Supervised Learning II: high-dimensional model selection

Yongjin Park
University of British Columbia

07 March, 2022

Learning Objectives

- Model selection, Bias-Variance tradeoff, a Bayesian View
- How do we handle $p \gg n$ situation in practice?
- Multiple Frequenist & Bayesian approaches

A working example: predicting gene expressions from genetic information

Q. Can we predict gene expressions based on genetic information?

$$\text{DNA} \xrightarrow{\text{here?}} \text{mRNA} \rightarrow \text{protein}$$

If we could predict gene expression...

$$\text{DNA} \overset{\text{here?}}{\longrightarrow} \text{mRNA} \longrightarrow \text{protein}$$

We can guess potential mechanisms of genetic disorders:

$$\text{DNA change} \longrightarrow \overset{\text{black-box}}{(\cdots)} \longrightarrow \text{disease}$$

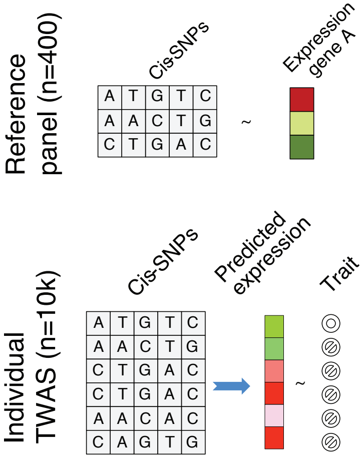
because we can do transcriptome-wide association studies (TWAS):

$$\Delta\text{DNA} \longrightarrow \text{mRNA}(\Delta\text{DNA}) \overset{\text{test this}}{\longrightarrow} \text{disease}$$

Gamazon *et al.* Nature Genetics (2015)

Gusev *et al.* Nature Genetics (2016)

If we could predict gene expression by genetic information...



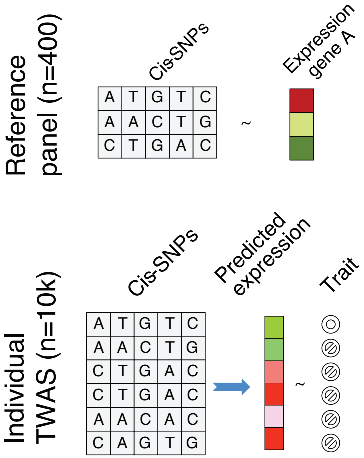
A	T	G	T	C
A	A	C	T	G
C	T	G	A	C

A	T	G	T	C
A	A	C	T	G
C	T	G	A	C
C	T	G	A	C
A	A	C	A	C
C	A	G	T	G

Gusev *et al.* Nature Genetics (2016)

Gamazon *et al.* Nature Genetics (2015)

If we could predict gene expression by genetic information...



Reference panel (n=400)

Cis-SNPs

A	T	G	T	C
A	A	C	T	G
C	T	G	A	C

~

Expression gene A

Individual TWAS (n=10k)

Cis-SNPs

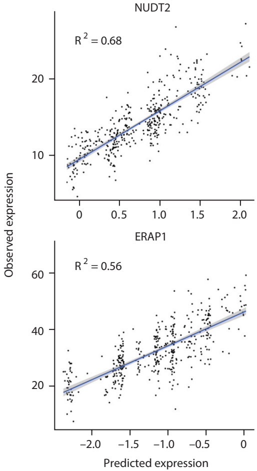
A	T	G	T	C
A	A	C	T	G
C	T	G	A	C
C	T	G	A	C
A	A	C	A	C
C	A	G	T	G

Predicted expression

~

Trait

Gusev *et al.* Nature Genetics (2016)

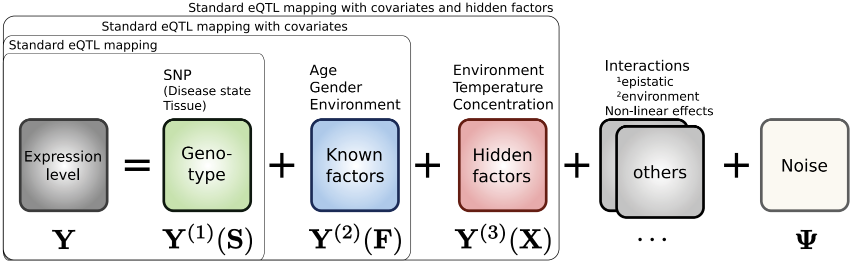


Gamazon *et al.* Nature Genetics (2015)

Today's problem: gene expression prediction

- We will focus on supervised learning (regression) of gene expression
- We will revisit the problem to discuss biological aspects in the GWAS lectures

Why regression?



- Handle multiple types of biological and technical factors
- Including all the variables often improve statistical powers
- What if there are too many variables?

Stegle *et al.* PLoS Genetics (2010)

Modeling gene expression as a function of genetic variants

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ X_{21} & \cdots & X_{2p} \\ & \cdots & \\ X_{n1} & \cdots & X_{np} \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}$$

Multivariate linear regression model:

$$\mathbf{y} = X\theta + \epsilon, \, \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)\,.$$

Example

- $\mathbf{y}$: a gene expression measured by RNA-seq / microarray.
- (X_{ij}) : genetic variants at locus j measured on individual i. X can be anything of interest, such as other genes and phenotypes.
- We can fit the model gene by gene (independence) or all the genes jointly (dependency between genes)

Two major interests in regression analysis

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ X_{21} & \cdots & X_{2p} \\ & \cdots & \\ X_{n1} & \cdots & X_{np} \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}$$

Multivariate linear regression model:

$$\mathbf{y} = X\theta + \epsilon, \, \epsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 I\right).$$

Two major interests in regression analysis

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ X_{21} & \cdots & X_{2p} \\ & \cdots & \\ X_{n1} & \cdots & X_{np} \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}$$

Multivariate linear regression model:

$$\mathbf{y} = X\theta + \epsilon, \, \epsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 I\right).$$

1. Estimation of unknown parameters (**posterior probability**):

$$p(\theta|X, \mathbf{y}) \propto p(\mathbf{y}|X, \theta)p(\theta)$$

Two major interests in regression analysis

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ X_{21} & \cdots & X_{2p} \\ & \cdots & \\ X_{n1} & \cdots & X_{np} \end{pmatrix}, \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}$$

Multivariate linear regression model:

$$\mathbf{y} = X\theta + \epsilon, \, \epsilon \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 I\right).$$

1. Estimation of unknown parameters (**posterior probability**):

$$p(\theta|X, \mathbf{y}) \propto p(\mathbf{y}|X, \theta)p(\theta)$$

2. Prediction of future phenotype (**posterior prediction**):

$$p(\mathbf{y}^{\text{new}}|X^{\text{new}}, X, \mathbf{y}) = \int p(\mathbf{y}^{\text{new}}|X^{\text{new}}, \theta)p(\mathbf{y}|X, \theta)p(\theta)d\theta$$

Of many important questions, we will try to tackle this one...

$$p \gg n$$

- n: sample size
- p: number of parameters

Today's lecture

Bias-variance tradeoff

High-dimensional multivariate regression

Knock-off filter to control False Discovery Rate

Discussion

A class of models $\hat{f} \in \mathcal{F}$

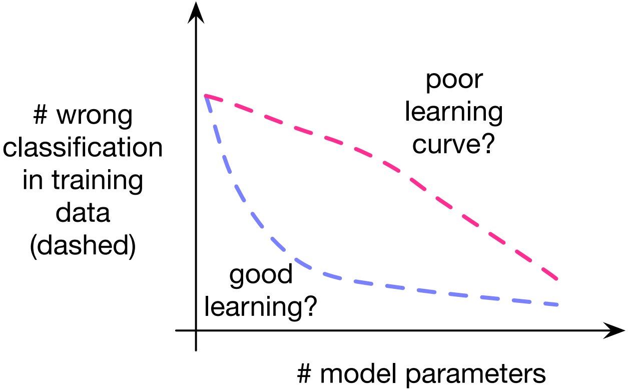
$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \theta_1 \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{pmatrix} + \cdots \theta_p \begin{pmatrix} X_{1p} \\ X_{2p} \\ \vdots \\ X_{np} \end{pmatrix}.$$

- Different number of variables will define a class of potential models
- E.g., $\mathcal{F}_1$: a class of models with one variable
- $\mathcal{F}_2$: a class of models with two variables
- (...)
- $\mathcal{F}_q$: a class of models with q variables

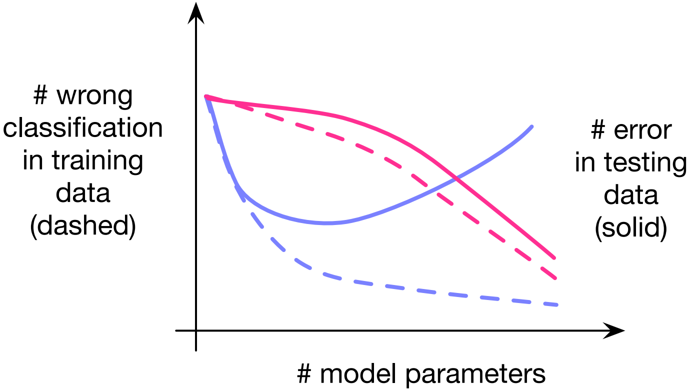
How do we know if one model is better than the other?

- Training vs. (unseen) testing data
- Our hope: training $\approx$ testing

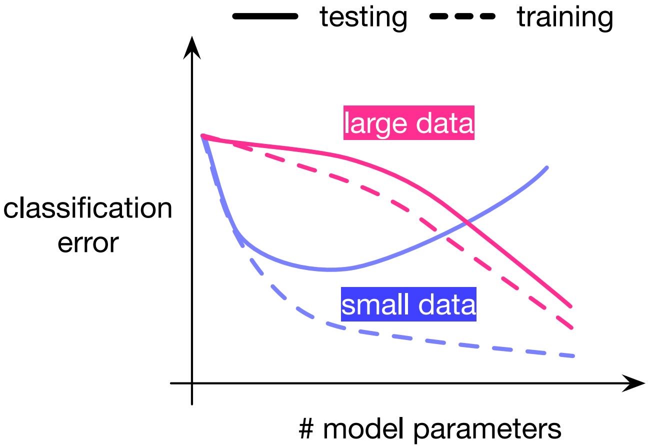
What is a good classifier?



What is a good classifier?



What is a good classifier?



