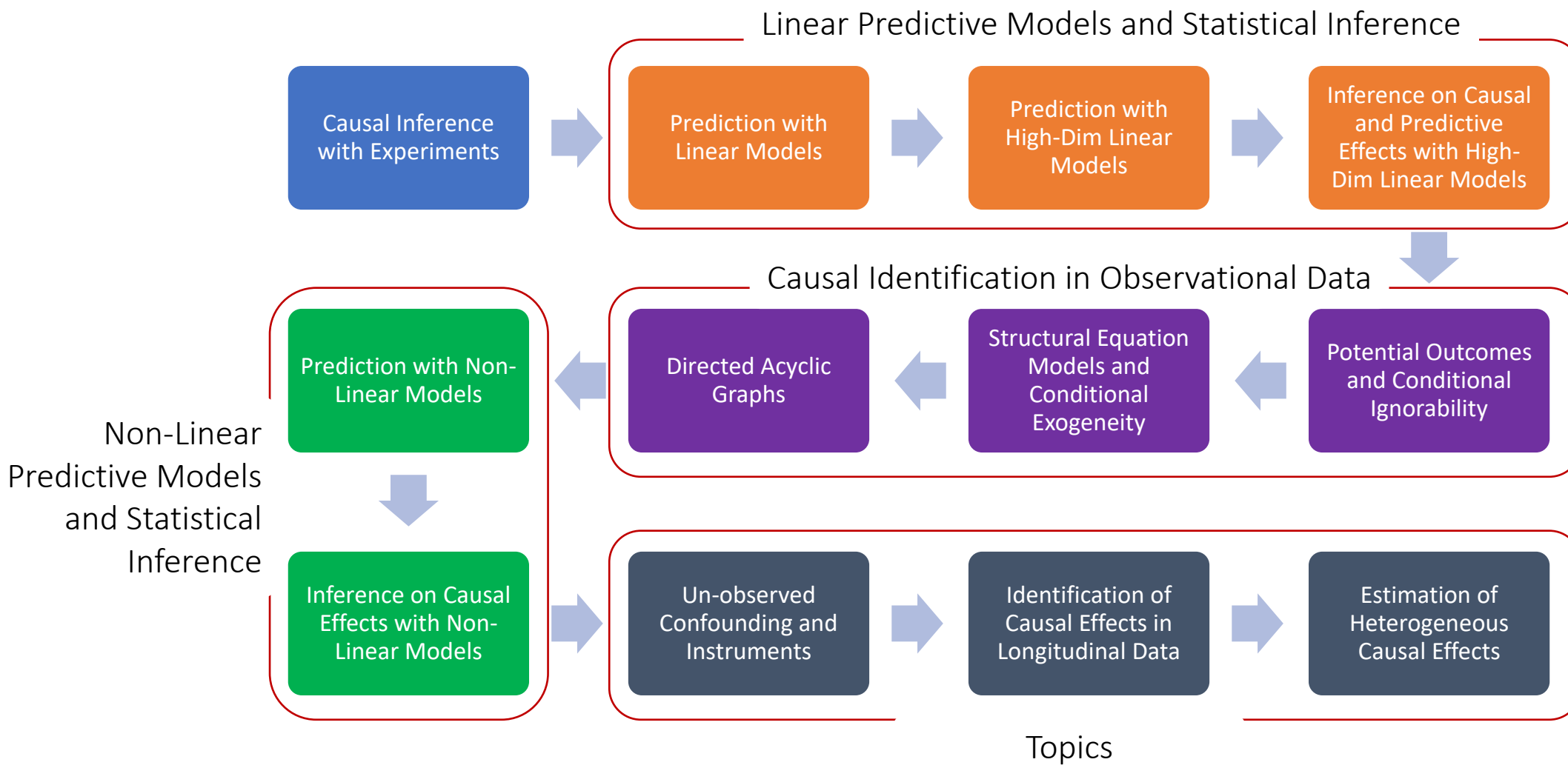
