Introduction to Statistical Learning and Machine Learning

Chap 2 - Linear Regression(1)

Yanwei Fu
SDS, Fudan University

Chap 2 - Linear Regression(1)

Main Content

1. Simple Linear Model
2. Least Squares;
3. The Bias-Variance tradeoff;

Regression

simple linear regression;

multiple regression;

logistic regression;

poisson regression

Chap 2 - Linear Regression(1)

Recap & Bias-Variance Trade-off

Recap: Notations of Supervised Learning (1)

We use uppercase letters such as X, Y or G when referring to the generic aspects of a variable.

Recap: Notations of Supervised Learning (1)

We use uppercase letters such as X, Y or G when referring to the generic aspects of a variable.

X input variables , a.k.a., features, predictors, independent variables.

Y output variables, a.k.a., response or dependent variable.

Recap: Notations of Supervised Learning (1)

We use uppercase letters such as X, Y or G when referring to the generic aspects of a variable.

X input variables , a.k.a., features, predictors, independent variables.

Y output variables, a.k.a., response or dependent variable.

$Y = f(X) + \epsilon$ ϵ captures measurement errors and other discrepancies.

$l : X \to Y$ Loss function, $l\left(y, y^{'}\right)$ is the cost of predicting $y^{'}$ if y is correct.

Recap: Notations of Supervised Learning (1)

We use uppercase letters such as X, Y or G when referring to the generic aspects of a variable.

X input variables , a.k.a., features, predictors, independent variables.

Y output variables, a.k.a., response or dependent variable.

$Y = f(X) + \epsilon$ ϵ captures measurement errors and other discrepancies.

$l : X \to Y$ Loss function, $l\left(y, y^{'}\right)$ is the cost of predicting $y^{'}$ if y is correct.

Regression when we predict quantitative outputs (infinite set);
Classification when we predict qualitative outputs (finite set, e.g. Group labels, Ordered,)

Training set: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ sampled from the joint distribution (*X*, *Y*).

Recap: Notations of Supervised Learning (1)

We use uppercase letters such as X, Y or G when referring to the generic aspects of a variable.

X input variables , a.k.a., features, predictors, independent variables.

Y output variables, a.k.a., response or dependent variable.

$Y = f(X) + \epsilon$ ϵ captures measurement errors and other discrepancies.

$l : X \to Y$ Loss function, $l\left(y, y^{'}\right)$ is the cost of predicting $y^{'}$ if y is correct.

Regression when we predict quantitative outputs (infinite set);
Classification when we predict qualitative outputs (finite set, e.g. Group labels, Ordered,)

Training set: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$ sampled from the joint distribution (*X*, *Y*).

i.i.d: Independent and identically distributed random variables.
A sequence or other collection of random variables is i.i.d. if each random variable has the same probability distribution as the others and all are ***mutually*** independent.

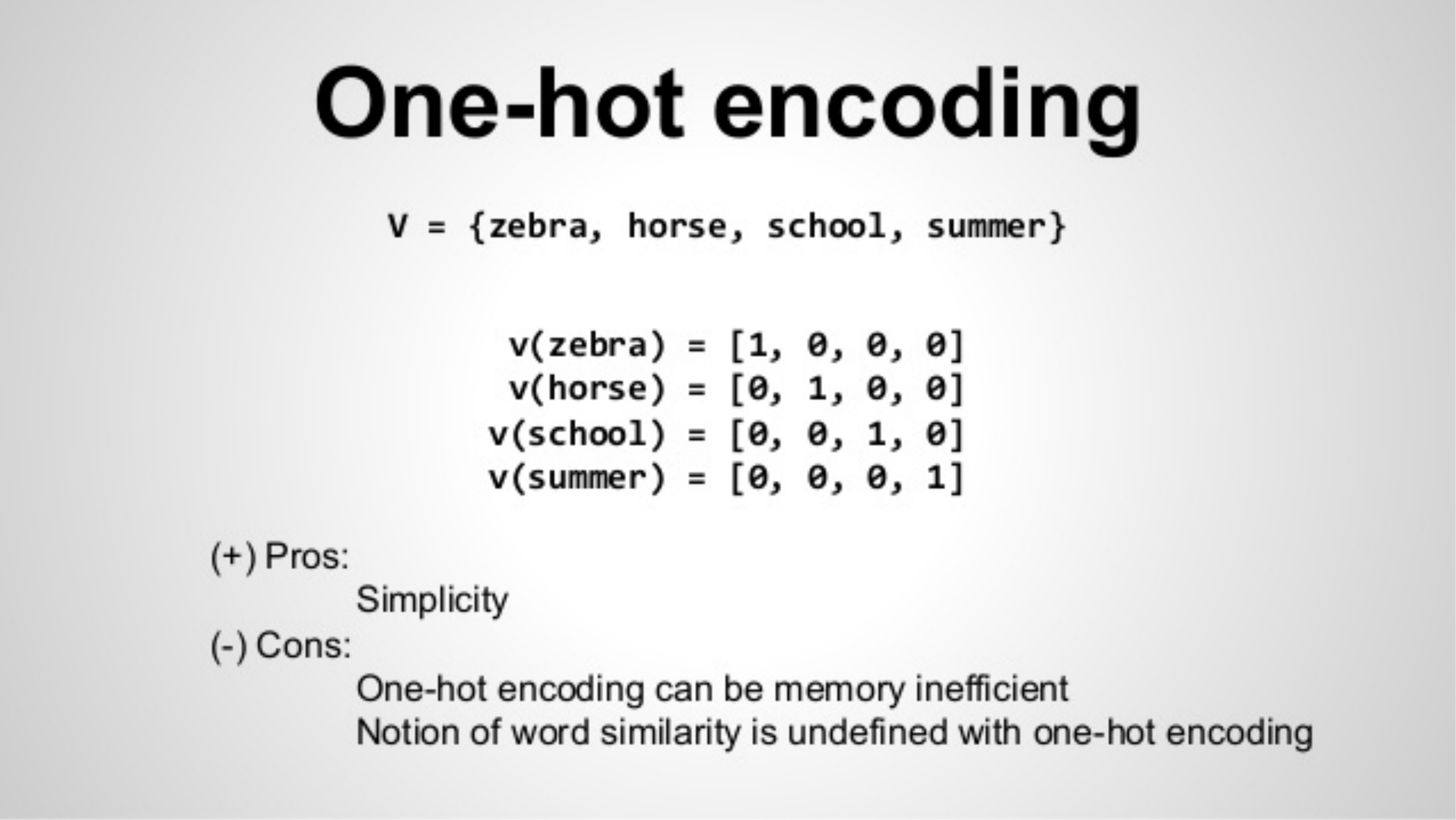
$$\mathrm{P}(A \cap B) = \mathrm{P}(A)\mathrm{P}(B).$$

Recap: Notations of Supervised Learning

Matrices are represented by bold uppercase letters. $\boldsymbol{X}$

Observed values are written in lowercase; hence the *i*-th observed value of X is written as x_i

Dummy Variable： K-level qualitative variable is represented by a vector of K binary variables or bits, only one of which is “on" at a time. a.k.a. *One-hot vector* Vs. Distributed Representation in Deep Learning.



One-hot encoding

```
V = {zebra, horse, school, summer}

 v(zebra) = [1, 0, 0, 0]
 v(horse) = [0, 1, 0, 0]
v(school) = [0, 0, 1, 0]
v(summer) = [0, 0, 0, 1]
```

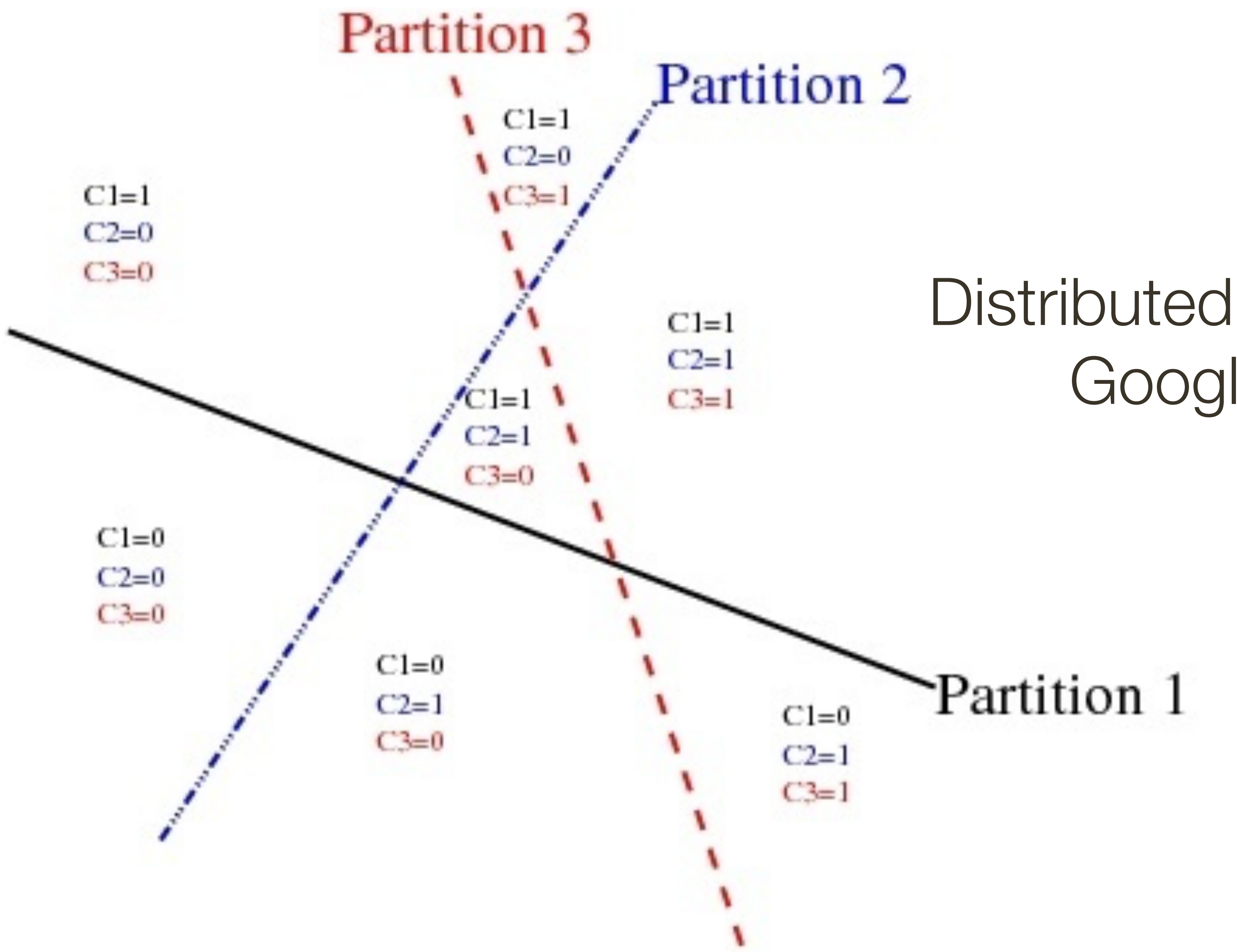
(+) Pros:

Simplicity

(-) Cons:

One-hot encoding can be memory inefficient

Notion of word similarity is undefined with one-hot encoding



Distributed Representation.
Google Word2vec

Some Important Concepts

Overfitting: a method yields a small training MSE but a large test MSE, we are said to be overfitting the data

This happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f.

Underfitting: a method function is not sufficient to fit the training samples. (Not small enough MSE on training data).

Some Important Concepts

Mean squared error (MSE), $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$

Overfitting: a method yields a small training MSE but a large test MSE, we are said to be overfitting the data

This happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f.

Underfitting: a method function is not sufficient to fit the training samples. (Not small enough MSE on training data).

Some Important Concepts

Mean squared error (MSE), $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$

We are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.

Overfitting: a method yields a small training MSE but a large test MSE, we are said to be overfitting the data

This happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f.

Underfitting: a method function is not sufficient to fit the training samples. (Not small enough MSE on training data).