The ultimate goal: generalization error minimization

k-fold CV error → leave-one-out CV error → generalization error

- No matter what may come, we will still predict as good as this...
- We will use k-fold cross validation error to estimate generalization error

Bias-variance tradeoff in generalization error

$$\mathbb{E}\left[\overset{\text{true unknown model}}{f(X)} - \overset{\text{our attempt}}{\hat{f}(X)}\right]^2 = \mathbb{E}\left[f(X) \overbrace{-\mathbb{E}\left[\hat{f}\right] + \mathbb{E}\left[\hat{f}\right]}^{\text{average model within the class}} - \hat{f}(X)\right]^2$$

Bias-variance tradeoff in generalization error

$$\begin{aligned}
\mathbb{E}\left[\overset{\text{true unknown model}}{f(X)} - \overset{\text{our attempt}}{\hat{f}(X)}\right]^2 &= \mathbb{E}\left[f(X) \overbrace{- \mathbb{E}\left[\hat{f}\right] + \mathbb{E}\left[\hat{f}\right]}^{\text{average model within the class}} - \hat{f}(X)\right]^2 \\
\text{(expand the square)} &= \mathbb{E}\Big[(f(X) - \mathbb{E}\left[\hat{f}\right])^2 + (\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))^2 \\
&\quad + 2(f(X) - \mathbb{E}\left[\hat{f}\right])(\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))\Big]
\end{aligned}$$

Bias-variance tradeoff in generalization error

$$
\begin{aligned}
\mathbb{E}\bigg[\overset{\text{true unknown model}}{f(X)} - \overset{\text{our attempt}}{\hat f(X)}\bigg]^2 &= \mathbb{E}\bigg[f(X) \overbrace{-\,\mathbb{E}\big[\hat f\big] + \mathbb{E}\big[\hat f\big]}^{\text{average model within the class}} -\hat f(X)\bigg]^2 \\
\text{(expand the square)} &= \mathbb{E}\bigg[(f(X) - \mathbb{E}\big[\hat f\big])^2 + (\mathbb{E}\big[\hat f\big] - \hat f(X))^2 \\
&\quad +2(f(X) - \mathbb{E}\big[\hat f\big])(\mathbb{E}\big[\hat f\big] - \hat f(X))\bigg] \\
\text{(rearrange the terms)} &= \mathbb{E}\big[f(X) - \mathbb{E}\big[\hat f\big]\big]^2 + \mathbb{E}\big[\mathbb{E}\big[\hat f\big] - \hat f(X))\big]^2 \\
&\quad +2\mathbb{E}\big[f(X) - \mathbb{E}\big[\hat f\big]\big]\,\cancelto{0}{\mathbb{E}\big[\mathbb{E}\big[\hat f\big] - \hat f(X)\big]}
\end{aligned}
$$

Bias-variance tradeoff in generalization error

$$
\begin{aligned}
\mathbb{E}\left[\overset{\text{true unknown model}}{f(X)} - \overset{\text{our attempt}}{\hat{f}(X)}\right]^2 &= \mathbb{E}\left[f(X) \overbrace{-\mathbb{E}\left[\hat{f}\right] + \mathbb{E}\left[\hat{f}\right]}^{\text{average model within the class}} - \hat{f}(X)\right]^2 \\
\text{(expand the square)} &= \mathbb{E}\left[(f(X) - \mathbb{E}\left[\hat{f}\right])^2 + (\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))^2\right. \\
&\quad \left. + 2(f(X) - \mathbb{E}\left[\hat{f}\right])(\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))\right] \\
\text{(rearrange the terms)} &= \mathbb{E}\left[f(X) - \mathbb{E}\left[\hat{f}\right]\right]^2 + \mathbb{E}\left[\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))\right]^2 \\
&\quad + 2\mathbb{E}\left[f(X) - \mathbb{E}\left[\hat{f}\right]\right] \cancelto{0}{\mathbb{E}\left[\mathbb{E}\left[\hat{f}\right] - \hat{f}(X)\right]} \\
&= \underbrace{\mathbb{E}\left[f(X) - \mathbb{E}\left[\hat{f}\right]\right]^2}_{\text{bias}^2} + \underbrace{\mathbb{E}\left[\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))\right]^2}_{\text{variance}}
\end{aligned}
$$

Bias-variance tradeoff in generalization error

$$\mathbb{E}\left[\overset{\text{true unknown model}}{f(X)} - \overset{\text{our attempt}}{\hat{f}(X)}\right]^2 = \mathbb{E}\left[f(X) \overbrace{- \mathbb{E}\left[\hat{f}\right] + \mathbb{E}\left[\hat{f}\right]}^{\text{average model within the class}} - \hat{f}(X)\right]^2$$

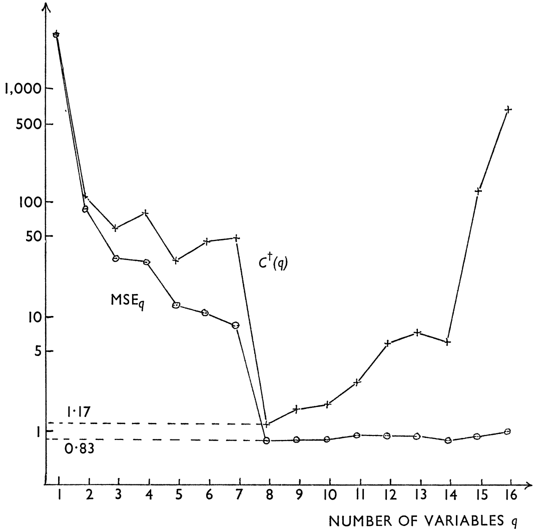
$$\begin{aligned}
\text{(expand the square)} \quad &= \mathbb{E}\Big[(f(X) - \mathbb{E}\left[\hat{f}\right])^2 + (\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))^2 \\
&\quad +2(f(X) - \mathbb{E}\left[\hat{f}\right])(\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))\Big] \\
\text{(rearrange the terms)} \quad &= \mathbb{E}\left[f(X) - \mathbb{E}\left[\hat{f}\right]\right]^2 + \mathbb{E}\left[\mathbb{E}\left[\hat{f}\right] - \hat{f}(X))\right]^2 \\
&\quad +2\mathbb{E}\left[f(X) - \mathbb{E}\left[\hat{f}\right]\right] \cancelto{0}{\mathbb{E}\left[\mathbb{E}\left[\hat{f}\right] - \hat{f}(X)\right]}
\end{aligned}$$

Remark: We didn't factor out irreducible errors.

Bias-variance tradeoff in generalization error

$$\mathbb{E}\left[\overset{\text{true unknown model}}{f(X)} \quad - \quad \overset{\text{our attempt}}{\widehat{f}(X)}\right]^2 = \underbrace{\mathbb{E}\left[f(X) - \mathbb{E}\left[\widehat{f}\right]\right]^2}_{\text{bias}^2} + \underbrace{\mathbb{E}\left[\mathbb{E}\left[\widehat{f}\right] - \widehat{f}(X))\right]^2}_{\text{variance}}$$

k-fold cross validation in regression modelling (M. Stone 1974)



Today's lecture

Bias-variance tradeoff

High-dimensional multivariate regression

Knock-off filter to control False Discovery Rate

Discussion

In multivariate regression modelling

Model selection $\approx$ variable selection

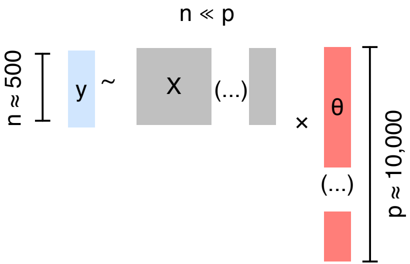
Challenges in our $p \gg n$ regression problem

Degeneracy

High degree of freedom, many, many unknown, but very title information

Col-linearity

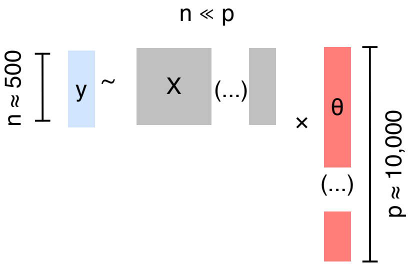
Variables are somewhat similar to each other



Challenges in our $p \gg n$ regression problem

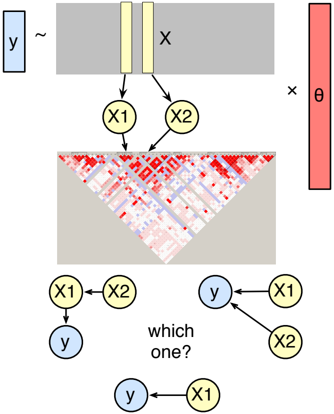
Degeneracy

High degree of freedom, many, many unknown, but very title information

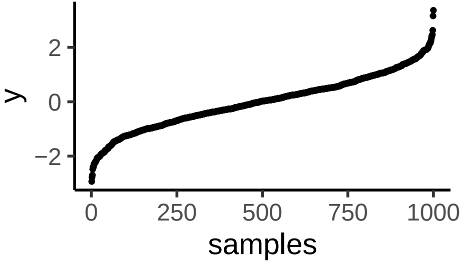


Col-linearity

Variables are somewhat similar to each other

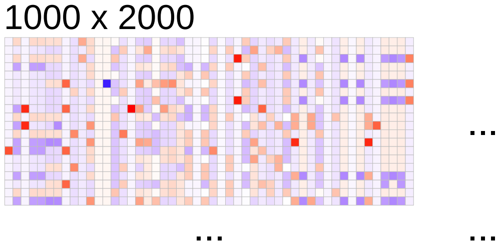


A working example - data



```
dim(y)
```

```
[1] 1000 1
```



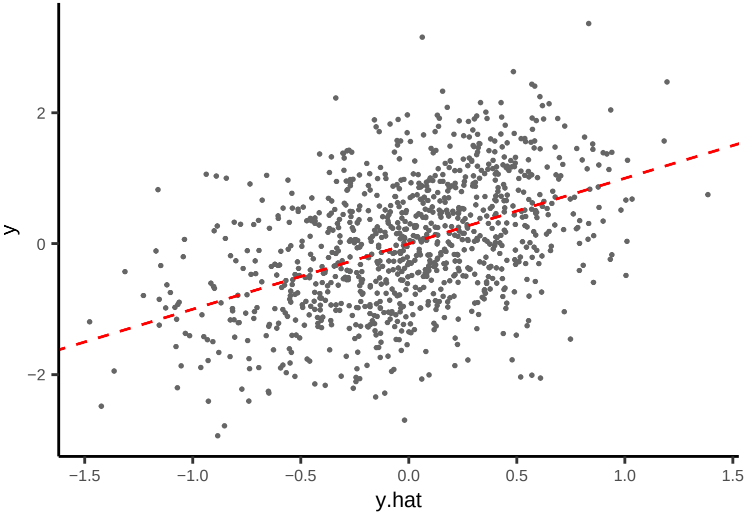
```
dim(X)
```

```
[1] 1000 2000
```

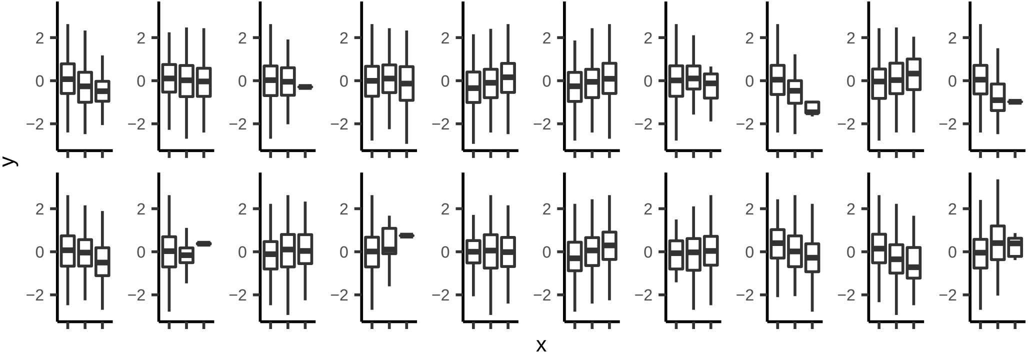
There are 20 true non-zero variables.

True causal variables explain a large fraction of variation

```r
.lm <- lm(y ~ X[, sim$causal, drop = FALSE] - 1)
```



Variant-by-variant correlations



How do we know “causal” variables from 2,000 variables?

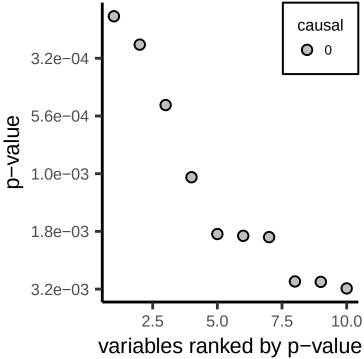
- Let’s try out one by one and rank them by univariate

```
cor.test(x,y)
```

How do we know “causal” variables from 2,000 variables?