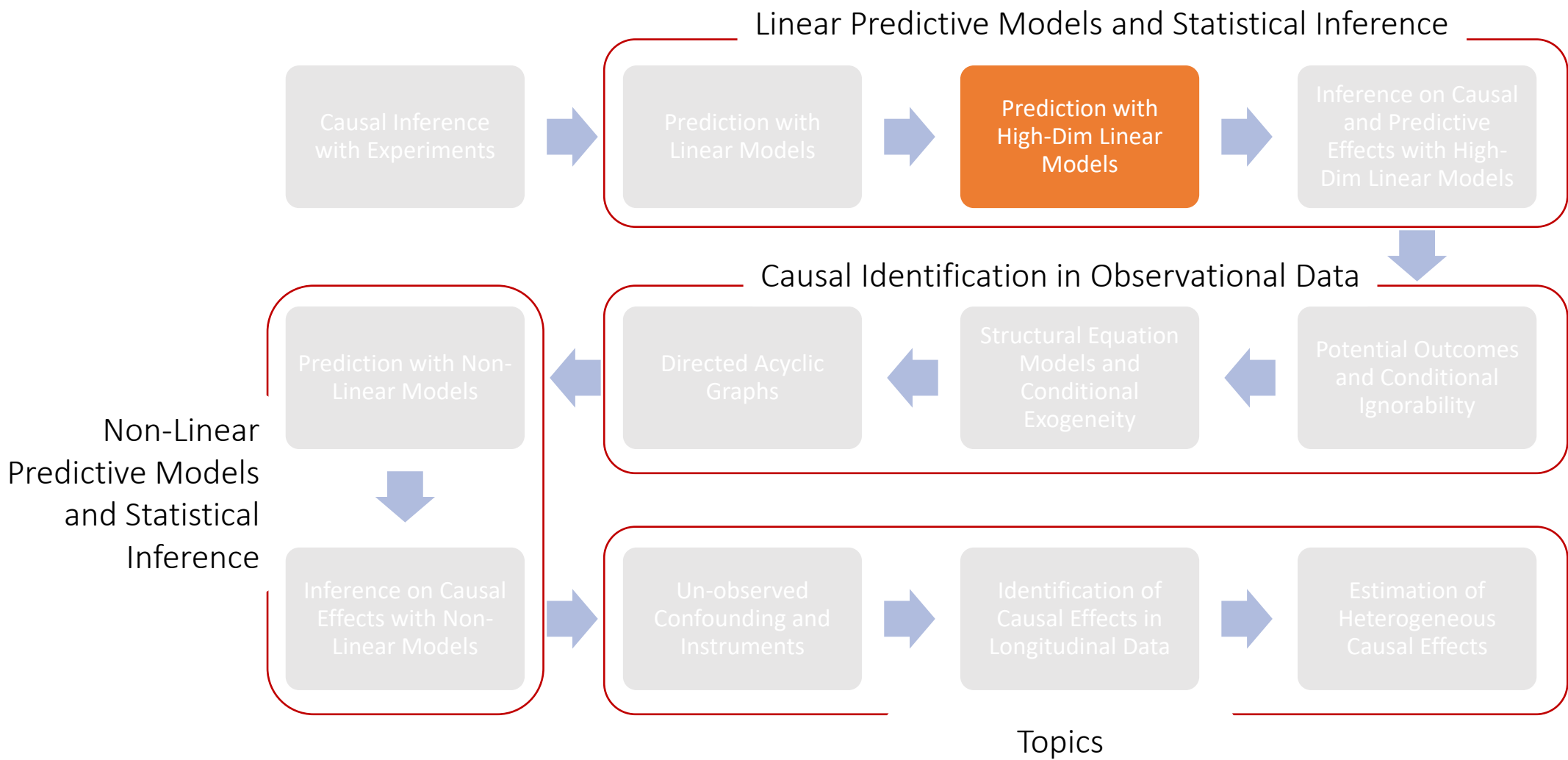