Some Important Concepts

Mean squared error (MSE), $$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

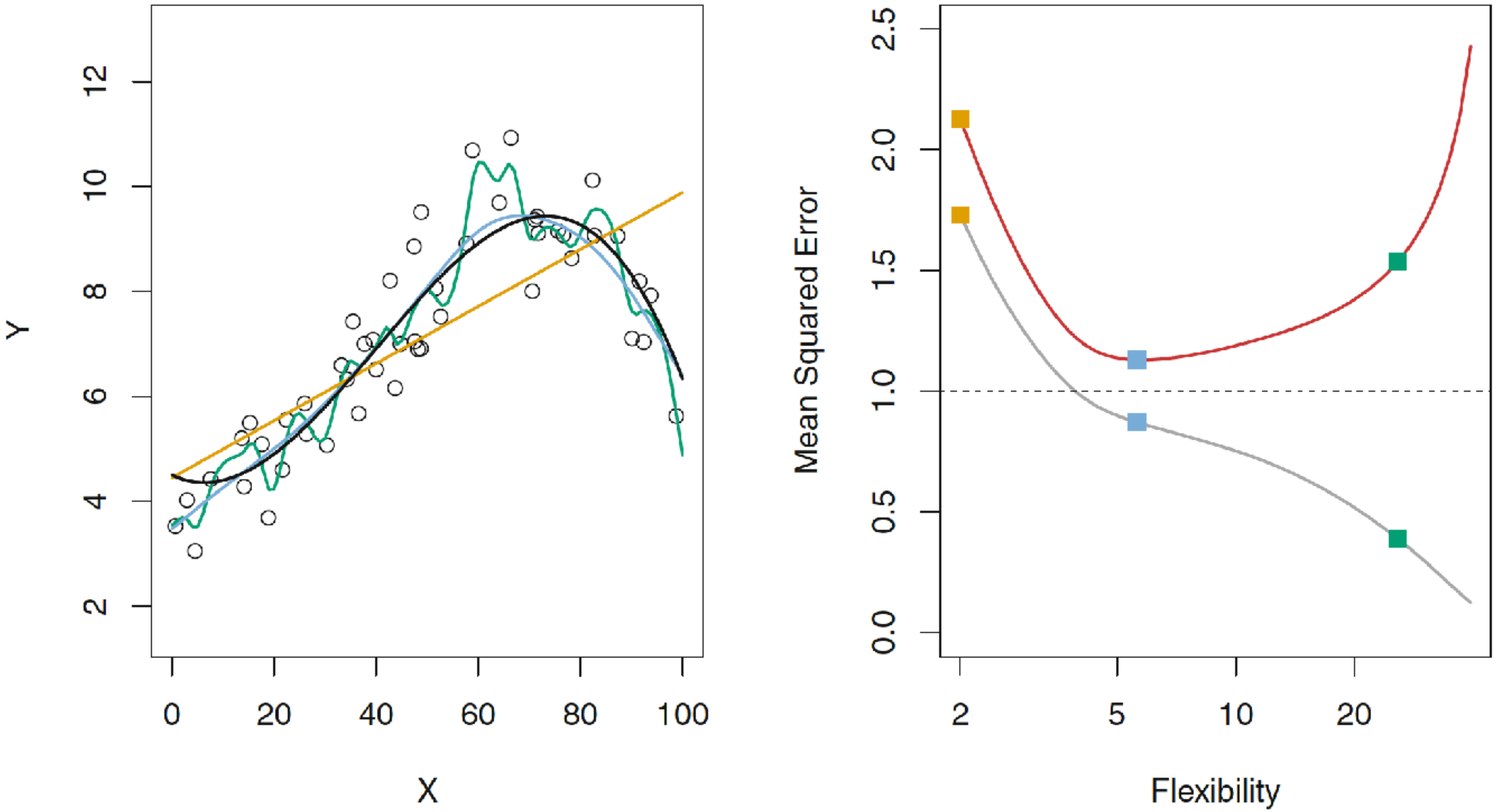
We are interested in the accuracy of the predictions that we obtain when we apply our method to previously unseen test data.

Test MSE $\mathrm{Ave}(y_0 - \hat{f}(x_0))^2,$ (x_0, y_0) is a previously unseen test observation.

Overfitting: a method yields a small training MSE but a large test MSE, we are said to be overfitting the data

This happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f.

Underfitting: a method function is not sufficient to fit the training samples. (Not small enough MSE on training data).



Left: Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves). Right: Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.

Simple Linear regression with two degrees of freedom.

Simple Linear regression with two degrees of freedom.

Expectation operator: $\mathrm{E}[\cdot]$ Constants, Monotonicity, Linearity.

$$\mathrm{E}[X+c] = \mathrm{E}[X] + c$$
$$\mathrm{E}[X+Y] = \mathrm{E}[X] + \mathrm{E}[Y]$$
$$\mathrm{E}[aX] = a\,\mathrm{E}[X]$$

$\mathrm{E}[c] = c.$

$X \leq Y$ Almost surely $\mathrm{E}[X] \leq \mathrm{E}[Y]$

Simple Linear regression with two degrees of freedom.

Expectation operator: $\mathbf{E}[\cdot]$ Constants, Monotonicity, Linearity.

$$\mathbf{E}[X+c]=\mathbf{E}[X]+c$$
$$\mathbf{E}[X+Y]=\mathbf{E}[X]+\mathbf{E}[Y]$$
$$\mathbf{E}[aX]=a\,\mathbf{E}[X]$$

$\mathbf{E}[c]=c.$

$X \leq Y$ Almost surely $\mathbf{E}[X] \leq \mathbf{E}[Y]$

Conditional expectation, *For any two discrete random variables X, Y.*

$$\mathrm{E}[X \mid Y=y]=\sum_x x \cdot \mathrm{P}(X=x \mid Y=y), \qquad f: y \mapsto \mathbf{E}(X \mid Y=y).$$

We call it *conditional expectation of X with respect to Y.* $\mathrm{E}[X]=\mathrm{E}[\mathrm{E}[X \mid Y]].$

The number of **degrees of freedom** (flexibility) is the number of values in the final calculation of a statistic that are free to vary.

Simple Linear regression with two degrees of freedom.

Expectation operator: $\mathrm{E}[\cdot]$ Constants, Monotonicity, Linearity.

$$\mathrm{E}[c] = c.$$

$X \leq Y$ Almost surely $\mathrm{E}[X] \leq \mathrm{E}[Y]$

$$\begin{aligned}\mathrm{E}[X+c] &= \mathrm{E}[X] + c\\ \mathrm{E}[X+Y] &= \mathrm{E}[X] + \mathrm{E}[Y]\\ \mathrm{E}[aX] &= a\,\mathrm{E}[X]\end{aligned}$$

Conditional expectation, *For any two discrete random variables X, Y.*

$$\mathrm{E}[X \mid Y=y] = \sum_x x \cdot \mathrm{P}(X=x \mid Y=y), \qquad f : y \mapsto \mathrm{E}(X \mid Y=y).$$

We call it *conditional expectation of X with respect to Y.* $\mathrm{E}[X] = \mathrm{E}[\mathrm{E}[X \mid Y]].$

The number of **degrees of freedom** (flexibility) is the number of values in the final calculation of a statistic that are free to vary.

Simple Linear regression with two degrees of freedom.

Expectation operator: $\mathrm{E}[\cdot]$ Constants, Monotonicity, Linearity.

$$\mathrm{E}[c] = c.$$

$X \leq Y$ Almost surely $\mathrm{E}[X] \leq \mathrm{E}[Y]$

$$\begin{aligned} \mathrm{E}[X+c] &= \mathrm{E}[X] + c \\ \mathrm{E}[X+Y] &= \mathrm{E}[X] + \mathrm{E}[Y] \\ \mathrm{E}[aX] &= a\,\mathrm{E}[X] \end{aligned}$$

If the probability distribution of X admits a probability density function $f(x)$, then the expected value can be computed as

$$\mathrm{E}[X] = \int_{-\infty}^{\infty} x f(x)\,\mathrm{d}x.$$

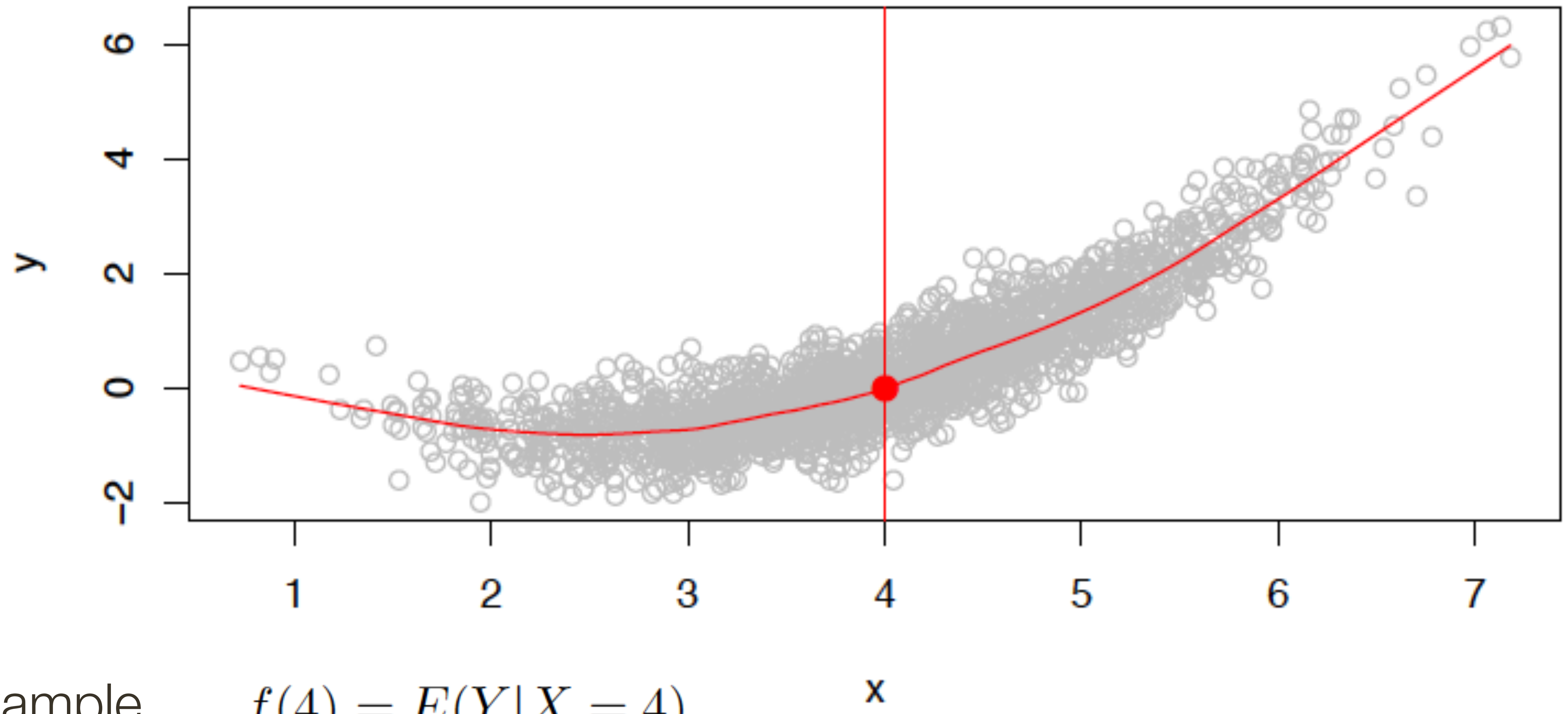
Conditional expectation, *For any two discrete random variables X, Y.*

$$\mathrm{E}[X \mid Y = y] = \sum_x x \cdot \mathrm{P}(X = x \mid Y = y), \qquad f : y \mapsto \mathrm{E}(X \mid Y = y).$$

We call it *conditional expectation of X with respect to Y.* $\mathrm{E}[X] = \mathrm{E}[\mathrm{E}[X \mid Y]].$

Bias-Variance Trade-off(1)

Is there an ideal f(X)?



Take X=4 as and example, $f(4) = E(Y|X=4)$

$f(x) = E(Y|X=x)$ is called the regression function.

We minimise least square errors over all points X=x

$$E[(Y-\hat{f}(X))^2|X=x] = \underbrace{[f(x)-\hat{f}(x)]^2}_{Reducible} + \underbrace{\mathrm{Var}(\epsilon)}_{Irreducible}$$

Bias-Variance Trade-off(2)

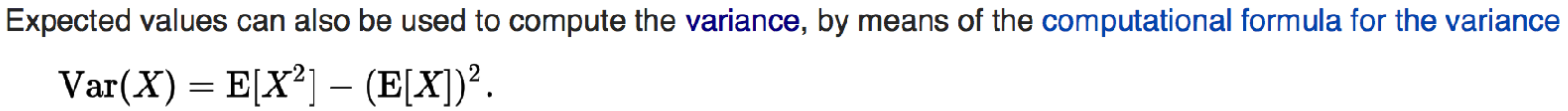
$$E[(Y-\hat{f}(X))^2|X=x] = \underbrace{[f(x)-\hat{f}(x)]^2}_{\textit{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\textit{Irreducible}}$$

$$\hat{Y} = \hat{f}(X),$$

$$\begin{aligned} E(Y-\hat{Y})^2 &= E[f(X)+\epsilon-\hat{f}(X)]^2 \\ &= \underbrace{[f(X)-\hat{f}(X)]^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}(\epsilon)}_{\text{Irreducible}} \quad , \end{aligned}$$

$E(Y-\hat{Y})^2$ represents the average, or expected value, of the squared difference between the predicted and actual value of *Y.*

$\mathrm{Var}(\epsilon)$ represents the variance associated with the error term ϵ.

