n-1))$$

and

$$\hat{y}_t^{\text{lasso}}(\lambda) = \hat{\beta}_{\text{train},0}^{\text{lasso}}(\lambda) + \hat{\beta}_{\text{train},1}^{\text{lasso}}(\lambda)(t-1) + \hat{\beta}_{\text{train},2}^{\text{lasso}}(\lambda)\text{ReLU}(t-2) + \cdots + \hat{\beta}_{\text{train},n-1}^{\text{lasso}}(\lambda)\text{ReLU}(t-(n-1))$$

The discrepancy between the actual values of y_t and the predicted values can be calculated as:

$$\text{Test-Error}^{\text{ridge}}(\lambda) = \sum_{t \in T_{\text{test}}} \left(y_t - \hat{y}_t^{\text{ridge}}(\lambda)\right)^2 \text{ and } \text{Test-Error}^{\text{lasso}}(\lambda) = \sum_{t \in T_{\text{test}}} \left(y_t - \hat{y}_t^{\text{lasso}}(\lambda)\right)^2$$

This test error is for a single train-test split. One can consider multiple train-test splits and add the test errors to obtain one measure of the test error for each value of λ:

$$\text{AllSplit-Test-Error}^{\text{ridge}}(\lambda) = \sum_{\text{all splits}} \text{Test-Error}^{\text{ridge}}(\lambda)$$

and

$$\text{AllSplit-Test-Error}^{\text{lasso}}(\lambda) = \sum_{\text{all splits}} \text{Test-Error}^{\text{lasso}}(\lambda)$$

This test error over all splits would be calculated for a set of candidate λ values (e.g., $\lambda = 10^a$ for $a = -8, -7, \ldots, 7, 8$) and then choose the value of λ which gives the smallest test error (this would give one choice of λ for ridge, and one choice of λ for lasso).

One common choice of selecting the splits is to take the test data in **Split** i to be the i-th 20% of the times t and training data to consist of the data for all other times t i.e., the test data in **Split** 1 corresponds to the first 20% of the data, test data in Split 2 corresponds to the second 20% of the data etc.

This method gives 5 different train-test splits, commonly known as 5-fold cross-validation.