- Let’s try out one by one and rank them by univariate

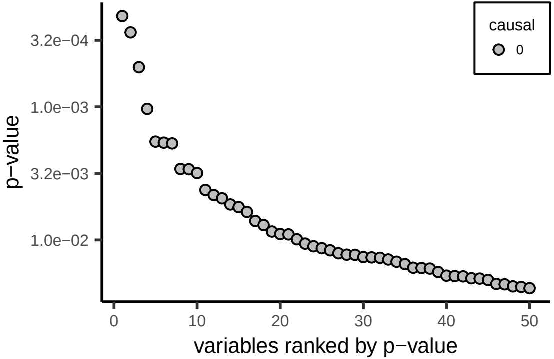
```
cor.test(x,y)
```



How do we know “causal” variables from 2,000 variables?

- Let’s try out one by one and rank them by univariate

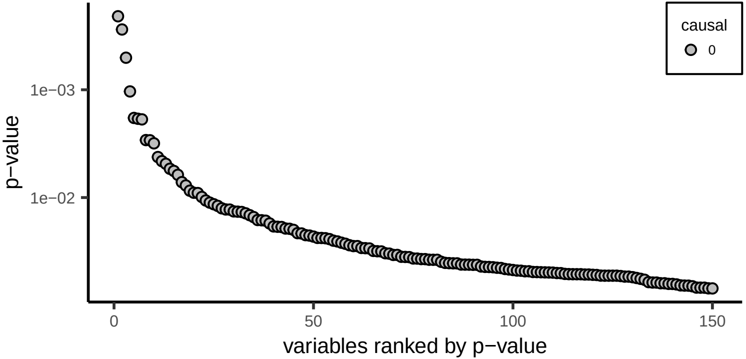
```
cor.test(x,y)
```



How do we know “causal” variables from 2,000 variables?

- Let’s try out one by one and rank them by univariate

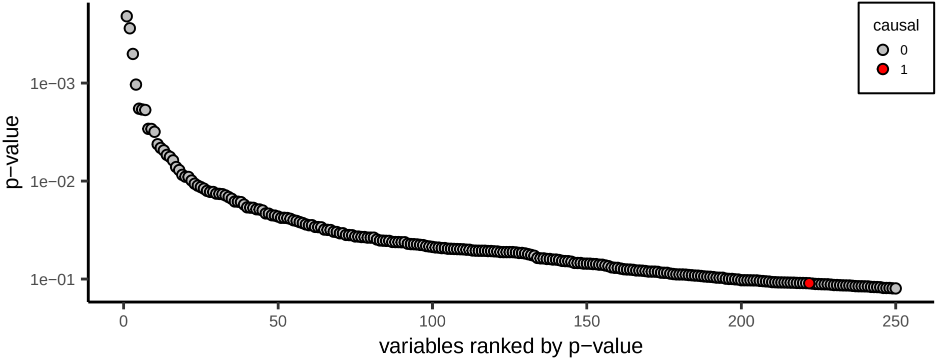
```
cor.test(x,y)
```



How do we know “causal” variables from 2,000 variables?

- Let’s try out one by one and rank them by univariate

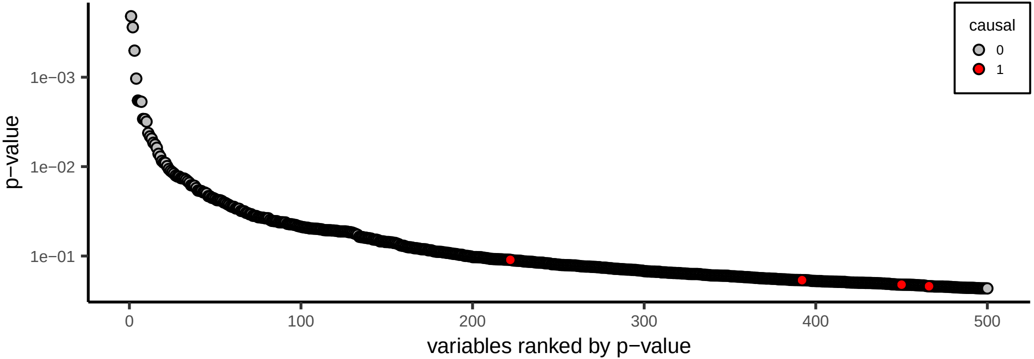
```
cor.test(x,y)
```



How do we know “causal” variables from 2,000 variables?

- Let’s try out one by one and rank them by univariate

```
cor.test(x,y)
```



- Classical variable selection by univariate (one-by-one) tests will not work for a $p \gg n$ regression problem
- Especially if we have col-linearity in the design matrix X

Can we get helped by multivariate regression?

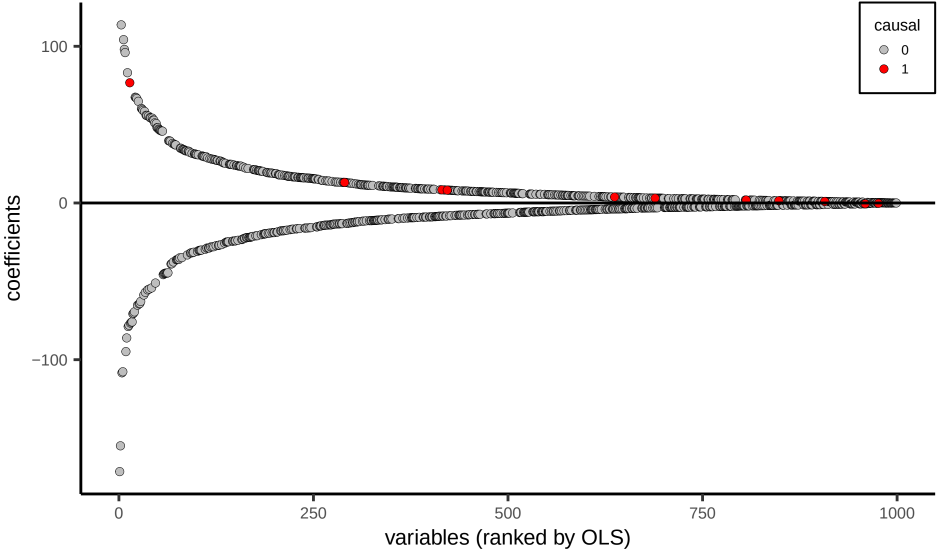
```r
lm.out <- lm(y ~ X - 1)
```

If you look at the coefficients:

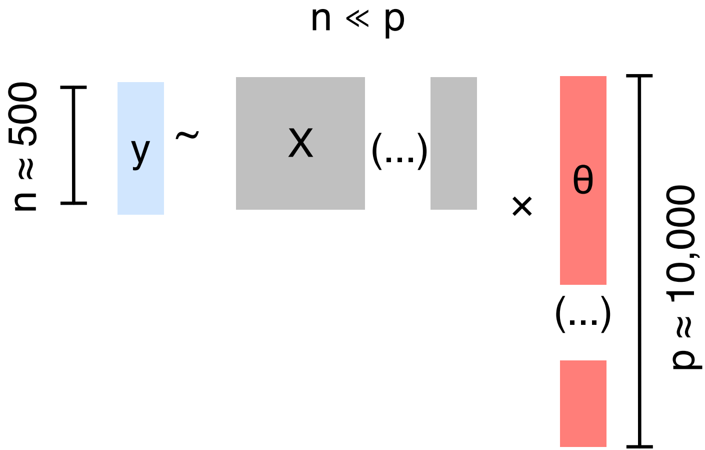
	Estimate	Std. Error	t value	Pr(>|t|)
X1	1.582654	0	117826708638	0
X2	-15.949781	0	-134571520044	0
X3	1.401209	0	117831497502	0
X4	-5.515155	0	-61749749454	0
X5	6.827798	0	66652250040	0
X6	-15.180671	0	-107848045369	0

Anything strange? Hint: $\hat{\theta} = (X^\top X)^{-1} X^\top \mathbf{y}$.

OLS overfits to the data



Can we get helped by multivariate regression?



OLS (a.k.a. MLE/MSE) is degenerate if $p \gg n$

Variable selection in high-dimensional genotype matrix ($n \ll p$)

Regression analysis = projecting the observed $\mathbf{y}$ vector on to column space of $\{\mathbf{x}_j : j \in [p]\}$,

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \theta_1 \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{pmatrix} + \cdots \theta_p \begin{pmatrix} X_{1p} \\ X_{2p} \\ \vdots \\ X_{np} \end{pmatrix}.$$

Variable selection = column selection.

- Intuitive idea : choose the best combination of variables. $\rightarrow 2^p$ choices (even harder).

Variable selection in high-dimensional genotype matrix ($n \ll p$)

Regression analysis = projecting the observed $\mathbf{y}$ vector on to column space of $\{\mathbf{x}_j : j \in [p]\}$,

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \theta_1 \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{pmatrix} + \cdots \theta_p \begin{pmatrix} X_{1p} \\ X_{2p} \\ \vdots \\ X_{np} \end{pmatrix}.$$

Variable selection = column selection.

- Intuitive idea : choose the best combination of variables. $\rightarrow 2^p$ choices (even harder).
- Alternative idea : make as many θ_j's nearly zero values.

Variable selection in high-dimensional genotype matrix ($n \ll p$)

Regression analysis = projecting the observed $\mathbf{y}$ vector on to column space of $\{\mathbf{x}_j : j \in [p]\}$,

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \theta_1 \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{pmatrix} + \cdots \theta_p \begin{pmatrix} X_{1p} \\ X_{2p} \\ \vdots \\ X_{np} \end{pmatrix}.$$

Variable selection = column selection.

- Intuitive idea : choose the best combination of variables. $\rightarrow 2^p$ choices (even harder).
- Alternative idea : make as many θ_j's nearly zero values.
- What prior does: penalize $|\theta_j| > 0$ so that only the strong enough variables take non-zero values.

Reconciling two related concepts – MLE and MSE

Equivalence of maximum-likelihood estimation and mean square error minimization (isotropic Gaussian error distribution).

[MLE] Find θ maximizing

$$\ln p(\mathbf{y}|X,\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2 + \text{const.}$$

without prior contribution of parameter, and σ is known.

[MSE] Find θ minimizing

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2.$$

MLE, MSE, an optimization problem

Minimization of the convex loss function:

$$L(\theta) = (\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta)$$

MLE, MSE, an optimization problem

Minimization of the convex loss function:

$$L(\theta) = (\mathbf{y} - X\theta)^\top (\mathbf{y} - X\theta)$$

We can optimize setting the derivative with respect to θ to zero:

$$\nabla_\theta L = X^\top (\mathbf{y} - X\theta) = 0$$

Rearranging the equation

$$\mathbf{y}^\top X = X^\top X\theta \implies \hat{\theta}_{MLE} = (X^\top X)^{-1} X^\top y.$$

MLE, MSE, an optimization problem

Minimization of the convex loss function:

$$L(\theta) = (\mathbf{y} - X\theta)^\top (\mathbf{y} - X\theta)$$

We can optimize setting the derivative with respect to θ to zero:

$$\nabla_\theta L = X^\top (\mathbf{y} - X\theta) = 0$$

Rearranging the equation

$$\mathbf{y}^\top X = X^\top X\theta \implies \hat{\theta}_{MLE} = (X^\top X)^{-1} X^\top y.$$

- Approximately, $p(\theta|\mathbf{y}, X) \approx \mathcal{N}\left(\theta \middle| \hat{\theta}_{MLE}, \sigma^2 (X^\top X)^{-1}\right)$.