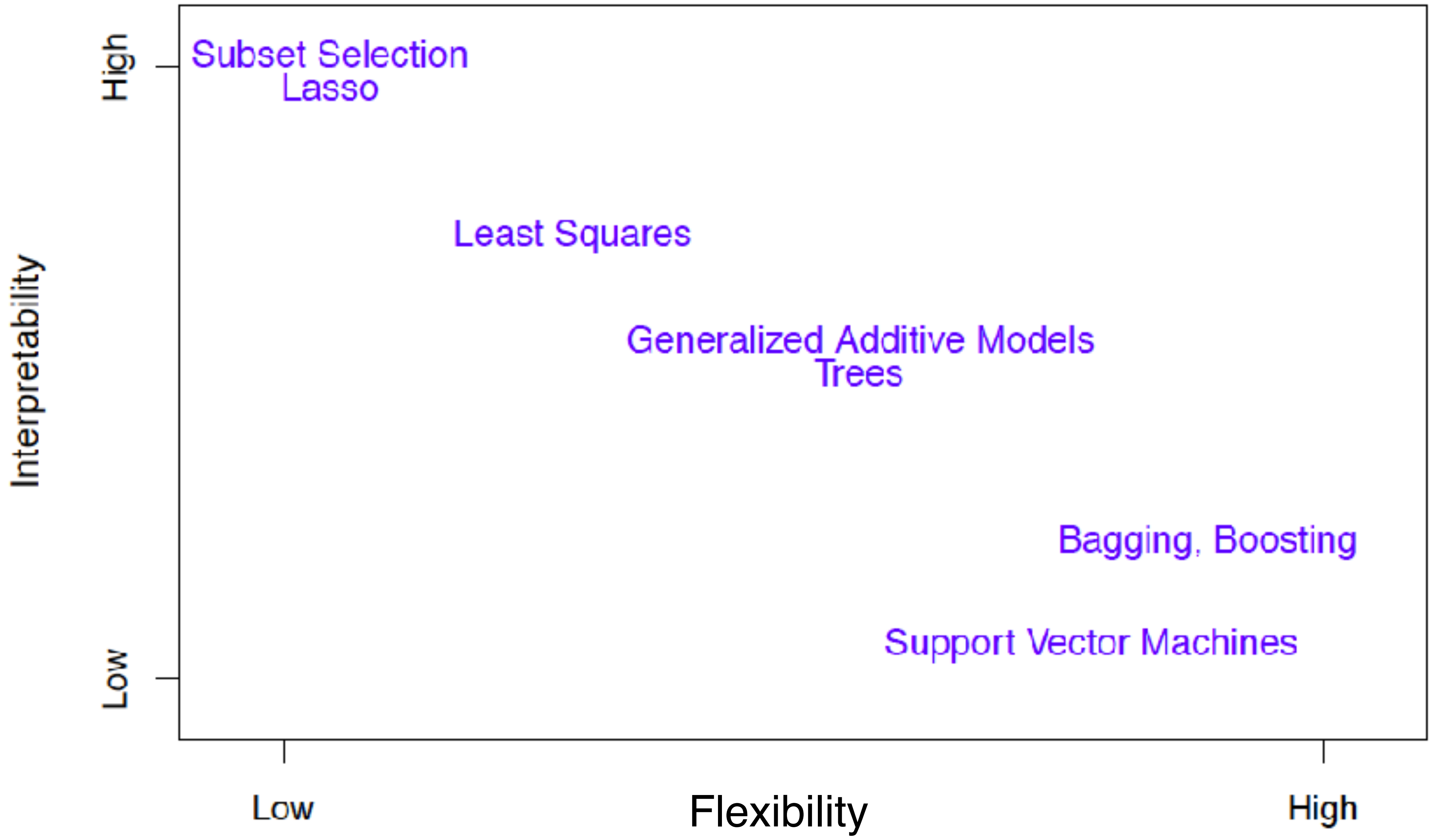


Expected values can also be used to compute the variance, by means of the computational formula for the variance

$$\mathbf{Var}(X) = \mathbf{E}[X^2] - (\mathbf{E}[X])^2.$$

Some Trade-off

- Prediction accuracy versus interpretability.
  - Linear models are easy to interpret; thin-plate splines(薄板样条插值) are not.
- Good fit versus over-fit or under-fit.
  - How do we know when the fit is just right?
- Parsimony versus black-box.
  - We often prefer a simpler model involving fewer variables over a black-box predictor involving them all.



Chap 2 - Linear Regression(1)

Non-parametric methods.
Vs. Parametric methods

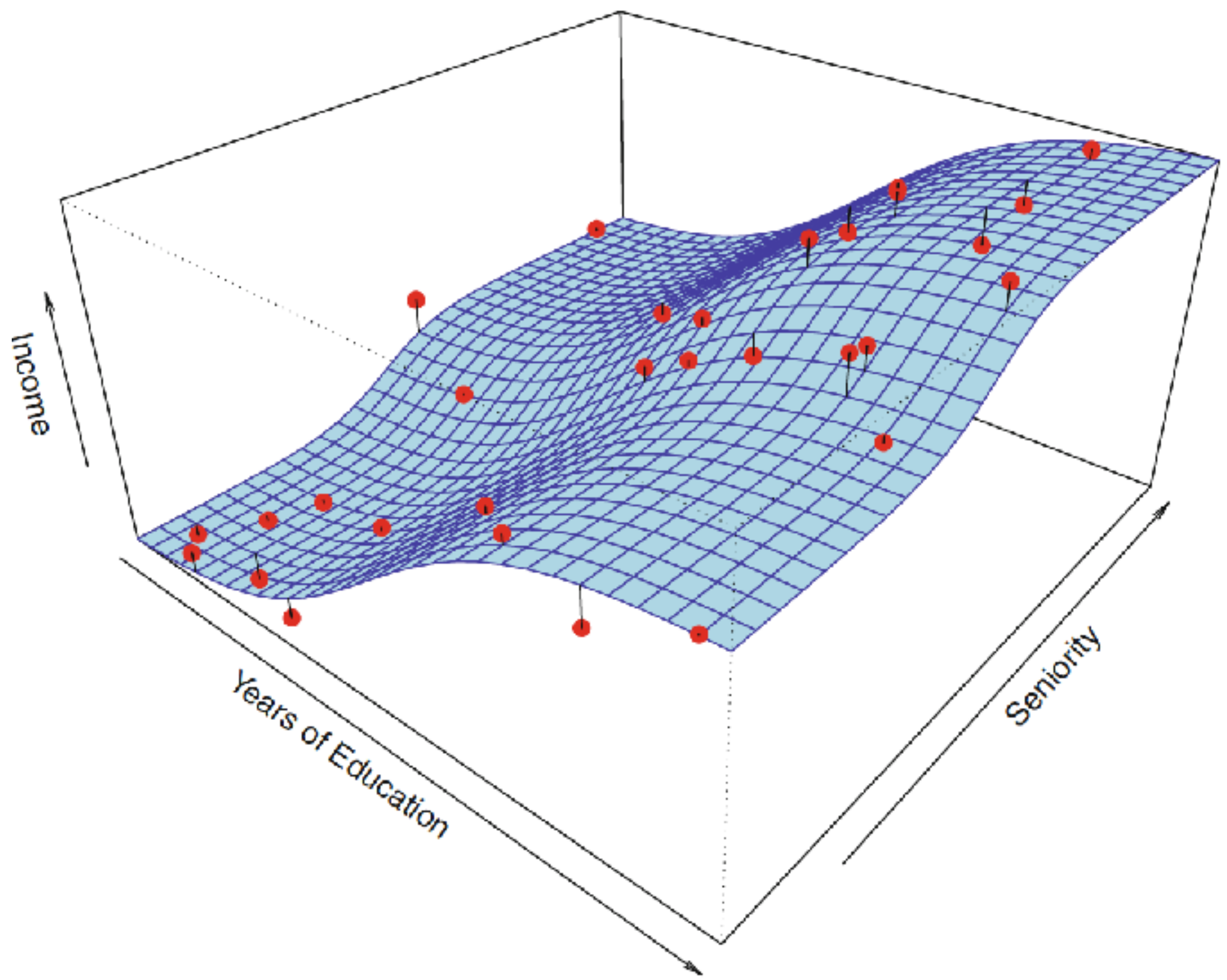
Two basic ideas of *How Do We Estimate f?*

- **Parametric Methods**: Linear Least Square -> generalized linear models
  1. we make an assumption about the functional form, or shape, of f $f(X) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p.$
  2. we use the training data to fit the model (parameters); $Y \approx \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p.$

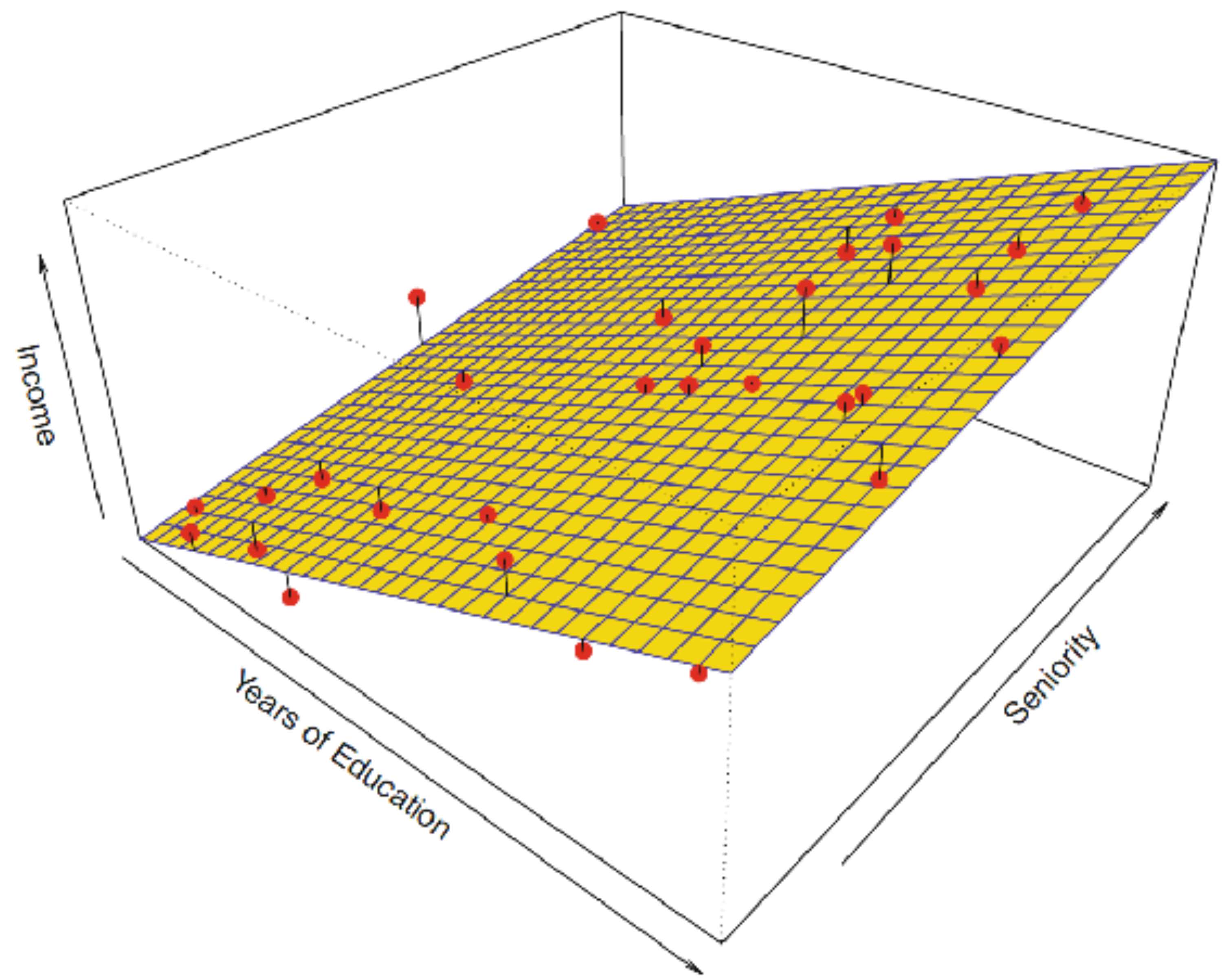
- **Non-parametric Methods**: Nearest Neighbors -> kernel method and SVM
  1. We do not make explicit assumptions about the functional form of f. Instead they seek an estimate of f that gets as close to the data points as possible without being too rough or wiggly.
  2. Not make explicit assumptions about the functional form of f.

An example of Parametric Vs. Non-parametric methods

The observations are displayed in red; the yellow plane indicates the fitted model;



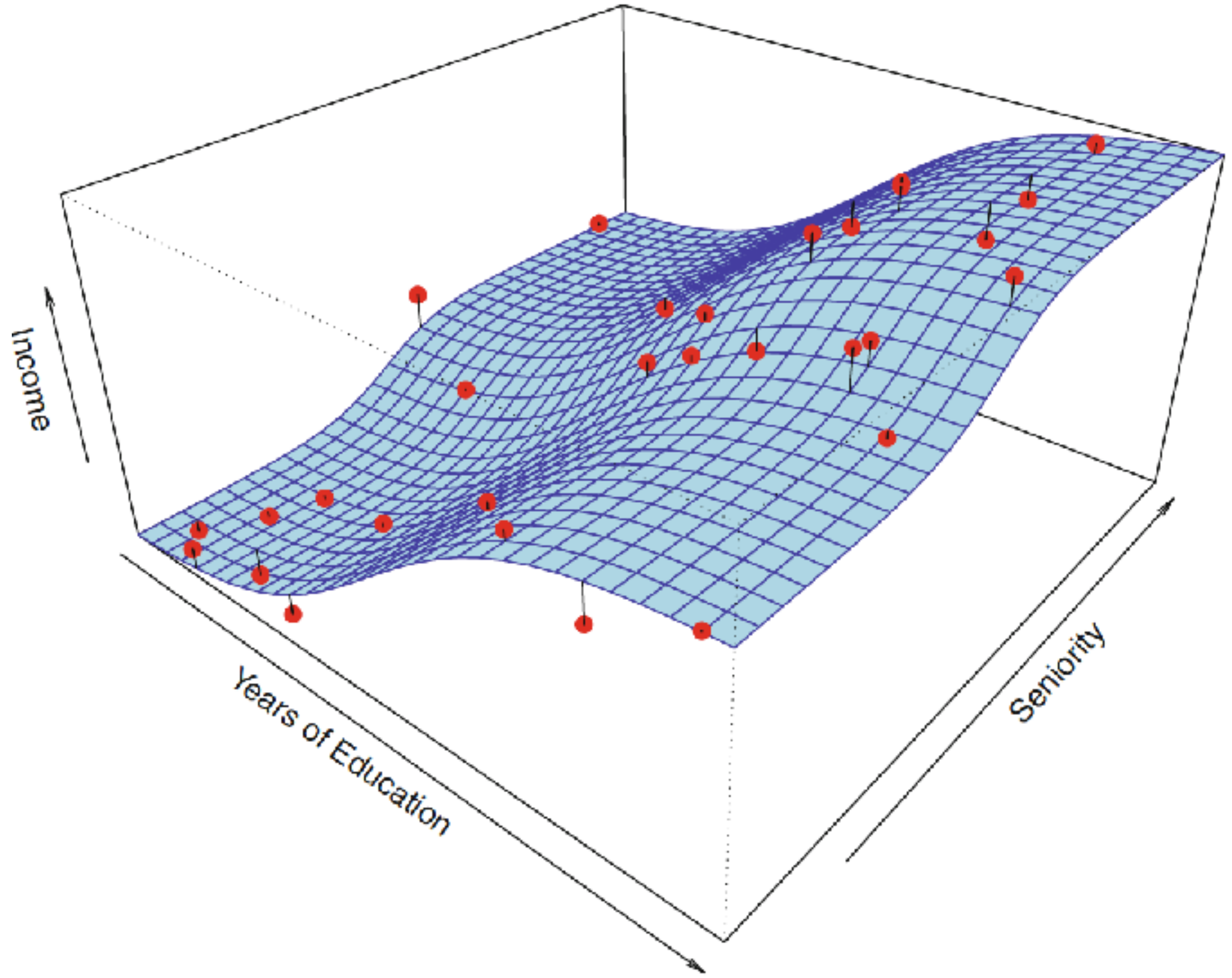
A linear model fit by
least squares to
the Income data



The plot displays income as a function of years of education and seniority in the Income data set. The blue surface represents the true underlying relationship between income and years of education and seniority, which is known since the data are simulated. The red dots indicate the observed values of these quantities for 30 individuals.