MS&E 228: Prediction with High-Dimensional Linear Models

Vasilis Syrgkanis

MS&E, Stanford





Recap of Previous Lecture

The coefficient associated with treatment D in OLS with co-variate adjustment (with or without interactions) is always consistent for the treatment effect, when run on data from a randomized experiment, as-long-as covariates are de-meanned.

The true relationship of outcome with covariates does not need to be linear.





The standard error for the coefficient associated with treatment D in OLS with co-variate adjustment can be larger than the standard error of the simple two-means estimate, if we don't also adjust for interactions of these co-variates with the treatment.



The standard error is weakly smaller for the coefficient associated with treatment D in OLS with co-variate adjustment if interaction terms of these co-variates with the treatment are also included.



Even if you only care about ATE, if you have p covariates and $p \ll n$ run OLS with interactive terms (after de-meaning covariates)!

High Dimensions: $p > n$

Best Linear Predictor

- We want to learn the BLP of Y using $X = (X_1, \dots, X_j, \dots, X_p)$

$$Y = \beta'X + \epsilon, \quad E[\epsilon X] = 0 \Leftrightarrow \epsilon \perp X$$

- From n samples $\{(X_i, Y_i)\}_{i=1}^n$
- When number of variables p is larger than the number of samples n

When do we encounter $p > n$?

Inherent High-Dimensionality

Inherent High-Dimensional Data

- Rich datasets with many covariates
- Country characteristics in cross-country wealth analysis
- Housing characteristics in housing pricing/appraisal analysis
- Individual electronic health records and claims data
- Product characteristics at point of purchase in demand analysis
- Customer characteristics in operations management
- User characteristics and history in the digital economy

Fabricated High-Dimensionality

As a means for better real-world approximations

High-Dimensionality from Feature Engineering

- A purely linear model in the given features X can be a poor approximation to reality

- Recall that if we care about RMSE, then optimal is $E[Y|X]$

$$E[Y|X] = \operatorname{argmin}_f E[(Y - f(X))^2]$$

- By variance decomposition

$$E[(Y - b'X)^2] = E[(Y - E[Y|X])^2] + E[(E[Y|X] - b'X)^2]$$

- The BLP minimizes

$$\min_{b \in \mathbb{R}^p} E[(E[Y|X] - b'X)^2]$$

- The BLP is the **Best Linear Approximation (BLA)** of the CEF

High-Dimensionality from Feature Engineering

- If instead we first construct variables $P(X) = (P_1(X), \dots, P_p(X))$ that correspond to non-linear functions of the raw variables X

- Then the Best Linear Predictor using $P(X)$ can be a much better BLA

$$\min_b E \left[(E[Y|X] - b'P(X))^2 \right] \ll \min_b E \left[(E[Y|X] - b'X)^2 \right]$$

These non-linear functions can involve

- Interactions of the raw features: $X_1 \cdot X_2, X_1 \cdot X_2 \cdot X_3$
- Polynomials of raw features: $X_1^2, X_1^3, X_1 \cdot X_2^2$
- Other non-linear transformations: $\log(X_1), \exp(X_2)$

High-Dimensionality from Feature Engineering

- These larger feature vectors are always a good idea in population; if we had infinite data
- But when we have finite data, more features introduce more noise!
- Eventually, will lead to severe overfitting to the samples and poor out-of-sample performance
- Recall that error scales with p/n
- We can easily reach a high-dimensional regime!

Coding Example

Not imposing any restrictions or biases on the parameters can lead to un-stable estimation in finite samples

Solution: add penalty terms to your estimation that induce biases towards solutions we a prior believe are more probable

Sparsity and the Lasso

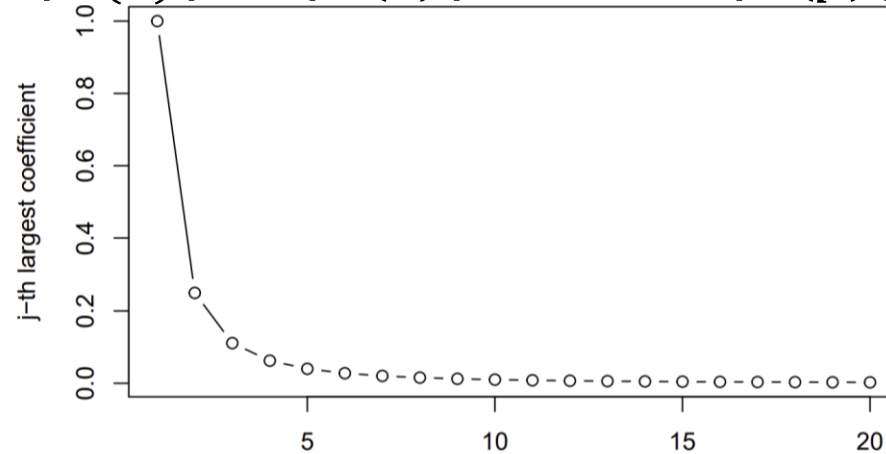
The Premise

- We might be willing to believe that most of the parameters β are roughly zero
- In that case we should be penalizing finite sample solutions $\hat{\beta}$ that have many non-zero and large coefficients
- This can stabilize the finite sample solution while keeping it close to the true β