MLE, MSE, an optimization problem

Minimization of the convex loss function:

$$L(\theta) = (\mathbf{y} - X\theta)^\top (\mathbf{y} - X\theta)$$

We can optimize setting the derivative with respect to θ to zero:

$$\nabla_\theta L = X^\top (\mathbf{y} - X\theta) = 0$$

Rearranging the equation

$$\mathbf{y}^\top X = X^\top X\theta \implies \hat{\theta}_{MLE} = (X^\top X)^{-1} X^\top y.$$

- Approximately, $p(\theta|\mathbf{y}, X) \approx \mathcal{N}\left(\theta \middle| \hat{\theta}_{MLE}, \sigma^2 (X^\top X)^{-1}\right)$.
- How hard is $(X^\top X)^{-1}$ (i.e., inverse of $p \times p$ matrix)?

MLE, MSE, an optimization problem

Minimization of the convex loss function:

$$L(\theta) = (\mathbf{y} - X\theta)^\top (\mathbf{y} - X\theta)$$

We can optimize setting the derivative with respect to θ to zero:

$$\nabla_\theta L = X^\top (\mathbf{y} - X\theta) = 0$$

Rearranging the equation

$$\mathbf{y}^\top X = X^\top X\theta \implies \hat{\theta}_{MLE} = (X^\top X)^{-1} X^\top y.$$

- Approximately, $p(\theta|\mathbf{y}, X) \approx \mathcal{N}\left(\theta \middle| \hat{\theta}_{MLE}, \sigma^2 (X^\top X)^{-1}\right)$.
- How hard is $(X^\top X)^{-1}$ (i.e., inverse of $p \times p$ matrix)?
- What if $n \ll p$? What if we want to include $p(\theta)$?

Bayesian/regularization idea to add the missing probability component

We've been discussing the conditional likelihood

$$p(\mathbf{y}|X, \theta)$$

without a prior probability of regression coefficients,

$$p(\theta)$$

What will be a suitable prior distribution of θ?

Recall: Reconciling two related concepts – MLE and MSE

Equivalence of maximum-likelihood estimation and mean square error minimization (isotropic Gaussian error distribution).

[MLE] Find θ maximizing

$$\ln p(\mathbf{y}|X,\theta) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2 + \text{const.}$$

without prior contribution of parameter, and σ is known.

[MSE] Find θ minimizing

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2.$$

Ridge regression, a linear regression with Gaussian prior (L2)

Prior distribution

$$p(\theta) = \mathcal{N}(\theta|\mathbf{0}, \lambda^{-1}I) \propto \exp\left(-\frac{\lambda}{2}\|\theta\|^2\right)$$

where

$$\|\theta\|^2 = \sum_{j=1}^{p} \theta_j^2, \text{ L2-norm.}$$

Maximize

$$\ln p(\mathbf{y}|X,\theta) + \ln p(\theta|\lambda) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2 - \frac{\lambda}{2}\|\theta\|^2$$

Minimize L_2-regularized error

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2 + \frac{\lambda}{2}\|\theta\|^2$$

Lasso regression, a linear regression with Laplace prior (L1)

Prior distribution

$$p(\theta) = \mathsf{Laplace}(\theta|\lambda) \propto \exp\left(-\lambda\|\theta\|_1\right)$$

where

$$\|\theta\|_1 = \sum_{j=1}^{p} |\theta_j|, \text{ L1-norm.}$$

Maximize

$$\ln p(\mathbf{y}|X,\theta) + \ln p(\theta|\lambda) = -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2 - \lambda\|\theta\|_1$$

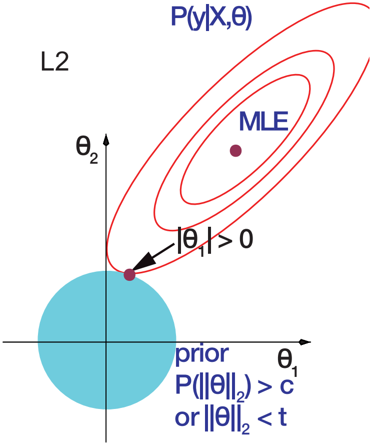
Minimize L_1-regularized error

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i\theta)^2 + \lambda\|\theta\|_1$$

(Tibshirani, 1996)

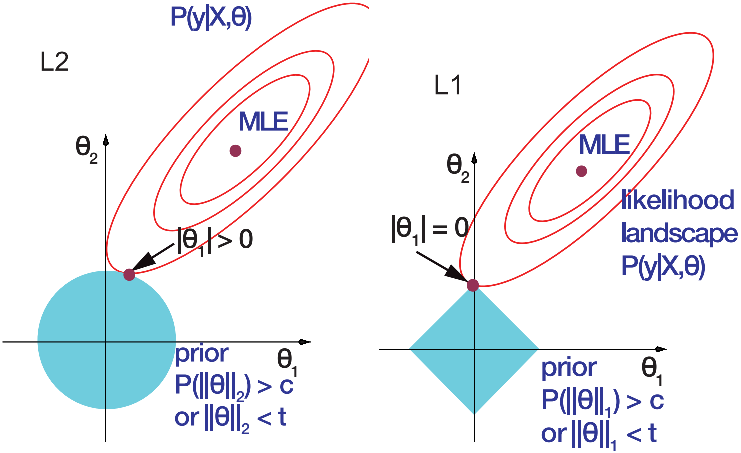
Geometric intuition of regularization.

Consider a simple regression model: $y_i = \theta_1 X_{i1} + \theta_2 X_{i2}$.



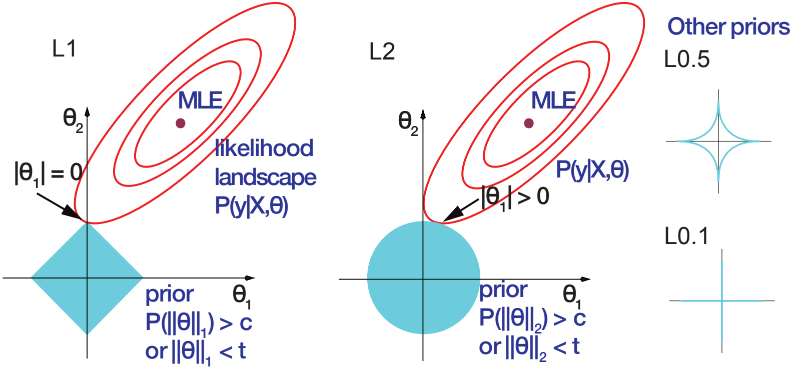
Geometric intuition of regularization.

Consider a simple regression model: $y_i = \theta_1 X_{i1} + \theta_2 X_{i2}$.



Geometric intuition of regularization.

Consider a simple regression model: $y_i = \theta_1 X_{i1} + \theta_2 X_{i2}$.



- Both regularization priors shrink the coefficients toward zero.
- But only L1 can effectively "select" variables; although we want L_0.

Hastie, Tibshirani, Friedman, *The Elements of Statistical Learning*.

Posterior inference of the regularized regression models

Our goal is to estimate (1) posterior distribution

$$p(\theta|\mathbf{y}, X) = \frac{p(\mathbf{y}|X, \theta)p(\theta)}{p(y|X)}.$$

Then (2) using $p(\theta|\mathbf{y}, X)$, predict $p(\mathbf{y}^\star|\mathbf{y}, X)$ by averaging over all possible θ sampled from the estimated posterior distribution.

- Usually posterior prediction (2) is can be easily simulated with accurate estimation of posterior distribution (1).
- Posterior inference can be done analytically or not, depending on the choice of $p(\theta)$[1].

[1] We term prior $p(\theta)$ a *conjugate prior* if its posterior $p(\theta|\mathbf{y}, X)$ is of the same type of distribution.

We can find an analytical solution in L2-regularized regression

$$\ln p(\theta|\mathbf{y}, X) = -\frac{1}{2\sigma^2}(\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta) - \frac{\lambda}{2}\theta^\top\theta + \text{const.}$$

We can find an analytical solution in L2-regularized regression

$$\ln p(\theta|\mathbf{y}, X) = -\frac{1}{2\sigma^2}(\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta) - \frac{\lambda}{2}\theta^\top\theta + \text{const.}$$

By taking derivative with respect to θ and setting it the zero vector:

$$\nabla_\theta = -\frac{1}{\sigma^2}X^\top(\mathbf{y} - X\theta) - \lambda\theta = 0$$

We can find an analytical solution in L2-regularized regression

$$\ln p(\theta|\mathbf{y}, X) = -\frac{1}{2\sigma^2}(\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta) - \frac{\lambda}{2}\theta^\top\theta + \text{const.}$$

By taking derivative with respect to θ and setting it the zero vector:

$$\nabla_\theta = -\frac{1}{\sigma^2}X^\top(\mathbf{y} - X\theta) - \lambda\theta = 0$$

Rearranging the equation:

$$X^\top\mathbf{y} = (X^\top X + \lambda\sigma^2 I)\theta \implies \hat{\theta} = (X^\top X + \lambda\sigma^2 I)^{-1}X^\top\mathbf{y}$$

We can find an analytical solution in L2-regularized regression

$$\ln p(\theta|\mathbf{y}, X) = -\frac{1}{2\sigma^2}(\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta) - \frac{\lambda}{2}\theta^\top\theta + \text{const.}$$

By taking derivative with respect to θ and setting it the zero vector:

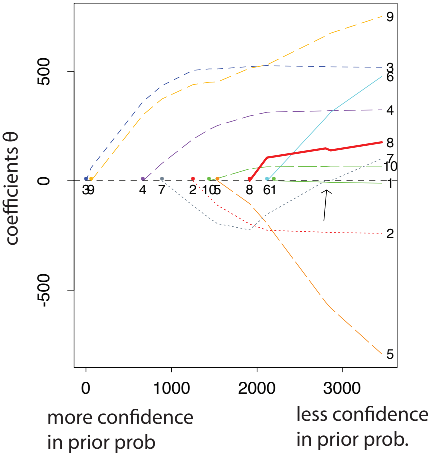
$$\nabla_\theta = -\frac{1}{\sigma^2}X^\top(\mathbf{y} - X\theta) - \lambda\theta = 0$$

Rearranging the equation:

$$X^\top\mathbf{y} = (X^\top X + \lambda\sigma^2 I)\theta \implies \hat{\theta} = (X^\top X + \lambda\sigma^2 I)^{-1}X^\top\mathbf{y}$$

Remark: For $n \ll p$, the inverse $(X^\top X)^{-1}$ may not exists, but $(X^\top X + \lambda\sigma^2 I)^{-1}$ can exist with a proper λ.