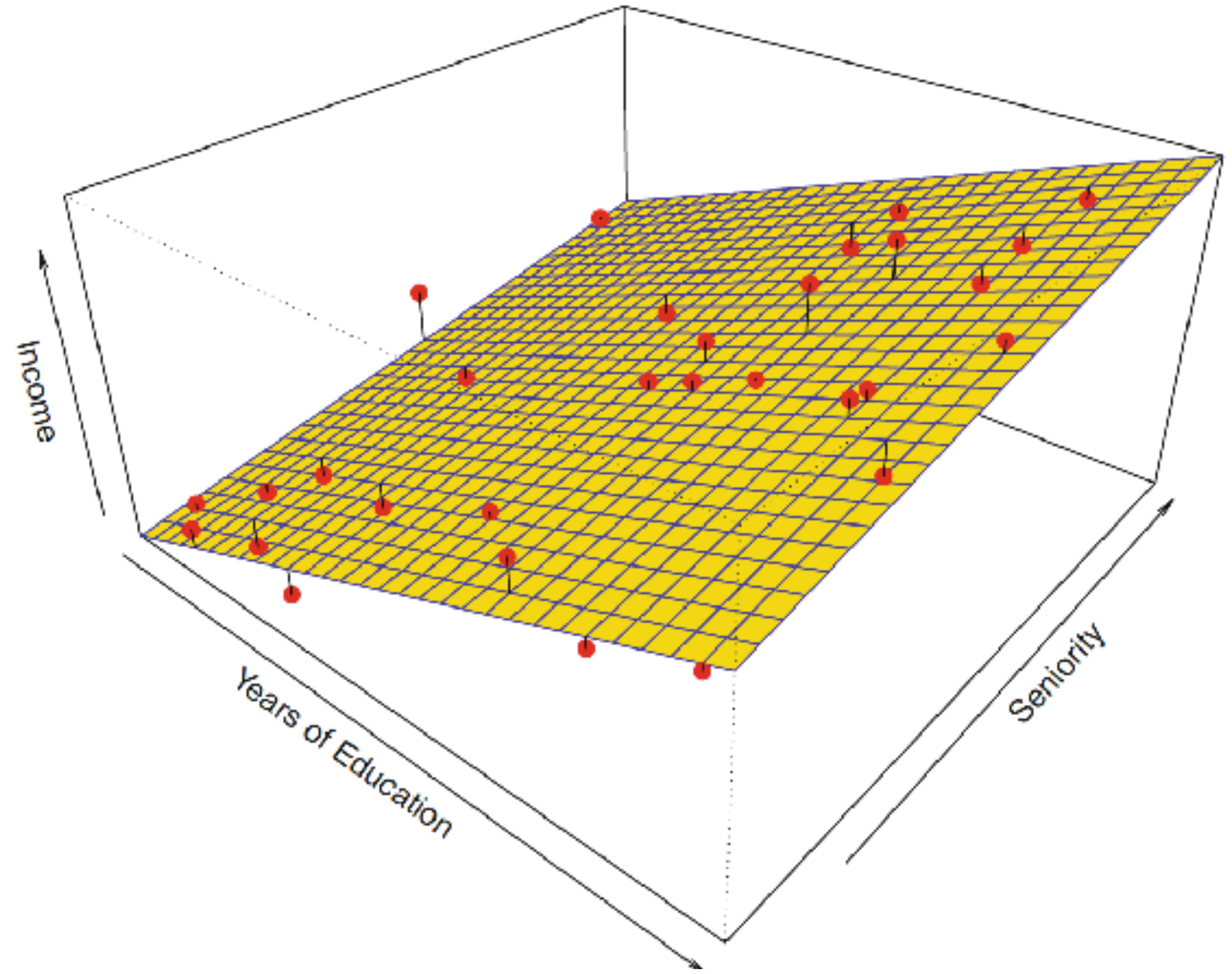
An example of Parametric Vs. Non-parametric methods

The observations are displayed in red; the yellow plane indicates the fitted model;

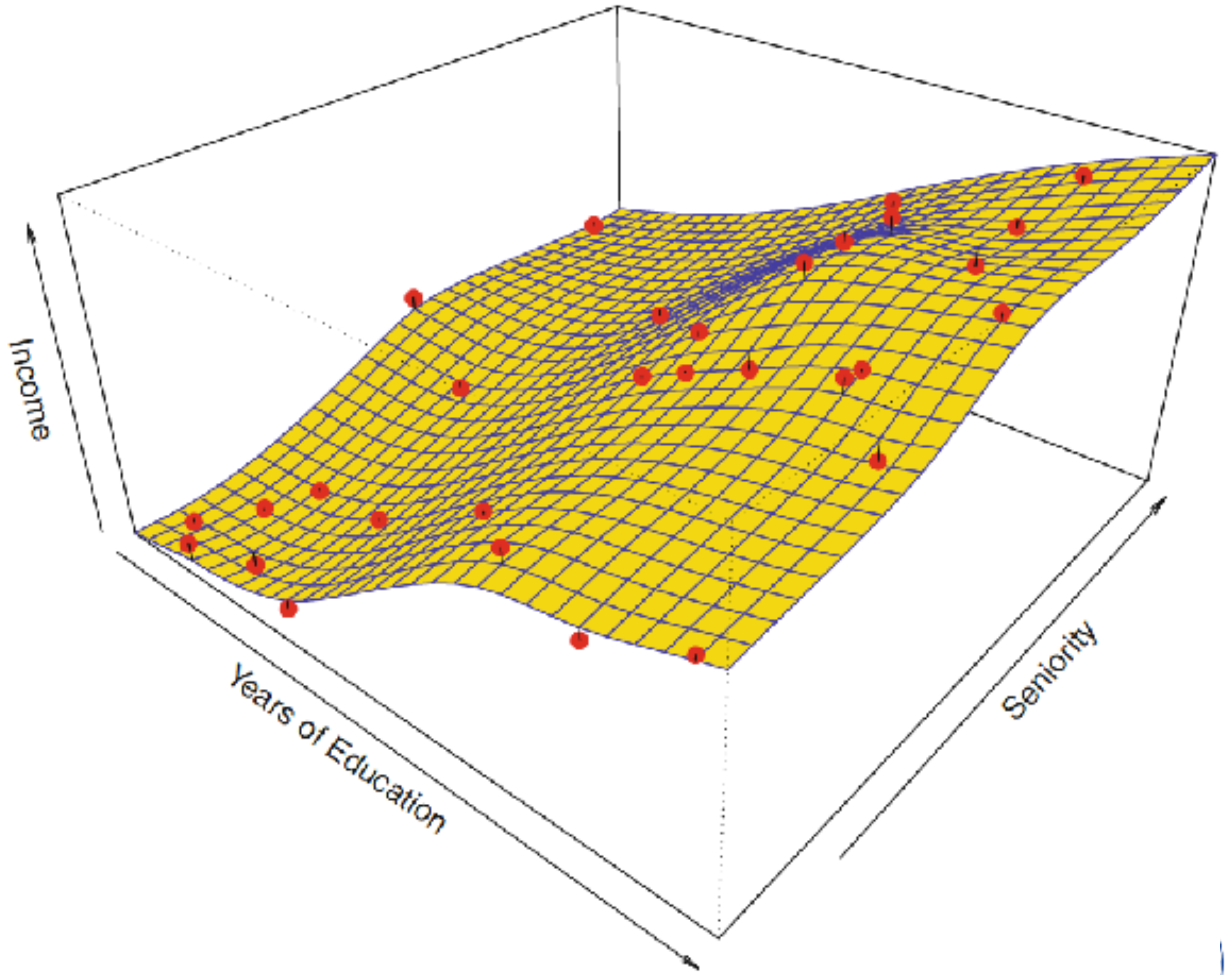


The plot displays income as a function of years of education and seniority in the Income data set. The blue surface represents the true underlying relationship between income and years of education and seniority, which is known since the data are simulated. The red dots indicate the observed values of these quantities for 30 individuals.

A linear model fit by least squares to the Income data

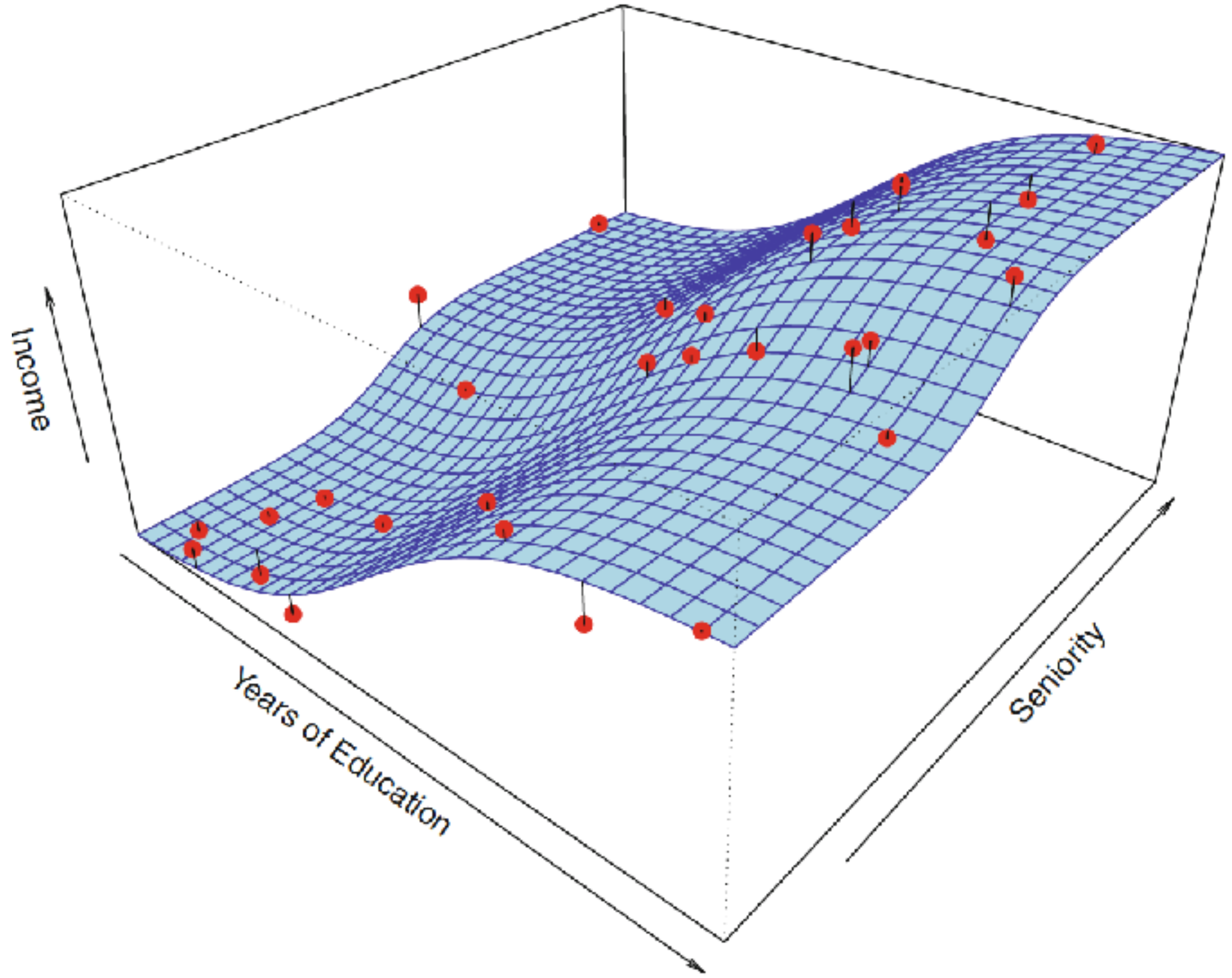


A smooth thin-plate spline fit to the Income data.