Approximate Sparsity

- If we order our coefficients in decreasing order of magnitude

$$|\beta_{(1)}| \geq |\beta_{(2)}| \geq \cdots \geq |\beta_{(p)}|$$



- We assume that for some constant A and for some constant a

$$|\beta_{(j)}| \leq \frac{A}{j^a}$$

Effective Dimension

- The effective dimension is

$$s = c \cdot A^{\frac{1}{a}} \cdot n^{\frac{1}{2a}}$$

- The number of coefficients that have magnitude larger than $\frac{1}{\sqrt{n}}$

$$\frac{A}{j^a} \gtrsim \frac{1}{\sqrt{n}} \Rightarrow j \lesssim A^{\frac{1}{a}} \cdot n^{\frac{1}{2a}}$$

- These are roughly the number of parameters we are estimating

Example: Exact Sparsity

- If only $k < p$ of the coefficients are non-zero and bounded by C
- Then for $j \leq k$, for any a :

$$|\beta_{(j)}| \leq C \left(\frac{k}{j}\right)^a$$

- For $j > k$, for any a :

$$|\beta_{(j)}| = 0 \leq C \left(\frac{k}{j}\right)^a$$

- Coefficient is $(C k^a, a)$ -approximately sparse for $a \rightarrow \infty$:

$$s = c C^{\frac{1}{a}} \cdot k \cdot n^{\frac{1}{2a}} \approx k$$

The Method

- We want to penalize solutions that have many large coefficients
- One rough measure of the number of a vector that penalizes vectors with many large coefficients is the ℓ_1 -norm

$$\|\beta\|_1 = |\beta_1| + \cdots + |\beta_p|$$

- So instead of minimizing the empirical RMSE, we will add a penalty

$$\min_b \frac{1}{2} E_n[(Y - b'X)^2] + \lambda \|b\|_1$$

Note: Standardization

- Note that as currently stated the lasso method is not invariant to re-scaling or centering the parameters
- It is always advisable to standardize the variables before passing them to the optimization method

$$\tilde{X} = \frac{(X - E_n[X])}{\sqrt{Var_n(X)}}$$

- This way we are equally penalizing all the variables
- Otherwise, variables with larger variance are given more priority as they need smaller coefficients to be included

Note: Never Penalize the Intercept

- Most packages don't penalize the intercept of the model by default
- But be aware that this is the case
- If not, one can always demean the outcome and covariates first before passing to the Lasso package
- When evaluating, make sure that you only use the “training based calculated means” and don't calculate the “means” (of outcome and covariates) on the test set.

Choosing the Penalty

- A typical way to choose the penalty λ in practice is via cross-validation

Cross-Validation

- Partition the data into K folds (typically $K = 5$)
- Leave one block out. Fit prediction rule on the other. Predict the outcome on the left-out block and record RMSE
- Repeat for each block
- Average the RMSE across repetitions
- Repeat these steps for many values of the penalty level and choose the value that minimizes Average RMSE

The Intuition

- The j -th component $\hat{\beta}_j$ in the Lasso solution is set to zero if
$$\underbrace{\left| \partial_{b_j} E_n \left[(Y - \hat{\beta}' X)^2 \right] \right|}_{\text{Marginal benefit in prediction}} < \underbrace{\lambda}_{\text{Marginal increase in penalty}}$$

- Invoking the form of the gradient (Normal Equation)
$$\beta_j = 0 \text{ if } |\hat{S}_j| < \lambda, \quad \hat{S}_j := E_n[(Y - \hat{\beta}' X) X_j]$$

The intuition

- At the true parameter β we know that for all j
$$E[(Y - \beta'X)X_j] = 0$$

- But in finite samples

$$S_j := E_n[(Y - \beta'X)X_j]$$

- Might be non-zero. The amount of variation of this is called the noise of the problem; the noise of measuring the marginal predictive ability
- We need the penalty to be larger than this inherent noise
- Otherwise even good solutions, like the true β , will be ruled out

The intuition

- The S_j behave like a multi-variate normal distribution

$$(S_j)_{j=1}^p \sim \frac{\sigma}{\sqrt{n}} (N_j)_{j=1}^p, \quad N_j \sim N(0,1)$$

- By union bound and symmetry

$$P\left(\max_j |N_j| > \Phi^{-1}\left(1 - \frac{\alpha}{2p}\right)\right) \leq \alpha$$

- Thus we have our condition of λ dominating the noise, if we set:

$$\lambda = \frac{\sigma}{\sqrt{n}} \Phi^{-1}\left(1 - \frac{\alpha}{2p}\right) \approx \sigma \sqrt{\frac{\log(p/\alpha)}{n}}$$