We can solve L1-regularized regression numerically



Algorithms from statistics:

- Efron *et al.* Least Angle Regression (2002)
- Hans *et al.*, Shotgun search (2007)
- Friedman *et al.*, `glmnet` (2010)

From ML:

- Figueiredo *et al.* PAMI (2003)
- Seeger *et al.* JMLR (2008)

In practice, the greedy algorithm of `glmnet` works so well

Goal:

$$\min_{\theta} \quad \overbrace{(\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta)}^{\text{RSS}} + \underbrace{\lambda\alpha\|\theta\|_1}_{\text{variable selection}} + \underbrace{\lambda(1-\alpha)\|\theta\|_2}_{\text{shrinkage}}$$

In practice, the greedy algorithm of `glmnet` works so well

Goal:

$$\min_{\theta} \quad \overbrace{(\mathbf{y} - X\theta)^\top (\mathbf{y} - X\theta)}^{\text{RSS}} + \underbrace{\lambda\alpha\|\theta\|_1}_{\text{variable selection}} + \underbrace{\lambda(1-\alpha)\|\theta\|_2}_{\text{shrinkage}}$$

The variable-by-variable update equation makes sense:

For each θ_j,

$$\hat{\theta}_j^{\text{glmnet}} \leftarrow \frac{S\left(\sum_{i=1}^n X_{ij}(y_i - \hat{y}_i^{(-j)}), \lambda\alpha\right)}{\sum_{i=1}^n X_{ij}^2 + \lambda(1-\alpha)} \quad \text{vs.} \quad \theta_j^{\text{MLE}} \leftarrow \frac{\sum_{i=1}^n X_{ij}\left(y_i - \sum_{k \neq j} X_{ik}\hat{\theta}_k\right)}{\sum_{i=1}^n X_{ij}^2}$$

Friedman *et al.*, Regularization Paths for Generalized Linear Models via Coordinate Descent (2010)

In practice, the greedy algorithm of `glmnet` works so well

Goal:

$$\min_{\theta} \overbrace{(\mathbf{y} - X\theta)^\top(\mathbf{y} - X\theta)}^{\text{RSS}} + \underbrace{\lambda\alpha\|\theta\|_1}_{\text{variable selection}} + \underbrace{\lambda(1-\alpha)\|\theta\|_2}_{\text{shrinkage}}$$

The variable-by-variable update equation makes sense:

For each θ_j,

$$\hat{\theta}_j \leftarrow \frac{\overset{\text{threshold}}{S}\left(\sum_{i=1}^n X_{ij} \overbrace{(y_i - y_i^{(-j)})}^{\text{residual w/o the variable } \theta_j}, \lambda\alpha\right)}{\sum_{i=1}^n X_{ij}^2 + \underbrace{\lambda(1-\alpha)}_{\text{shrinkage}}}$$

where $S(z, \tau)$ will set it to zero if $|z| < \tau$.

Cross-validation: How do we tune hyper-parameters (e.g., λ)?

1. Divide the total training data $\mathcal{D}^{\text{train}} = \{(X, y)\}$ into two parts:
   - (1) cross-validation training $\{(X, y)\}$ and
   - (2) CV testing data $\{(X^\star, y^\star)\}$
2. For each different (λ, α) combination,
   - Train coefficients θ using CV training $\{(X, y)\} \subset \mathcal{D}^{\text{train}}$
   - Test how well $\sum_j X^\star_{ij}\hat{\theta}_j$ predicts $y^\star$?
3. Choose the optimal $(\lambda^\star, \alpha^\star)$

How do we tune hyper-parameters (e.g., λ)?

Well, in R, we simply run

```
glm.cv.out <-
 glmnet::cv.glmnet(X,
          y,
          nfolds=5,
          alpha=1)
```

How do we tune hyper-parameters (e.g., λ)?

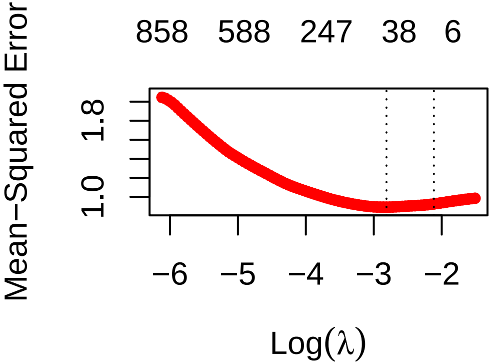
Well, in R, we simply run

```
glm.cv.out <-
 glmnet::cv.glmnet(X,
          y,
          nfolds=5,
          alpha=1)
```

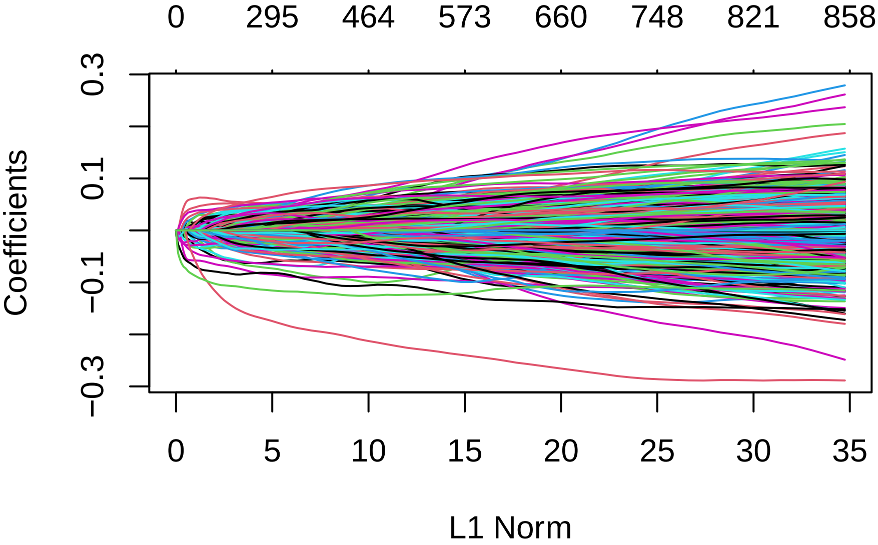
How do we tune hyper-parameters (e.g., λ)?

Well, in R, we simply run

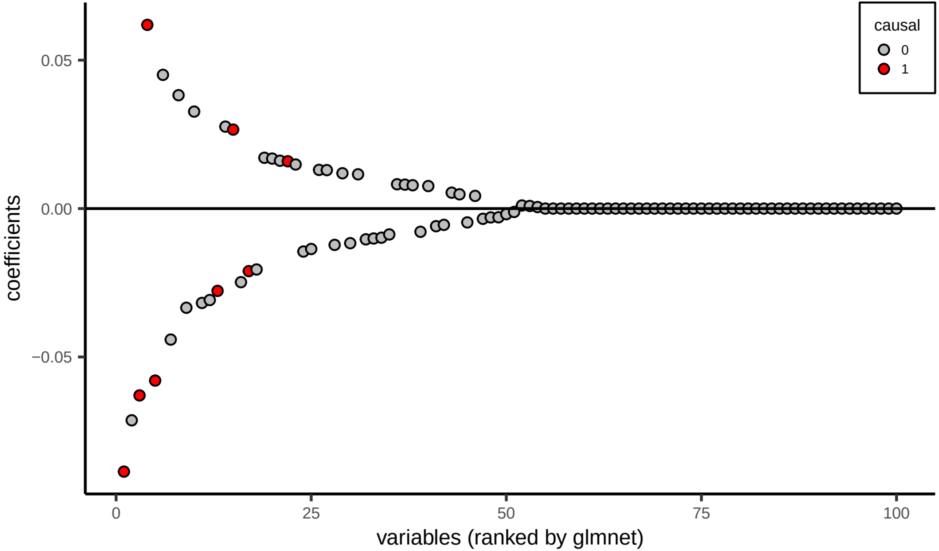
```
glm.cv.out <-
 glmnet::cv.glmnet(X,
          y,
          nfolds=5,
          alpha=1)
```



Revisit our working example with L1-regularization (`glmnet`)



At the optimal λ found by `cv.glmnet`

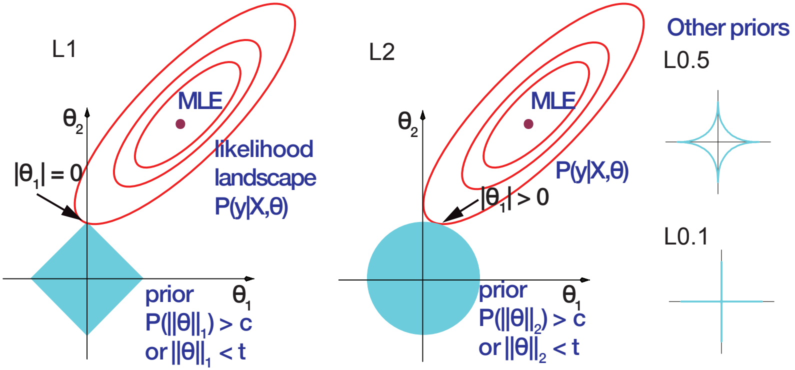


Bias-variance tradeoff explains why a regularized regression works in practice

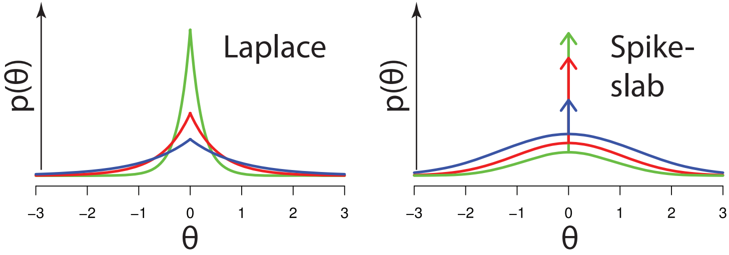
$$\min_{\theta} \quad \overbrace{(\mathbf{y} - X\theta)^{\top}(\mathbf{y} - X\theta)}^{\approx \text{bias}} + \quad \underbrace{\lambda\alpha\|\theta\|_1}_{\text{variable selection}} \quad + \underbrace{\lambda(1-\alpha)\|\theta\|_2}_{\text{shrinkage}}$$

- The second and the third terms control the model variance

Can we try out different prior (regularization)?



Bayesian spike-and-slab prior to achieve L0 norm



Hern'andez-Lobato {*et al.*} *(2015)*

Bayesian spike-and-slab prior to select variables (literally)

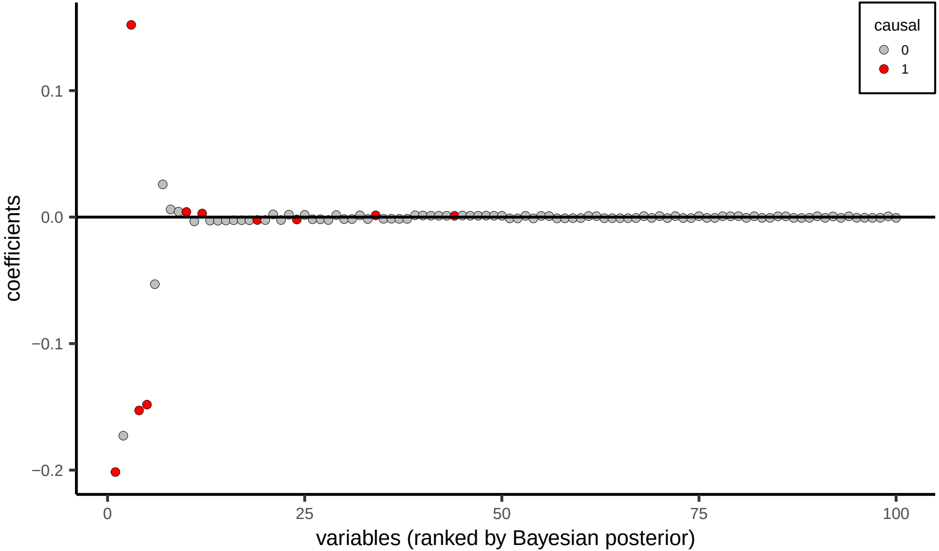
With **indicator** variables, $z_1, \ldots, z_p \in \{0, 1\}$,

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = z_1 \beta_1 \begin{pmatrix} X_{11} \\ X_{21} \\ \vdots \\ X_{n1} \end{pmatrix} + \cdots z_1 \beta_p \begin{pmatrix} X_{1p} \\ X_{2p} \\ \vdots \\ X_{np} \end{pmatrix},$$