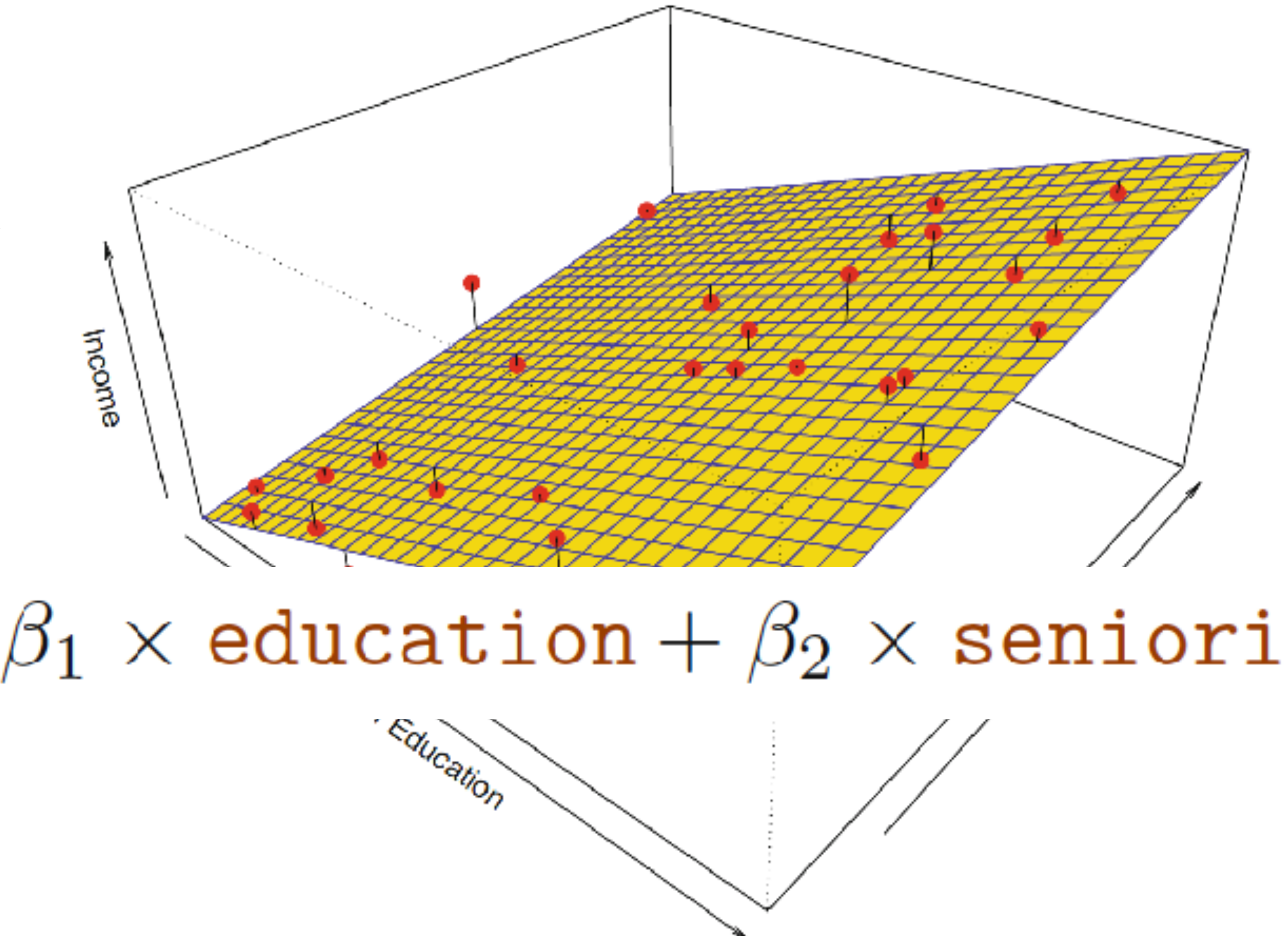
An example of Parametric Vs. Non-parametric methods

The observations are displayed in red; the yellow plane indicates the fitted model;



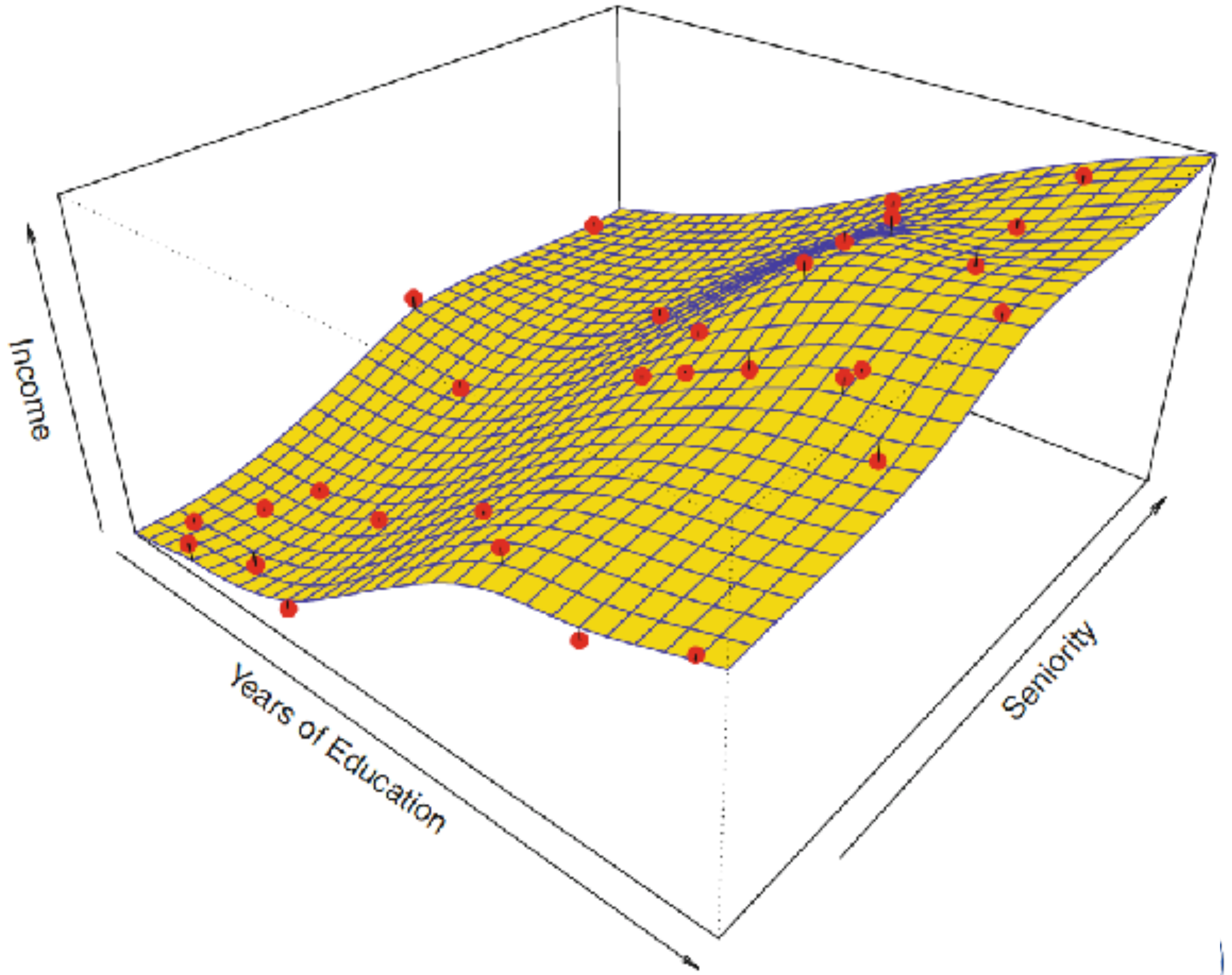
The plot displays income as a function of years of education and seniority in the Income data set. The blue surface represents the true underlying relationship between income and years of education and seniority, which is known since the data are simulated. The red dots indicate the observed values of these quantities for 30 individuals.

A linear model fit by least squares to the Income data



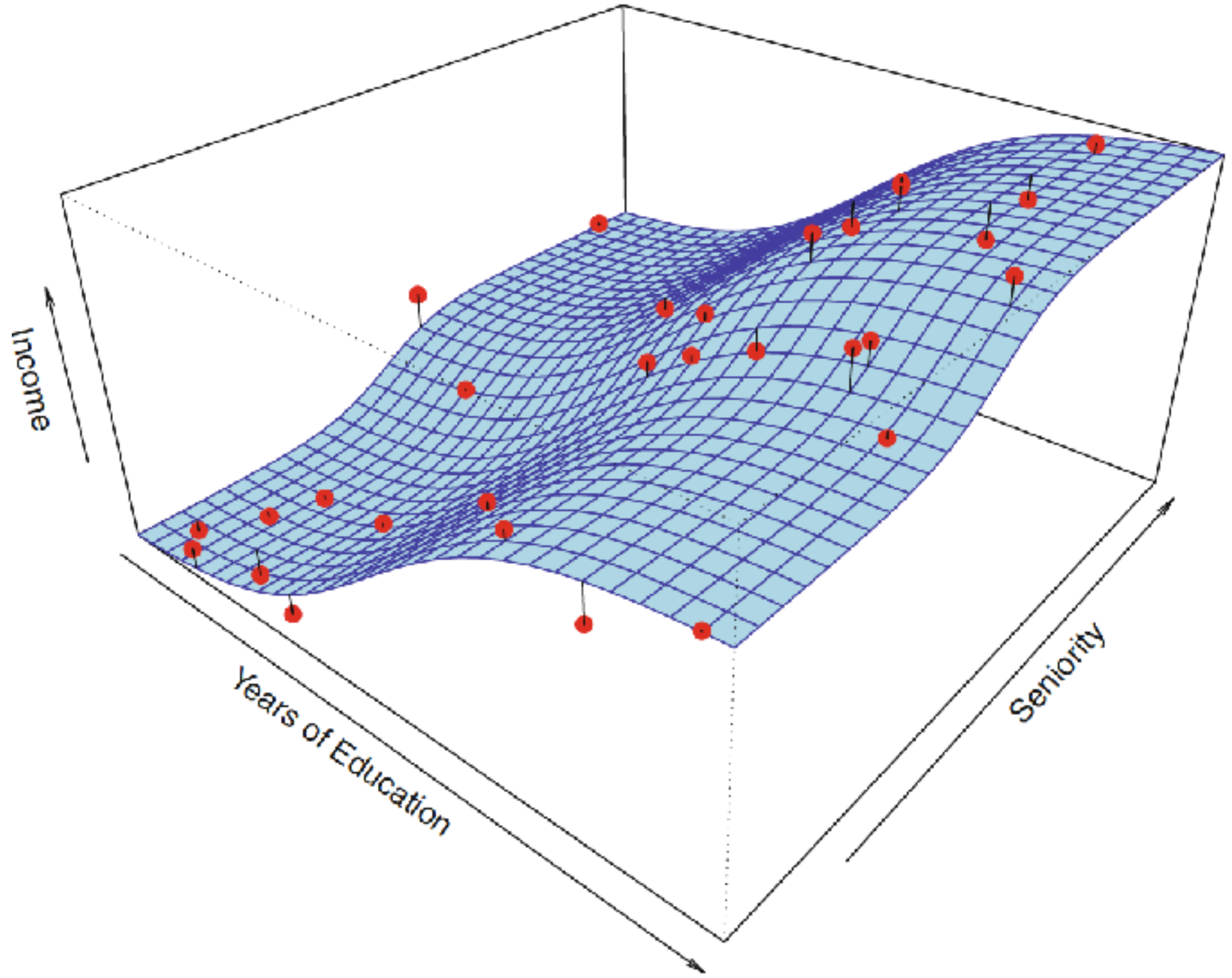
$$\texttt{income} \approx \beta_0 + \beta_1 \times \texttt{education} + \beta_2 \times \texttt{seniority}.$$

A smooth thin-plate spline fit to the Income data.

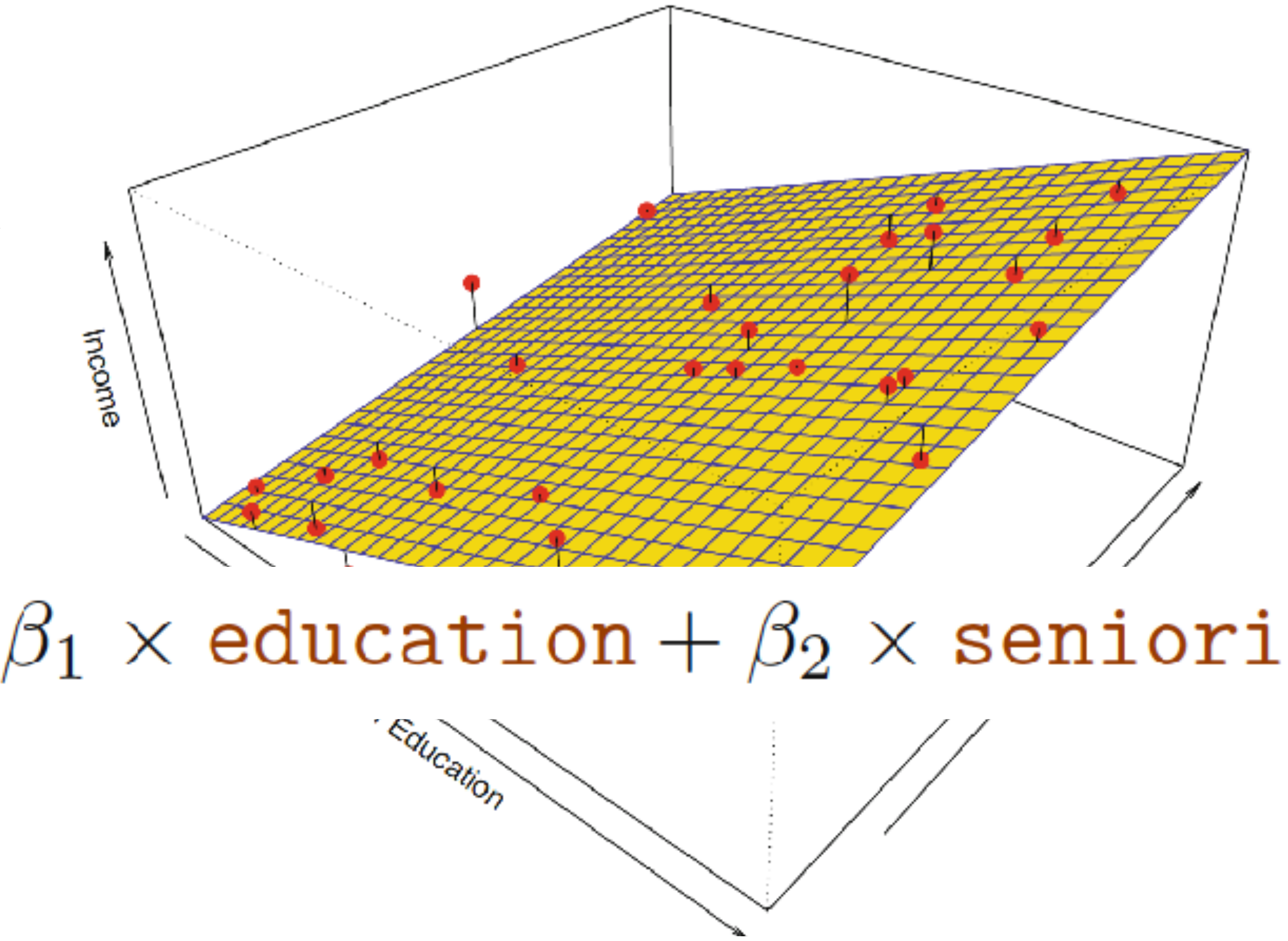


An example of Parametric Vs. Non-parametric methods

The observations are displayed in red; the yellow plane indicates the fitted model;