The Theory

Under approximate sparsity, *restricted isometry condition (RIP)* and other regularity conditions, with the theoretically driven penalty level λ , with probability approaching $1 - \alpha$:

$$\sqrt{E_X \left[(\beta'X - \hat{\beta}'X)^2 \right]} \leq \text{const} \cdot \sqrt{E[\epsilon^2]} \sqrt{\frac{s \log(p \vee n)}{n}}$$

where s is the effective dimension

$$s = \text{const} A^{\frac{1}{a}} n^{\frac{1}{2a}}$$

The number of regressors selected is at most $\text{const} \cdot s$

The Technical Assumption

- Restricted Isometry Condition
- For any subset Z of X of dimension $L = s \log(p \vee n)$
- The restricted co-variance matrix $E[ZZ']$ is well-posed and bounded
$$C_1 \preceq E[ZZ'] \preceq C_2$$
- The empirical co-variance converges to the population covariance in ℓ_∞ norm

$$\sup_{a: \|a\|_1=1} |a'(E_n[ZZ'] - E[ZZ'])a|$$

Post-Lasso OLS

- It could be beneficial in finite samples to attempt to remove the regularization bias, at least from the chosen coefficients
- We can do that by running an OLS step after the Lasso step on the subset of variables with a non-zero coefficient
- The method has the same worst-case guarantees as Lasso, so it cannot hurt

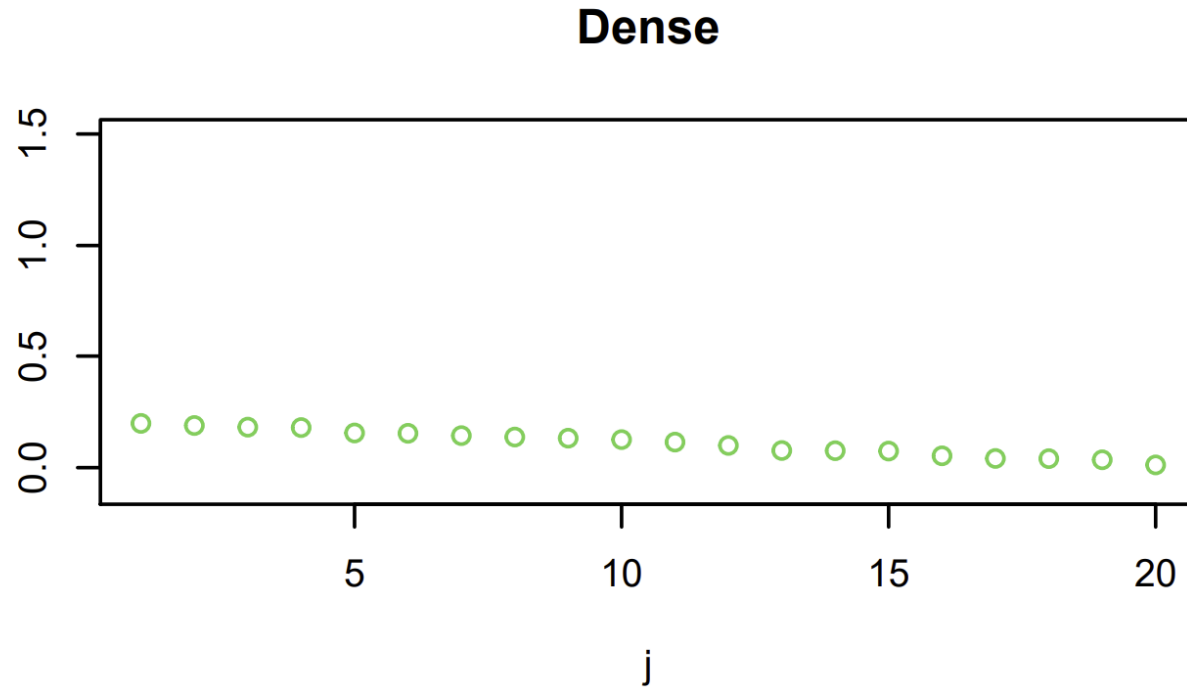
Watch out with regularizing Post Lasso OLS

Post (Lasso-CV) OLS \neq (Post Lasso OLS)-CV

- Either choose theoretically driven penalty
- Or run cross-validation for the overall Post Lasso OLS procedure
- LassoCV tends to choose too many non-zero coefficients than optimal
- Adding an OLS step to such a large model can lead to overfitting
- CV'ing the overall process enforces a large penalty more appropriate for the Post Lasso OLS procedure