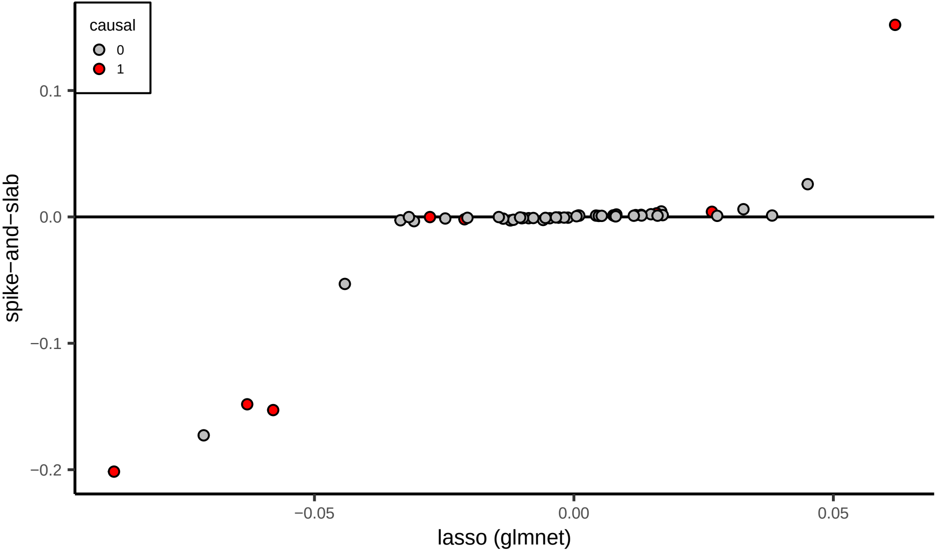
$$\mathbf{y} = X\theta + \epsilon, \quad \theta_j | z_j = 1 \sim \mathcal{N}\left(\beta_j, \sigma_j^2\right), \forall j.$$

Mitchell& Beauchamp (1988); Ishwaran& Rao (2005); (...); Carbonetto& Stephens (2012)

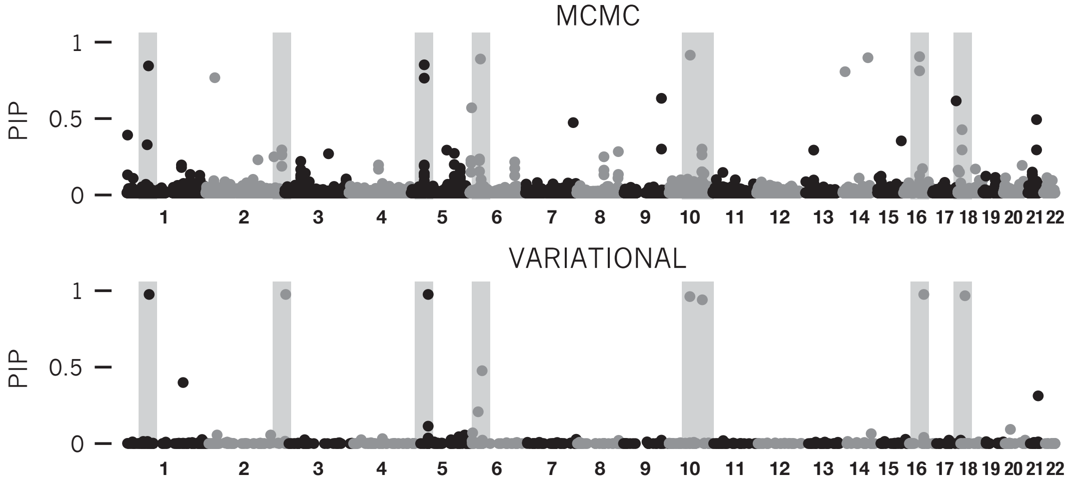
Bayesian inference with sparse Bayesian prior



Bayesian inference with sparse Bayesian prior



A Bayesian approach successfully handles high-dimensional regression problems



Carbonetto & Stephens (2012)

Today's lecture

Bias-variance tradeoff

High-dimensional multivariate regression

Knock-off filter to control False Discovery Rate

Discussion

Many variables to test for their non-zero-ness →

multiple hypothesis testing!

False discovery rate in high-dimensional variable selection

$$FDR(\tau) = \frac{\sum_{j=1}^{p} I\{|\hat{\theta}_j| > \tau \wedge \theta_j = 0\}}{\max\{1, \sum_{j=1}^{p} I\{|\hat{\theta}_j| > \tau\}\}}$$

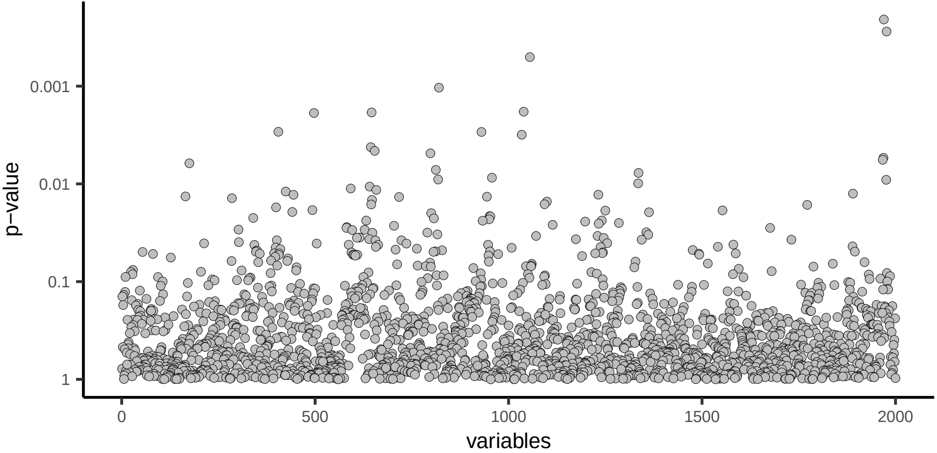
where

- $\hat{\theta}_j$: estimation using data
- θ_j: true random variable

Can we simply attempt to control FDR as in DEG analysis?

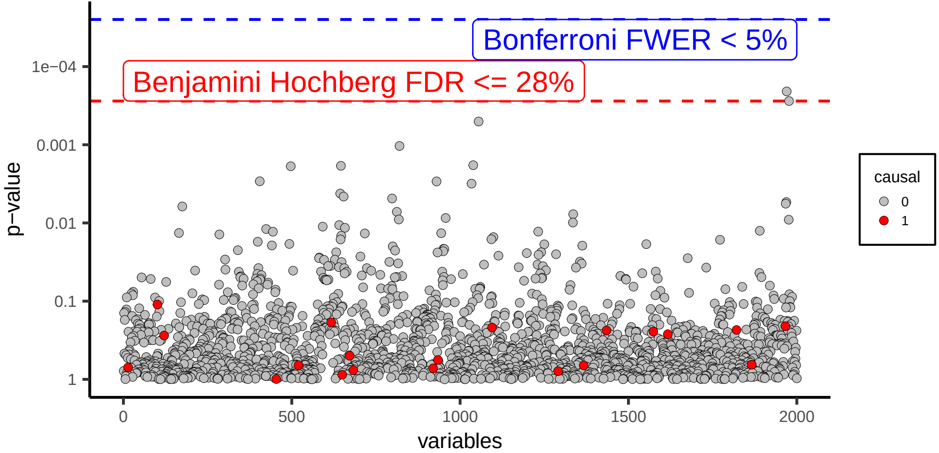
1. Perform variable-by-variable association tests
2. Combine p-values
3. Run multiple hypothesis correction (e.g., Bonferroni, Benjamini-Hochberg)

Can we simply attempt to control FDR as in DEG analysis?



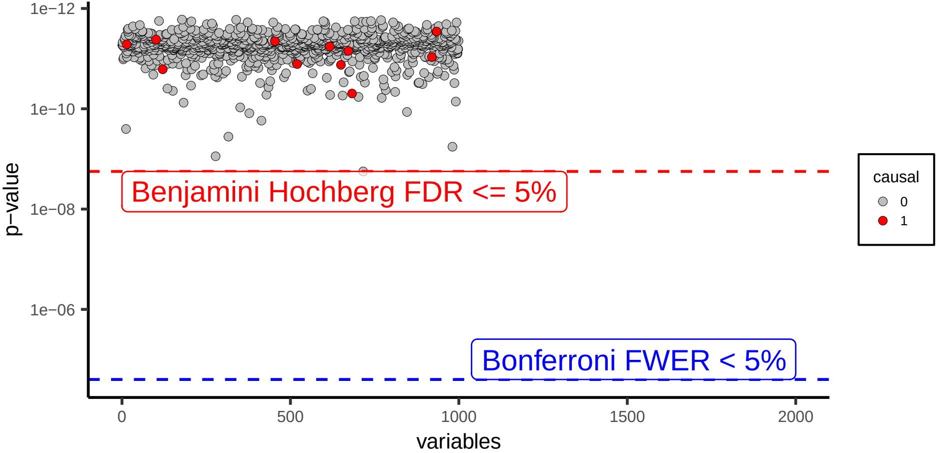
Variant-by-variant tests fail to control false discovery rate. Why?

Can we simply attempt to control FDR as in DEG analysis?



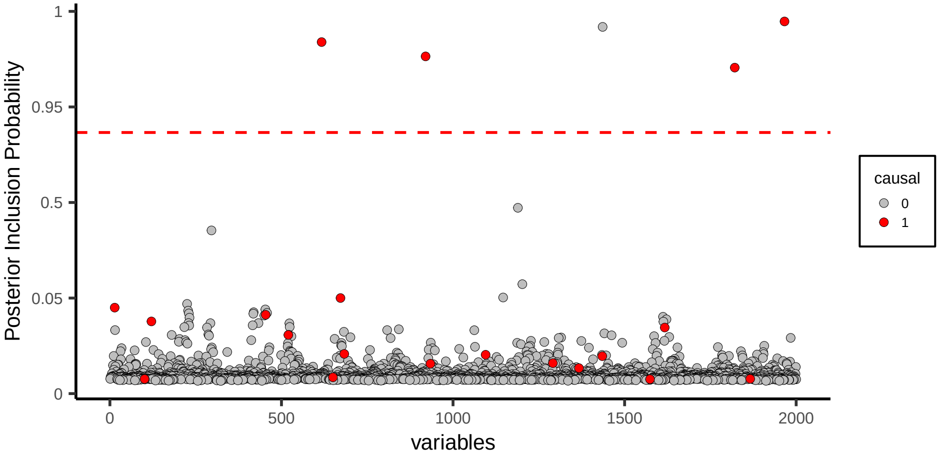
Variant-by-variant tests fail to control false discovery rate. Why?

How about using multivariate OLS results?



- What happened to the other 1000 variables?

Bayesian posterior inclusion probability can help



- Okay, but what is FDR? Can we consider $(1-$ PIP$)$ as FDR?

How we estimate the False Discovery Rate for non-zero regression coefficient?

- What should be the null distribution of regression coefficient?
- Is it t-distributed (the default option of `lm`)?