A linear model fit by least squares to the Income data

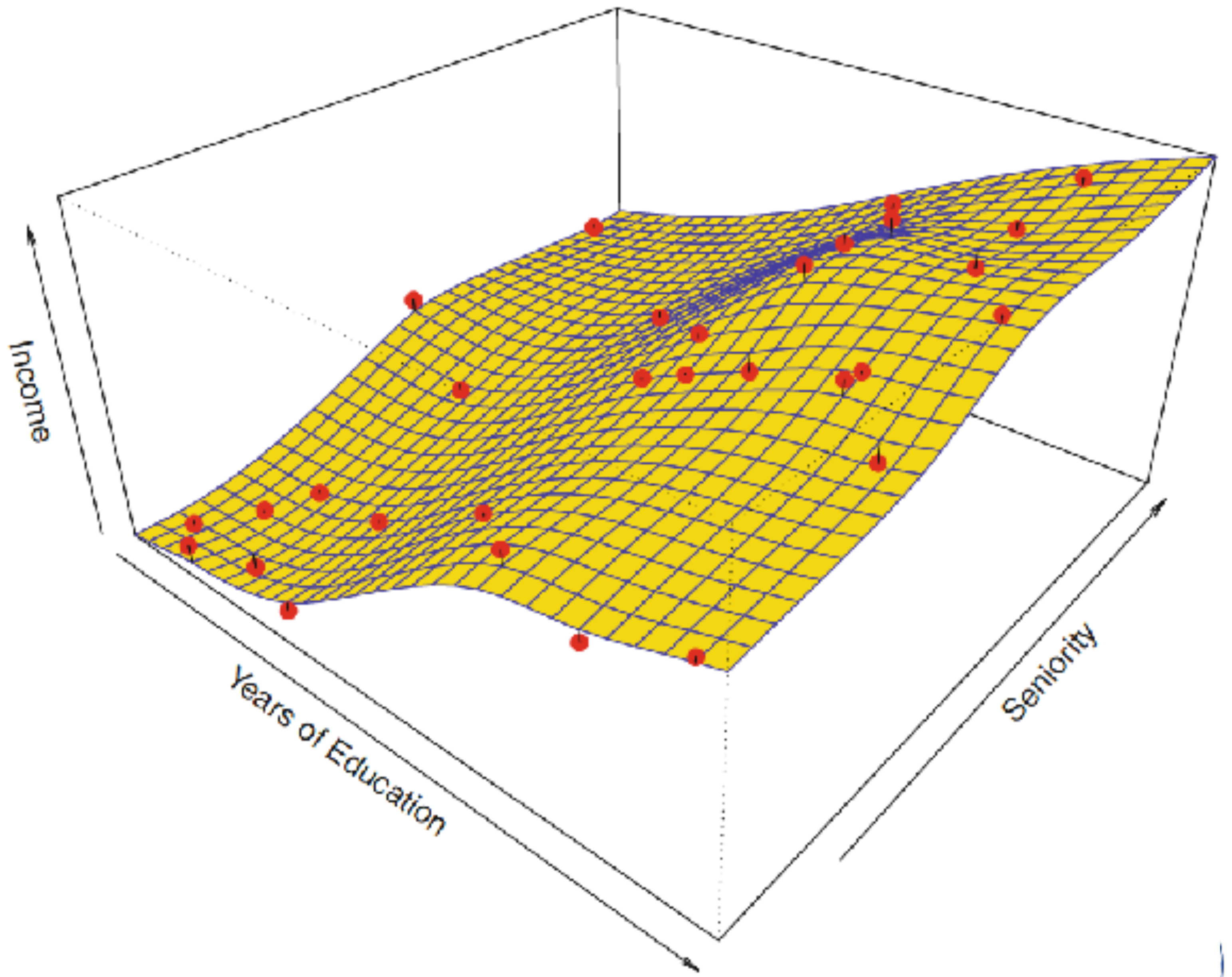


$$\texttt{income} \approx \beta_0 + \beta_1 \times \texttt{education} + \beta_2 \times \texttt{seniority}.$$

The plot displays income as a function of years of education and seniority in the Income data set. The blue surface represents the true underlying relationship between income and years of education and seniority, which is known since the data are simulated. The red dots indicate the observed values of these quantities for 30 individuals.

A smooth thin-plate spline fit to the Income data.

薄板样条函数



Parametric Method Vs. Non-parametric Methods

	Advantages	Disadvantages
Parametric method	- Reducing the *hard* problem down to estimating a set of parameters (*easy*); - Low variance;	- the model we choose will usually not match the true unknown form of f. - These more complex models can lead to a phenomenon known as overfitting the data, which means they follow the errors, or noise, too closely.
Non-Parametric method	- Avoiding the assumption of a particular functional form for f.	- they do not reduce the problem of estimating f to a small number of parameters, a very large number of observations (far more than is typically needed for a parametric approach) is required in order to obtain an accurate estimate for f.

Why is it necessary to introduce so many different statistical learning approaches, rather than just a single best method? *There is no free lunch in statistics*: no one method dominates all others over all possible data sets. On a particular data set, one specific method may work best, but some other method may work better on a similar but different data set.

Chap 2 - Linear Regression(1)

- Simple Linear Regression;
- Key concepts of Statistics in Linear Regression;

Simple Linear Regression

Parametric method

$$Y \approx \beta_0 + \beta_1 X.$$

Simple Linear Regression

Parametric method

- ***Simple Linear Regression***: *Y* is is **quantitative** (e.g price, blood pressure);on the basis of a single predictor variable *X.*

$$Y \approx \beta_0 + \beta_1 X.$$

Symbols explanations:

- You might read "≈" as "*is approximately modeled as*";
- β_0 and β_1 are two unknown constants that represent the intercept and slope terms;
- saying that we are regressing *Y* on *X* (or *Y* onto *X*).
- hat symbol, ˆ , to denote the estimated value for an unknown parameter or coefficient, or to denote the predicted value of the response.

So how to estimate the Coefficients?

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for *Y* based on the *i*-th value of *X*.

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for *Y* based on the *i*-th value of *X*.

$e_i = y_i - \hat{y}_i$ represents the *i*-th residual —this is the difference between the *i*-th observed response value and the *i*-th response value that is predicted by our linear model.

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for *Y* based on the *i*-th value of *X*.

$e_i = y_i - \hat{y}_i$ represents the *i*-th residual —this is the difference between the *i*-th observed response value and the *i*-th response value that is predicted by our linear model.

Residual sum of squares: $\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for *Y* based on the *i*-th value of *X*.

$e_i = y_i - \hat{y}_i$ represents the *i*-th residual —this is the difference between the *i*-th observed response value and the *i*-th response value that is predicted by our linear model.

Residual sum of squares: $\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$

Least squares coefficient estimators:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

How to compute the minimizer?

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for *Y* based on the *i*-th value of *X*.

$e_i = y_i - \hat{y}_i$ represents the *i*-th residual —this is the difference between the *i*-th observed response value and the *i*-th response value that is predicted by our linear model.

Residual sum of squares: $\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$

Least squares coefficient estimators:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

How to compute the minimizer?

$$\bar{y} \equiv \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \bar{x} \equiv \frac{1}{n}\sum_{i=1}^{n} x_i$$

Estimating the Coefficients of Simple Linear Regression

Simple Linear Regression

$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ be the prediction for *Y* based on the *i*-th value of *X*.

$e_i = y_i - \hat{y}_i$ represents the *i*-th residual —this is the difference between the *i*-th observed response value and the *i*-th response value that is predicted by our linear model.

Residual sum of squares: $\mathrm{RSS} = e_1^2 + e_2^2 + \cdots + e_n^2$

Least squares coefficient estimators:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

How to compute the minimizer?

Homework: prove it.

$$\bar{y} \equiv \frac{1}{n}\sum_{i=1}^n y_i \qquad \bar{x} \equiv \frac{1}{n}\sum_{i=1}^n x_i$$

Assessing the Accuracy of the Coefficient Estimates

Simple Linear Regression

Population regression line $Y = \beta_0 + \beta_1 X + \epsilon.$

β_0 is the intercept term—that is, the expected value of *Y* when *X* = 0,

β_1 is the slope—the average increase in Y associated with a one-unit increase in *X.*

Suppose we annotate μ as the population mean of random variable Y

A reasonable estimate $\hat{\mu} = \bar{y},$ $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

If we use the sample mean $\hat{\mu}$ to estimate μ, this estimate is unbiased.

So how accurate is the estimation?

Standard error of $\hat{\mu}$ $\mathrm{Var}(\hat{\mu}) = \mathrm{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n},$ σ is the standard deviation of each of the realisations y_i for uncorrelated observations.

Assessing the Accuracy of the Coefficient Estimates

Simple Linear Regression

Population regression line $Y = \beta_0 + \beta_1 X + \boxed{\epsilon.}$ mean-zero random error term.

β_0 is the intercept term—that is, the expected value of *Y* when *X* = 0,

β_1 is the slope—the average increase in Y associated with a one-unit increase in *X.*

Suppose we annotate μ as the population mean of random variable Y

A reasonable estimate $\hat{\mu} = \bar{y}$, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

If we use the sample mean $\hat{\mu}$ to estimate μ, this estimate is unbiased.

So how accurate is the estimation?

Standard error of $\hat{\mu}$ $\mathrm{Var}(\hat{\mu}) = \mathrm{SE}(\hat{\mu})^2 = \frac{\sigma^2}{n}$, σ is the standard deviation of each of the realisations y_i for uncorrelated observations.

Standard Error and Confidence Intervals

Simple Linear Regression

Standard Errors $\hat{\beta}_0$ and $\hat{\beta}_1$ $\mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right], \quad \mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$ $\sigma^2 = \mathrm{Var}(\epsilon).$

ϵ_i for each observation are uncorrelated with common variance σ^2

The estimate of σ residual standard error is known as the *residual standard error.*

$$\mathrm{RSE} = \sqrt{\mathrm{RSS}/(n-2)}.$$

For linear regression

Standard Error and Confidence Intervals

Simple Linear Regression

Standard Errors $\hat{\beta}_0$ and $\hat{\beta}_1$ $\mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \right], \quad \mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$ $\sigma^2 = \mathrm{Var}(\epsilon).$

ϵ_i for each observation are uncorrelated with common variance σ^2

The estimate of σ residual standard error is known as the *residual standard error.*

$$\mathrm{RSE} = \sqrt{\mathrm{RSS}/(n-2)}.$$

1, Standard errors can be used to compute *confidence intervals.* A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter:

For linear regression $\hat{\beta}_1 \pm 2 \cdot \mathrm{SE}(\hat{\beta}_1).$

Standard Error and Confidence Intervals

Simple Linear Regression

Standard Errors $\hat{\beta}_0$ and $\hat{\beta}_1$ $\mathrm{SE}(\hat{\beta}_0)^2 = \sigma^2\left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2}\right], \quad \mathrm{SE}(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$ $\sigma^2 = \mathrm{Var}(\epsilon).$

ϵ_i for each observation are uncorrelated with common variance σ^2

The estimate of σ residual standard error is known as the *residual standard error.*

$$\mathrm{RSE} = \sqrt{\mathrm{RSS}/(n-2)}.$$

1, Standard errors can be used to compute *confidence intervals.* A 95% confidence interval is defined as a range of values such that with 95% probability, the range will contain the true unknown value of the parameter:

For linear regression $\hat{\beta}_1 \pm 2 \cdot \mathrm{SE}(\hat{\beta}_1).$

There is approximately a 95% chance that the interval, (assume Gaussian Errors here).

$$\left[\hat{\beta}_1 - 2 \cdot \mathrm{SE}(\hat{\beta}_1),\ \hat{\beta}_1 + 2 \cdot \mathrm{SE}(\hat{\beta}_1)\right]$$

will contain the true value of β_1

Chap 2 - Linear Regression(1)

Linear Regression from Probabilistic Perspective

[1] Chap 3.1, Bishop 2006

Maximum Likelihood and Least Squares (1)

Assume observations from a deterministic function with added Gaussian noise:

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \qquad \text{where} \qquad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

which is the same as saying,

$$p(t|\mathbf{x}, \mathbf{w}, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1}).$$

Given observed inputs, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, and targets, $\mathbf{t} = [t_1, \ldots, t_N]^{\mathrm{T}}$ we obtain the likelihood function

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}).$$

$$\mathbf{w} = (w_0, \ldots, w_{M-1})^{\mathrm{T}} \text{ and } \boldsymbol{\phi} = (\phi_0, \ldots, \phi_{M-1})^{\mathrm{T}}$$

β precision (inverse variance)

Maximum Likelihood and Least Squares (2)

Taking the logarithm, we get

$$\begin{aligned} \ln p(\mathbf{t}|\mathbf{w}, \beta) &= \sum_{n=1}^{N} \ln \mathcal{N}(t_n|\mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n), \beta^{-1}) \\ &= \frac{N}{2}\ln\beta - \frac{N}{2}\ln(2\pi) - \beta E_D(\mathbf{w}) \end{aligned}$$

where

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2$$

is the sum-of-squares error.