Beyond Sparsity

Small Dense Coefficients and Ridge

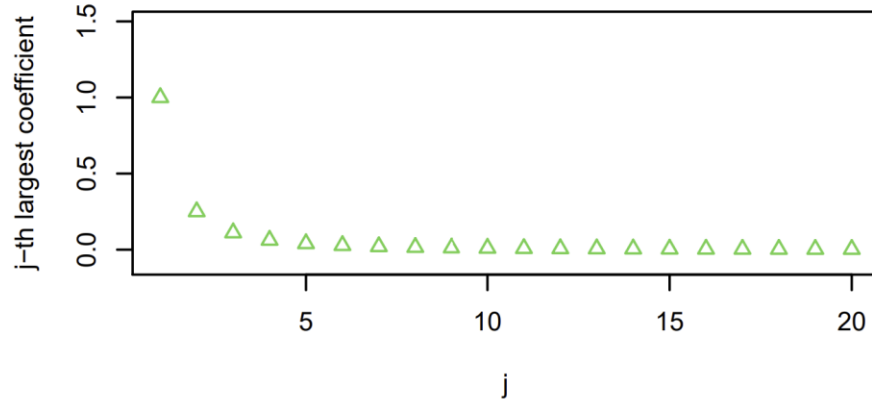


$$\min_b \frac{1}{2} E_n[(Y - b'X)^2] + \lambda \|\beta\|_2^2$$

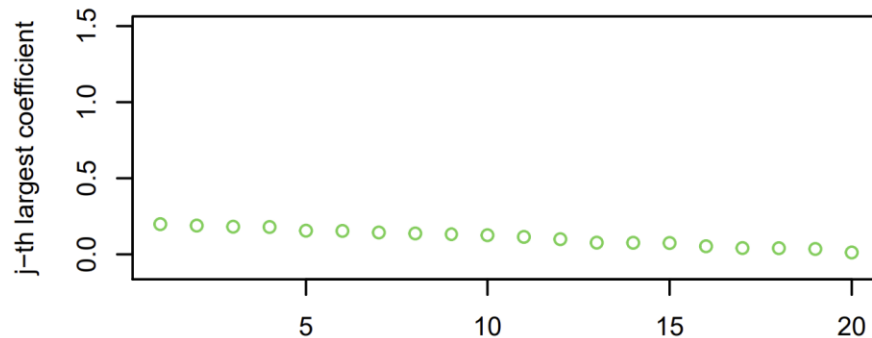
$$\|\beta\|_2^2 = \beta_1^2 + \dots + \beta_p^2$$

Dense or Sparse and ElasticNet

Approximately Sparse

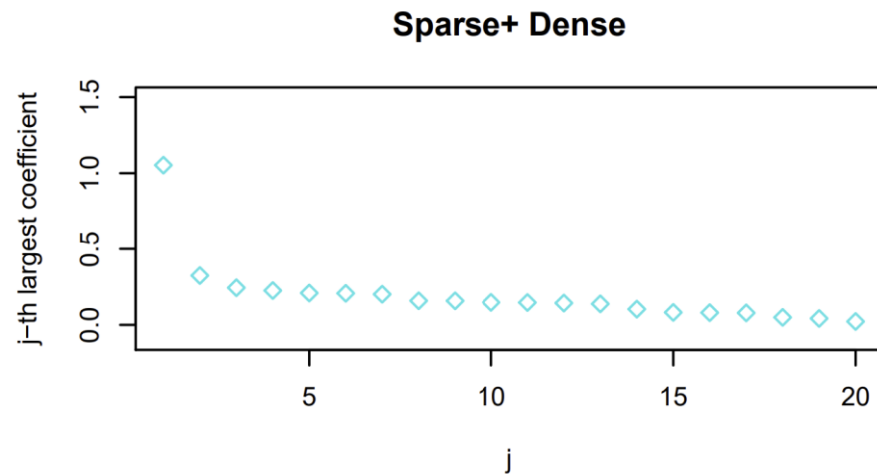


Dense



$$\min_b \frac{1}{2} E_n[(Y - b'X)^2] + \lambda \left((1 - \alpha) \|b\|_2^2 + \alpha \|b\|_1 \right)$$

Dense + Sparse and LAVA



$$\min_{b=\gamma+\delta} \frac{1}{2} E_n[(Y - b'X)^2] + \lambda_1 \|\gamma\|_2^2 + \lambda_2 \|\delta\|_1$$