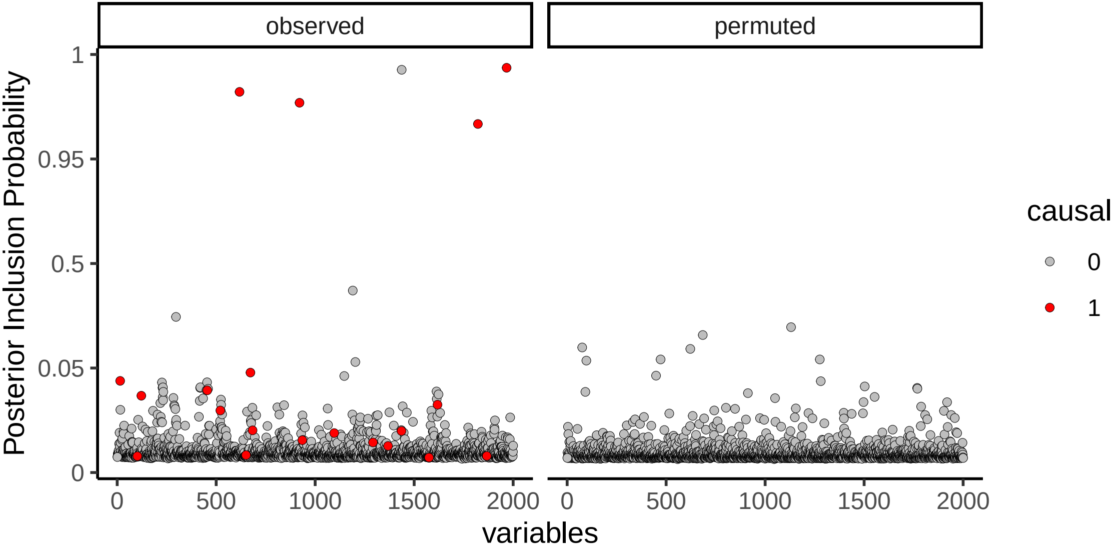
First attempt: Construct “null” regression data by sample permutation?

```r
set.seed(17)
X.perm <- apply(X, 2, sample)
```

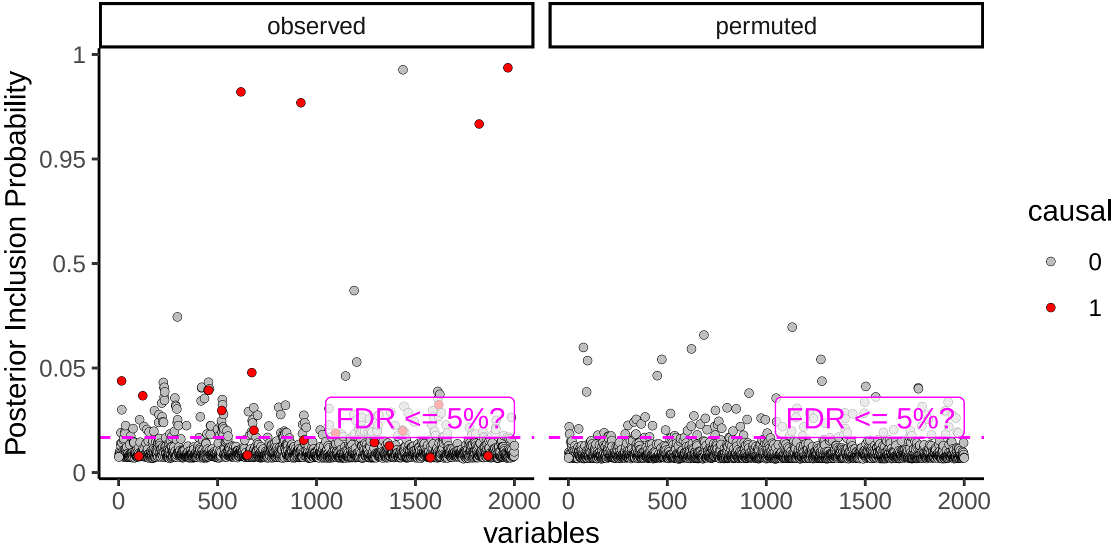
- What are we missing?
- It is not clear whether we can control Type-I error.

$$\mathbf{y} \sim [X, \quad \underbrace{\tilde{X}}_{\text{permuted}} \quad]$$

Can we learn FDR cutoff from the permuted coefficients?



Can we learn FDR cutoff from the permuted coefficients?



Can we calibrate FDR using the permuted data?

Not really...

- Let ρ_j be estimated posterior probability $p(\theta_j \neq 0|\text{data})$
- empirical False Discovery Rate $=$

$$\frac{\sum_j I\{\rho_j > \tau \wedge \theta_j = 0\}}{\sum_j I\{\rho_j > \tau\}}$$

In this example, we have 93 %

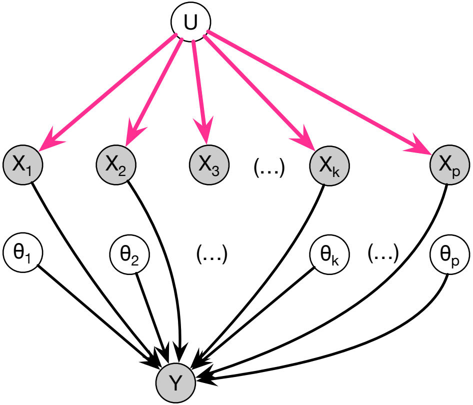
- What have we missed?

What went wrong?

1. We need to apply different threshold levels for different varaibles
2. We didn't consider correlation (col-linearity) structures between variables
3. Naive permutation steps break the covariance structure in X

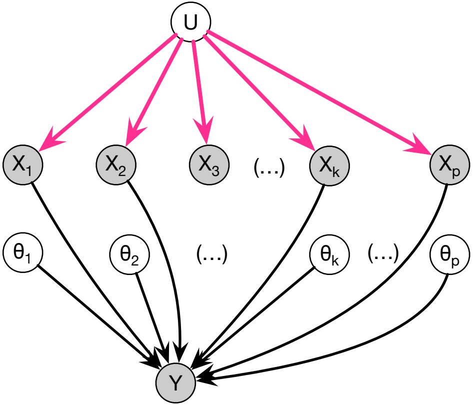
KO Idea 1: Preserve dependency structure between variables

Dependency structure in observed data

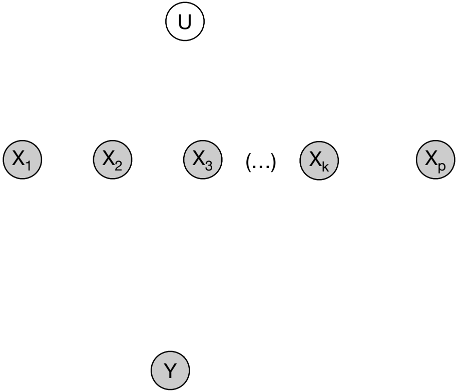


KO Idea 1: Preserve dependency structure between variables

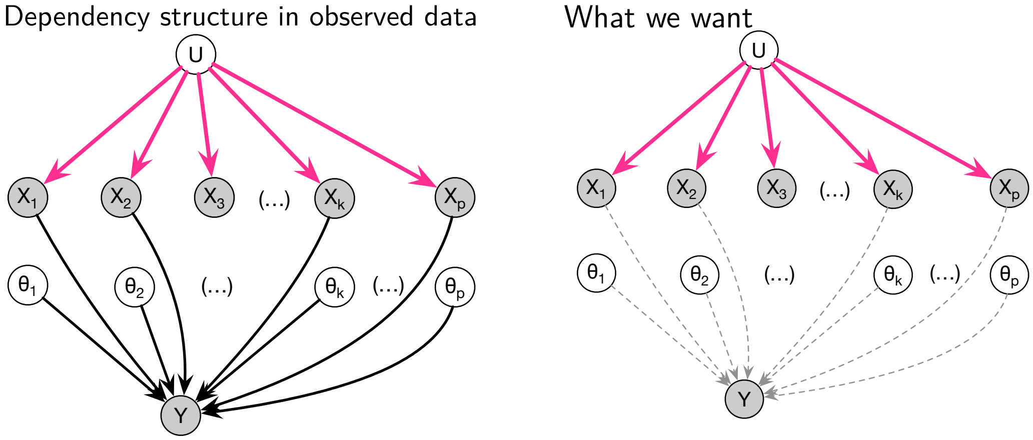
Dependency structure in observed data



What permutation did...



KO Idea 1: Preserve dependency structure between variables



Dependency structure in observed data

What we want

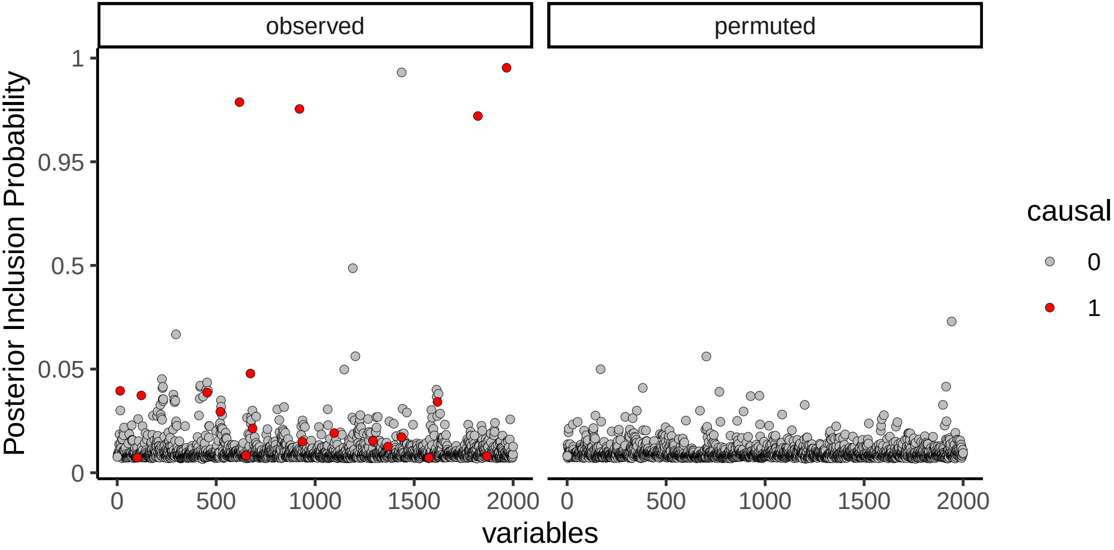
Second attempt: Construct “null” regression data by permutation preserving inter-variable dependency

```
set.seed(17)
X.perm <- X[sample(nrow(X)), ]
```

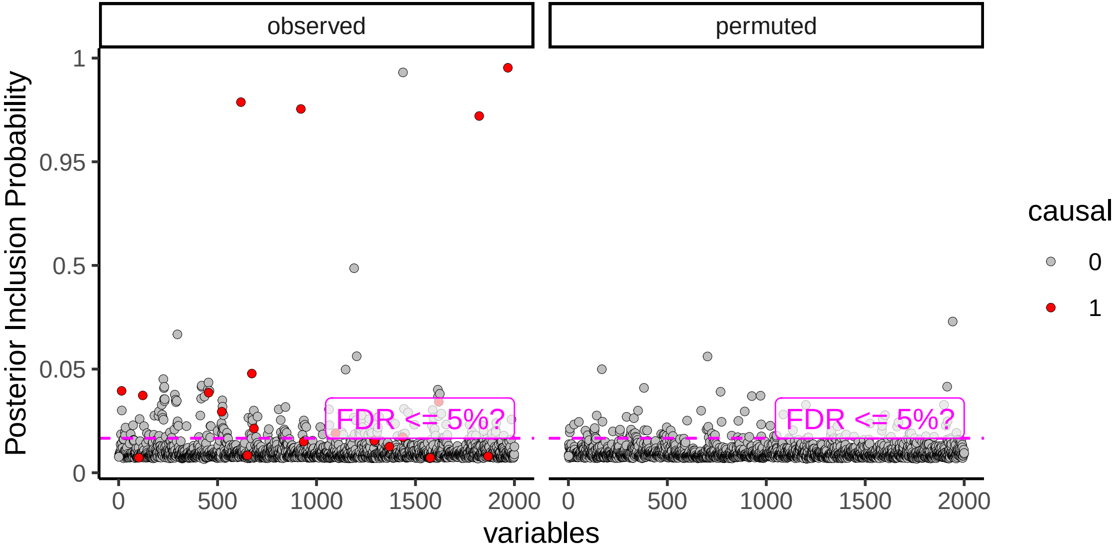
- What are we missing?
- It is not clear whether we can control Type-I error.

$$\mathbf{y} \sim [X, \quad \underbrace{\tilde{X}}_{\text{permuted}} \quad]$$

Can we learn FDR cutoff from the permuted coefficients?