Maximum Likelihood and Least Squares (3)

Optional subtitle

Computing the gradient and setting it to zero yields

$$\nabla_{\mathbf{w}} \ln p(\mathbf{t}|\mathbf{w}, \beta) = \beta \sum_{n=1}^{N} \left\{ t_n - \mathbf{w}^{\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n) \right\} \boldsymbol{\phi}(\mathbf{x}_n)^{\mathrm{T}} = \mathbf{0}.$$

Solving for w, we get

$$\mathbf{w}_{\mathrm{ML}} = \overbrace{\left(\boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^{\mathrm{T}}}^{\boxed{\text{The Moore-Penrose pseudo-inverse,} \boldsymbol{\Phi}^{\dagger}}} \mathbf{t}$$

where

$$\boldsymbol{\Phi} = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix}.$$

Geometry of Least Squares

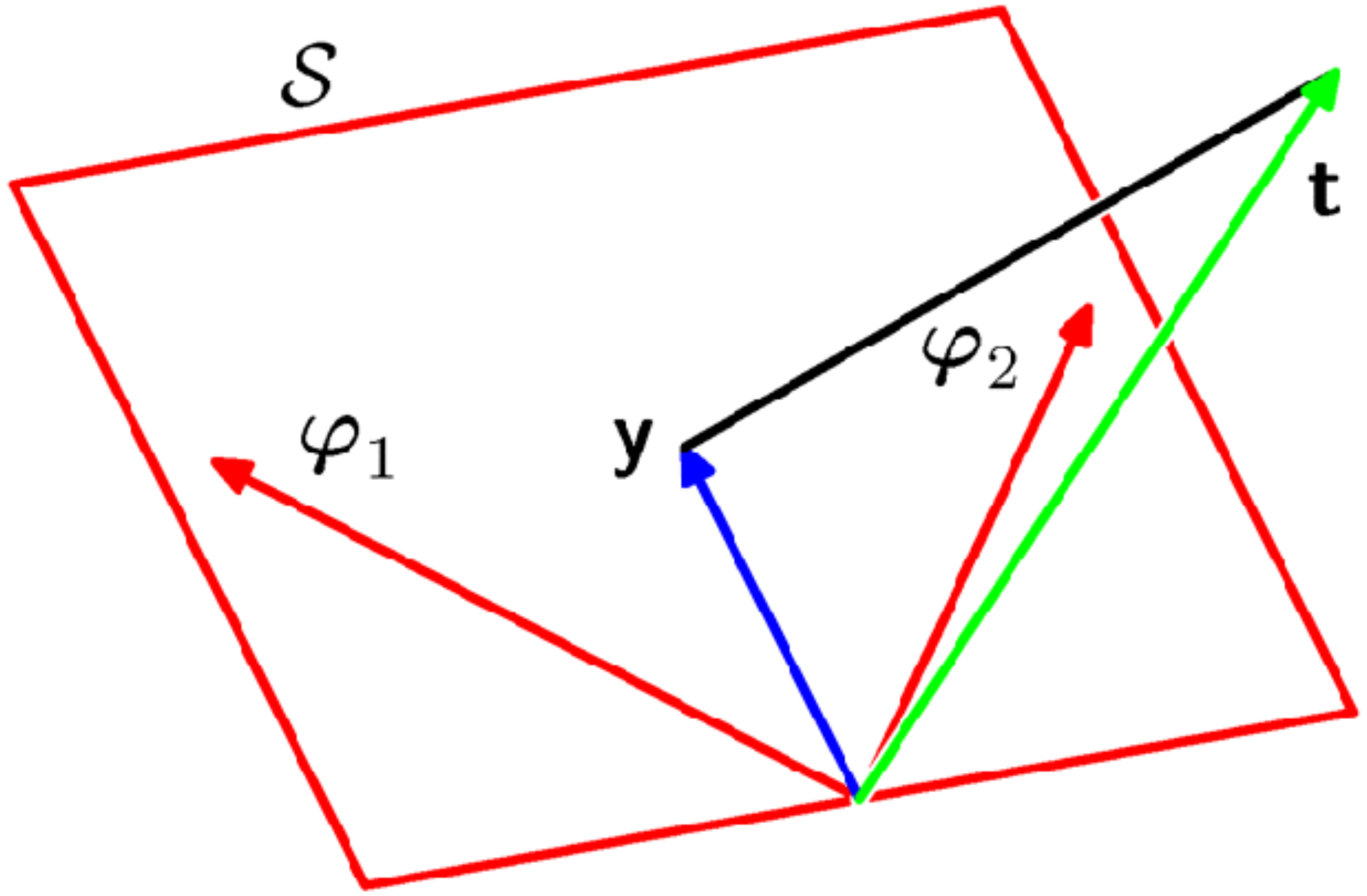
Consider $\mathbf{y} = \mathbf{\Phi}\mathbf{w}_{\mathrm{ML}} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M]\,\mathbf{w}_{\mathrm{ML}}$.

$\mathbf{y} \in \mathcal{S} \subseteq \mathcal{T}$ $\quad\quad$ $\mathbf{t} \in \mathcal{T}$

N-dimensional

M-dimensional

S is spanned by $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M$.



w_{ML} minimizes the distance between t and its orthogonal projection on S, i.e. y.

Sequential Learning

Big Data Problem? Lots of training data. Hard to load them all together.

Data items considered one at a time (a.k.a. online learning); use stochastic (sequential) gradient descent:

$$\begin{aligned}\mathbf{w}^{(\tau+1)} &= \mathbf{w}^{(\tau)} - \eta \nabla E_n \\ &= \mathbf{w}^{(\tau)} + \eta(t_n - \mathbf{w}^{(\tau)\mathrm{T}} \boldsymbol{\phi}(\mathbf{x}_n))\boldsymbol{\phi}(\mathbf{x}_n).\end{aligned}$$

This is known as the *least-mean-squares (LMS) algorithm*.

Regularized Least Squares (1)

Consider the error function:

$$E_D(\mathbf{w}) + \lambda E_W(\mathbf{w})$$

Data term + Regularization term

With the sum-of-squares error function and a quadratic regularizer, we get

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w}$$

Homework: prove it

λ is called the regularization coefficient.

which is minimized by

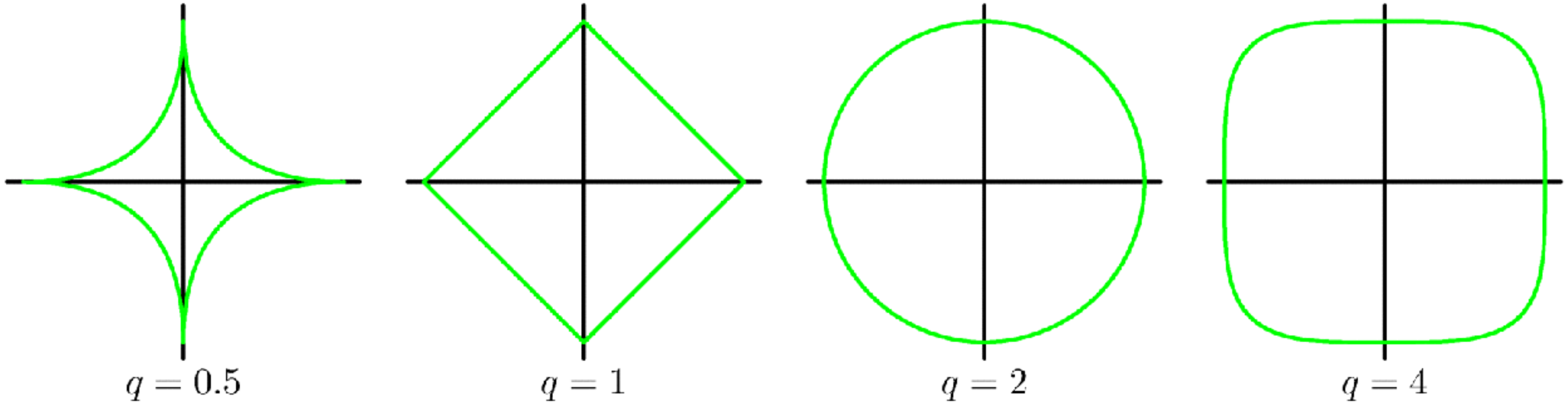
$$\mathbf{w} = \left(\lambda\mathbf{I} + \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{t}.$$

Regularized Least Squares (2)

Optional subtitle

With a more general regularizer, we have

$$\frac{1}{2}\sum_{n=1}^{N}\{t_n - \mathbf{w}^{\mathrm{T}}\boldsymbol{\phi}(\mathbf{x}_n)\}^2 + \frac{\lambda}{2}\sum_{j=1}^{M}|w_j|^q$$

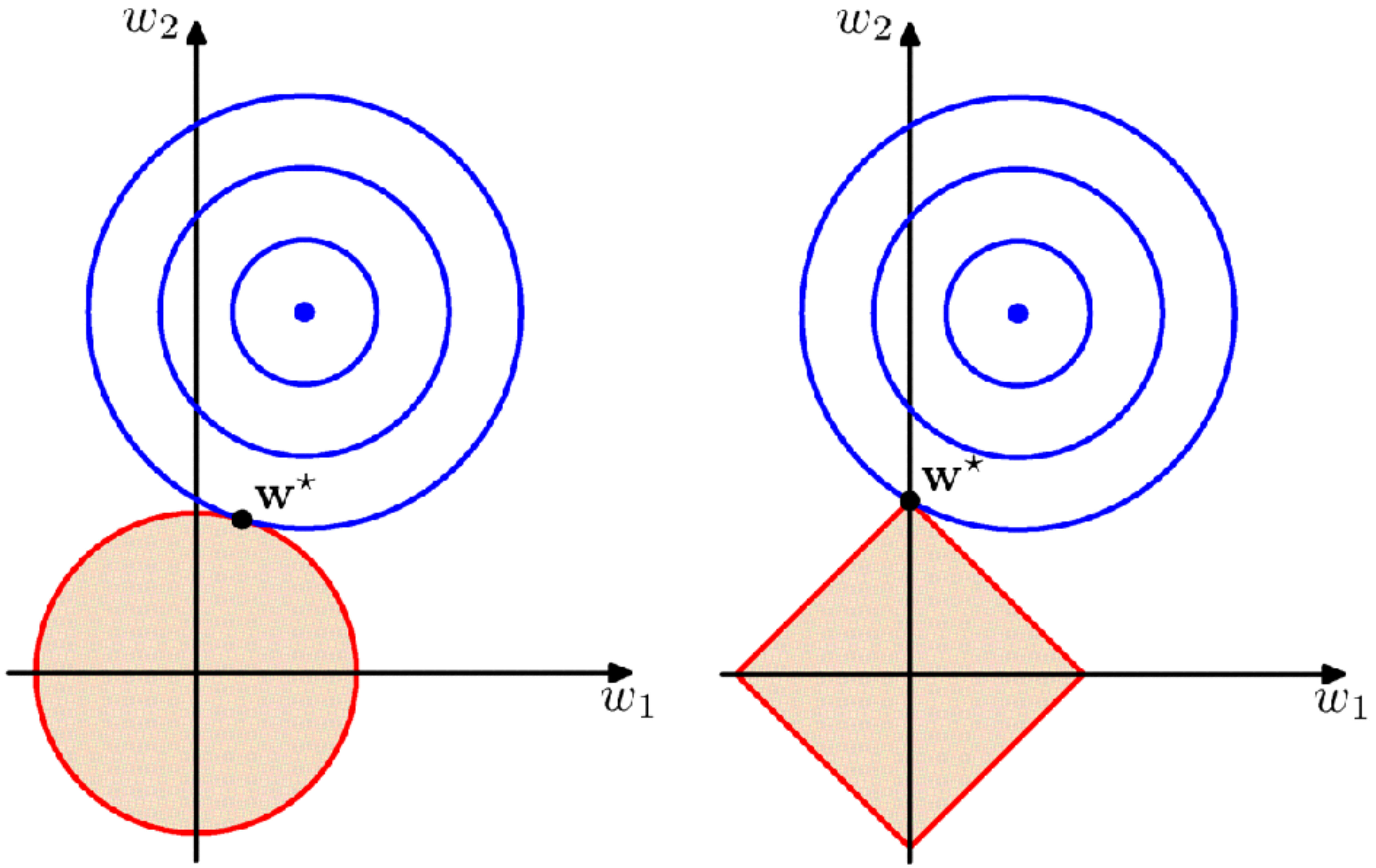


Lasso

Quadratic

Regularized Least Squares (3)

Lasso tends to generate sparser solutions than a quadratic regularizer.



The Bias-Variance Decomposition (1)

Optional subtitle

Recall the *expected squared loss*,

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} + \underbrace{\iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t}_{}$$

where

optimal prediction is given by the conditional expectation

$$h(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}] = \int t p(t|\mathbf{x})\,\mathrm{d}t.$$

The second term of E[L] corresponds to the noise inherent in the random variable t.

What about the first term?