Can we learn FDR cutoff from the permuted coefficients?



Can we calibrate FDR using the permuted data?

Not really...

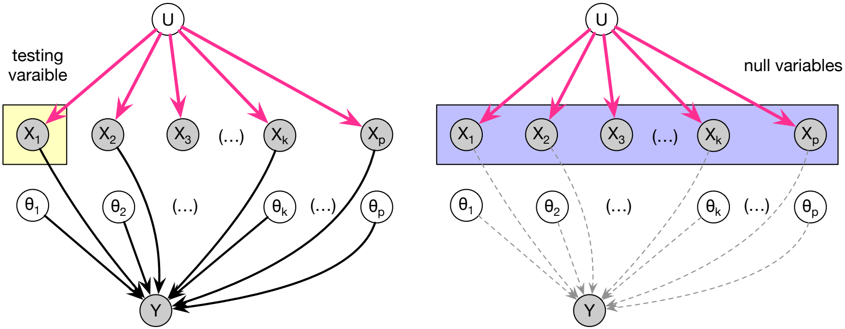
- Let ρ_j be estimated posterior probability $p(\theta_j \neq 0|\text{data})$
- empirical False Discovery Rate $=$

$$\frac{\sum_j I\{\rho_j > \tau \wedge \theta_j = 0\}}{\sum_j I\{\rho_j > \tau\}}$$

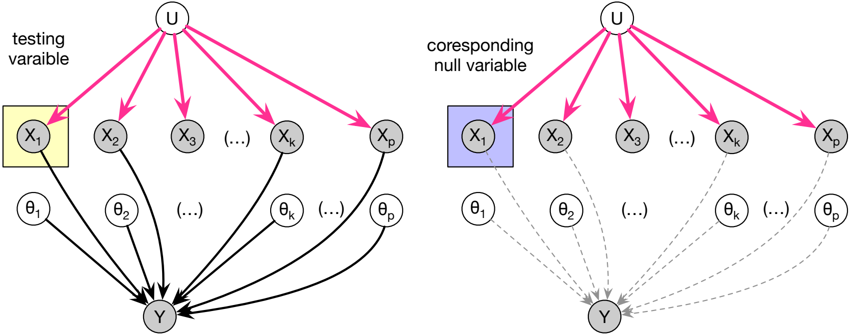
In this example, we have 93 %

- What are we still missing?

Is our comparison scheme fair? We comparing one variable against all the null variables



KO Idea 2: Matched comparison: X_j vs. $\tilde{X}_j$



Construct a knockoff matrix preserving both correlation structures

Knock-off filter

Given $X = (X_1, \dots, X_p)$, a new family of random variables, $\tilde{X} = (\tilde{X}_1, \dots, \tilde{X}_p)$ are considered a valid “knockoff” filter if

1. $\tilde{X}$ is independent of Y given X
2. distribution of $(X, \tilde{X})$ remain invariant to any swapping between the original and knockoff variables.

E.g.,

Candes, Fan, Janson, and Lv, *Panning for Gold: Model-X Knockoffs for High-dimensional Controlled Variable Selection*, (2018)

Construct a knockoff matrix preserving both correlation structures

Knock-off filter

Given $X = (X_1, \ldots, X_p)$, a new family of random variables, $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_p)$ are considered a valid “knockoff” filter if

1. $\tilde{X}$ is independent of Y given X
2. distribution of $(X, \tilde{X})$ remain invariant to any swapping between the original and knockoff variables.

E.g.,

$$(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3) \overset{d}{=} (X_1, \tilde{X}_2, X_3, \tilde{X}_1, X_2, \tilde{X}_3)$$

Candes, Fan, Janson, and Lv, *Panning for Gold: Model-X Knockoffs for High-dimensional Controlled Variable Selection*, (2018)

Construct a knockoff matrix preserving both correlation structures

Knock-off filter

Given $X = (X_1, \dots, X_p)$, a new family of random variables, $\tilde{X} = (\tilde{X}_1, \dots, \tilde{X}_p)$ are considered a valid “knockoff” filter if

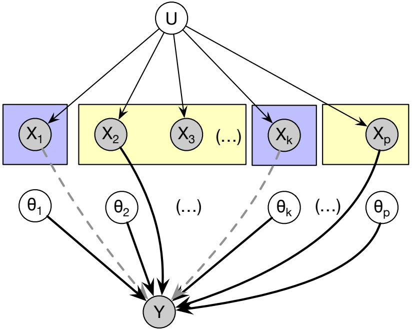
1. $\tilde{X}$ is independent of Y given X
2. distribution of $(X, \tilde{X})$ remain invariant to any swapping between the original and knockoff variables.

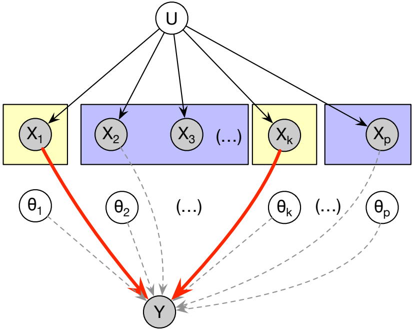
E.g.,

$$(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3) \overset{d}{=} (X_1, \tilde{X}_2, X_3, \tilde{X}_1, X_2, \tilde{X}_3)$$
$$(X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3) \overset{d}{=} (\tilde{X}_1, \tilde{X}_2, X_3, X_1, X_2, \tilde{X}_3)$$
$$(\dots)$$

Candes, Fan, Janson, and Lv, *Panning for Gold: Model-X Knockoffs for High-dimensional Controlled Variable Selection*, (2018)

Knockoff filter (null design matrix) preserves swap exchangeability





What would happen if we replace some variable X_k with its knock-off copy $\tilde{X}_k$ whilst dependence with all the other variables remain unchanged?

A reasonable approximation for knockoff construction

1. Fit $X \sim WZ$ matrix factorization

Zhu *et al.* DeepLINK: Deep learning inference using knockoffs with applications to genomics (2021)

A reasonable approximation for knockoff construction

1. Fit $X \sim WZ$ matrix factorization
2. Predict $\hat{X} \leftarrow \hat{U}\hat{Z}$

Zhu *et al.* DeepLINK: Deep learning inference using knockoffs with applications to genomics (2021)

A reasonable approximation for knockoff construction

1. Fit $X \sim WZ$ matrix factorization
2. Predict $\hat{X} \leftarrow \hat{U}\hat{Z}$
3. Take residuals $\epsilon = X - \hat{X}$

Zhu *et al.* DeepLINK: Deep learning inference using knockoffs with applications to genomics (2021)

A reasonable approximation for knockoff construction

1. Fit $X \sim WZ$ matrix factorization
2. Predict $\hat{X} \leftarrow \hat{U}\hat{Z}$
3. Take residuals $\epsilon = X - \hat{X}$
4. Add permuted residuals $\tilde{\epsilon}$, i.e., $\tilde{X} = \hat{X} + \tilde{\epsilon}$

Zhu *et al.* DeepLINK: Deep learning inference using knockoffs with applications to genomics (2021)

Knockoff statistics

$$\mathbf{y} \sim [X, \underset{\text{knockoff}}{\tilde{X}}]$$

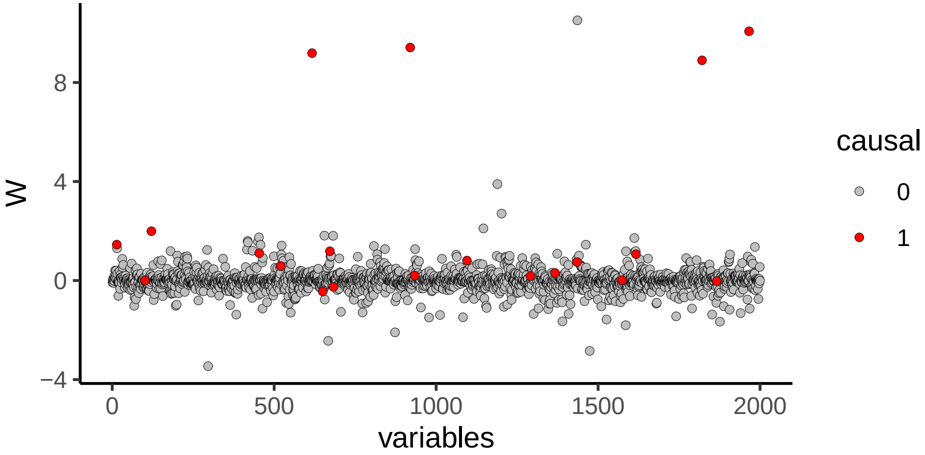
For each variable j (Lasso):

$$W_j = |\hat{\theta}_j| - |\tilde{\theta}_j|$$

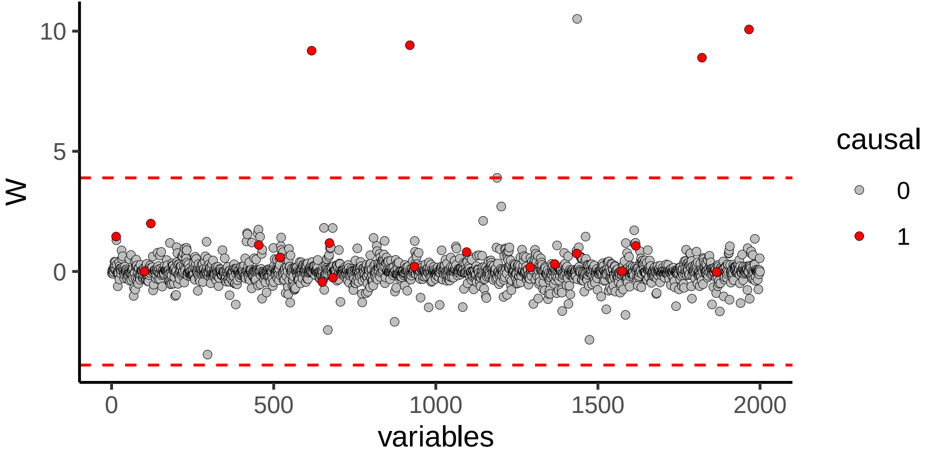
For each variable j (Bayesian PIP):

$$W_j = \hat{\rho}_j - \tilde{\rho}_j$$

Knockoff statistics: What is FDR here?



Knockoff statistics: What is FDR here?



Today's lecture

Bias-variance tradeoff

High-dimensional multivariate regression

Knock-off filter to control False Discovery Rate

Discussion

Other methods that we haven't had a chance to discuss

- Ensemble learning
  - Boosting, Model-averaging
- Bayesian non-parametric models
  - We select models by *not* selecting a model
  - Gaussian process
- Deep neural network model