The Bias-Variance Decomposition (2)

Optional subtitle

Suppose we were given multiple data sets, each of size N. Any particular data set, D, will give a particular function y(x;D). We then have

$$
\begin{aligned}
&\{y(\mathbf{x};\mathcal{D}) - h(\mathbf{x})\}^2 \\
&\quad = \quad \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] + \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\quad = \quad \{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2 + \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 \\
&\qquad\qquad +2\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}.
\end{aligned}
$$

The Bias-Variance Decomposition (3)

Optional subtitle

Taking the expectation over D yields

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-h(\mathbf{x})\}^2\right]\\&\quad=\underbrace{\{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]-h(\mathbf{x})\}^2}_{(\text{bias})^2}+\underbrace{\mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D})-\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right]}_{\text{variance}}.\end{aligned}$$

The Bias-Variance Decomposition (4)

Optional subtitle

Thus we can write

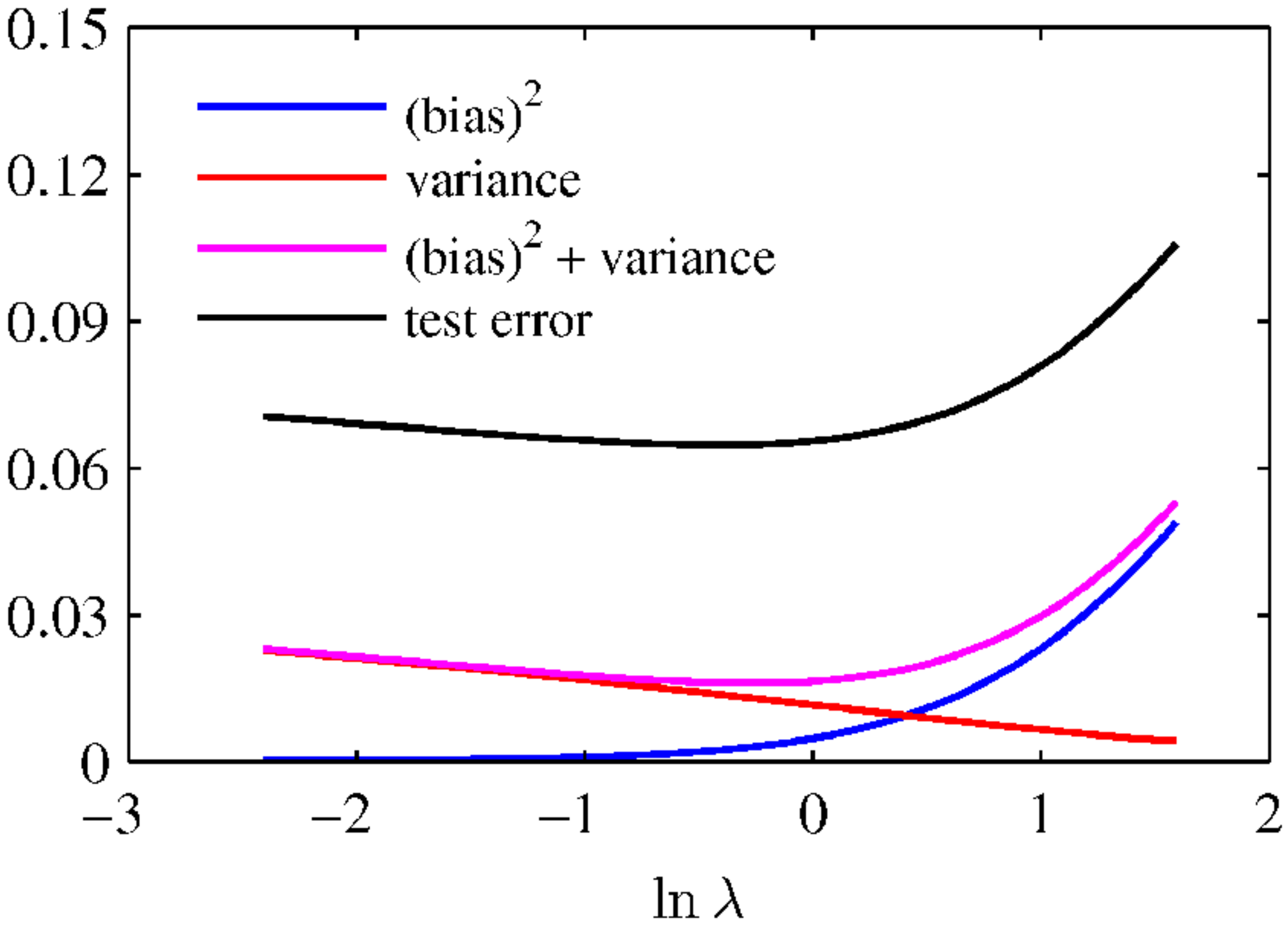
$$\text{expected loss} = (\text{bias})^2 + \text{variance} + \text{noise}$$

where

$$\begin{aligned}
(\text{bias})^2 &= \int \{\mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})] - h(\mathbf{x})\}^2 p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{variance} &= \int \mathbb{E}_{\mathcal{D}}\left[\{y(\mathbf{x};\mathcal{D}) - \mathbb{E}_{\mathcal{D}}[y(\mathbf{x};\mathcal{D})]\}^2\right] p(\mathbf{x})\,\mathrm{d}\mathbf{x} \\
\text{noise} &= \iint \{h(\mathbf{x}) - t\}^2 p(\mathbf{x},t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t
\end{aligned}$$

The Bias-Variance Trade-off

From these plots, we note that an over-regularized model (large ¸) will have a high bias, while an under-regularized model (small ¸) will have a high variance.

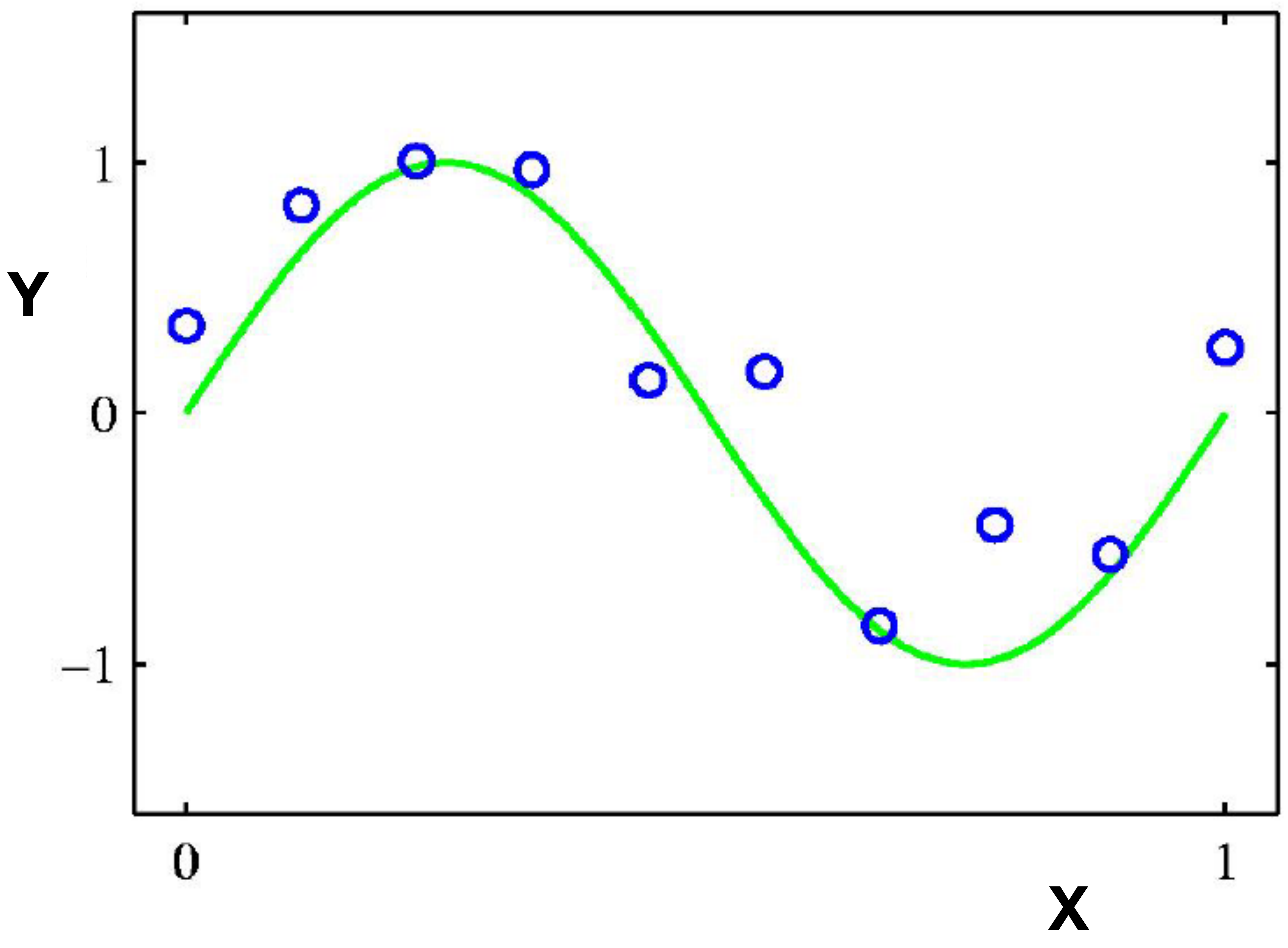


Chap 2 - Linear Regression(2)

Recap&Multiple Linear Regression

- Multiple Linear Regression Sec 3.2 [James, 2013]

Simple Linear Regression



Circles are data points (i.e., training examples) Given
In green is the "true" curve that we don't know

Goal : We want to fit a curve to these points.

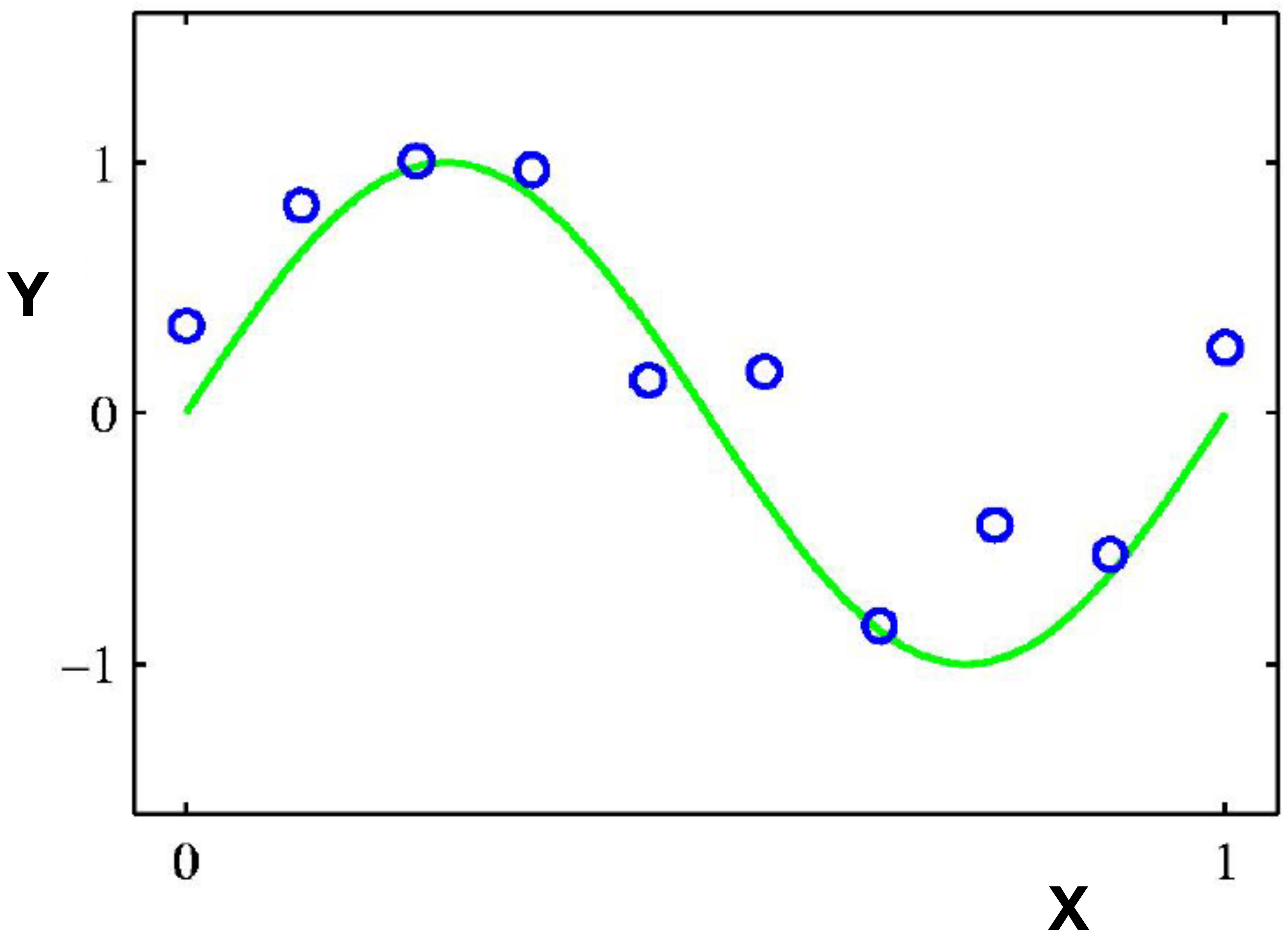
Key Questions:
(1)How do we parametrize the model ?
(2)What loss (objective) function should we use to judge the fit?
(3)How do we optimize fit to unseen test data (generalization)?

Training Set: $\mathcal{D}=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$

$$Y \approx \beta_0 + \beta_1 X.$$

$$Y = \beta_0 + \beta_1 X + \epsilon.$$

Simple Linear Regression



Circles are data points (i.e., training examples) Given
In green is the "true" curve that we don't know

Goal : We want to fit a curve to these points.

Key Questions:
(1)How do we parametrize the model ?
(2)What loss (objective) function should we use to judge the fit?
(3)How do we optimize fit to unseen test data (generalization)?

Training Set: $\mathcal{D}=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$

$$Y \approx \beta_0 + \beta_1 X.$$

$$Y = \beta_0 + \beta_1 X + \boxed{\epsilon.}$$

mean-zero
random error term.

Noise

A simple model typically does not exactly fit the data — lack of fit can be considered noise. Sources of noise:

-> Imprecision in data attributes (input noise)

-> Errors in data targets (mis-labeling)

-> Additional attributes not taken into account by data attributes, affect target values (latent variables)

-> Model may be too simple to account for data targets.

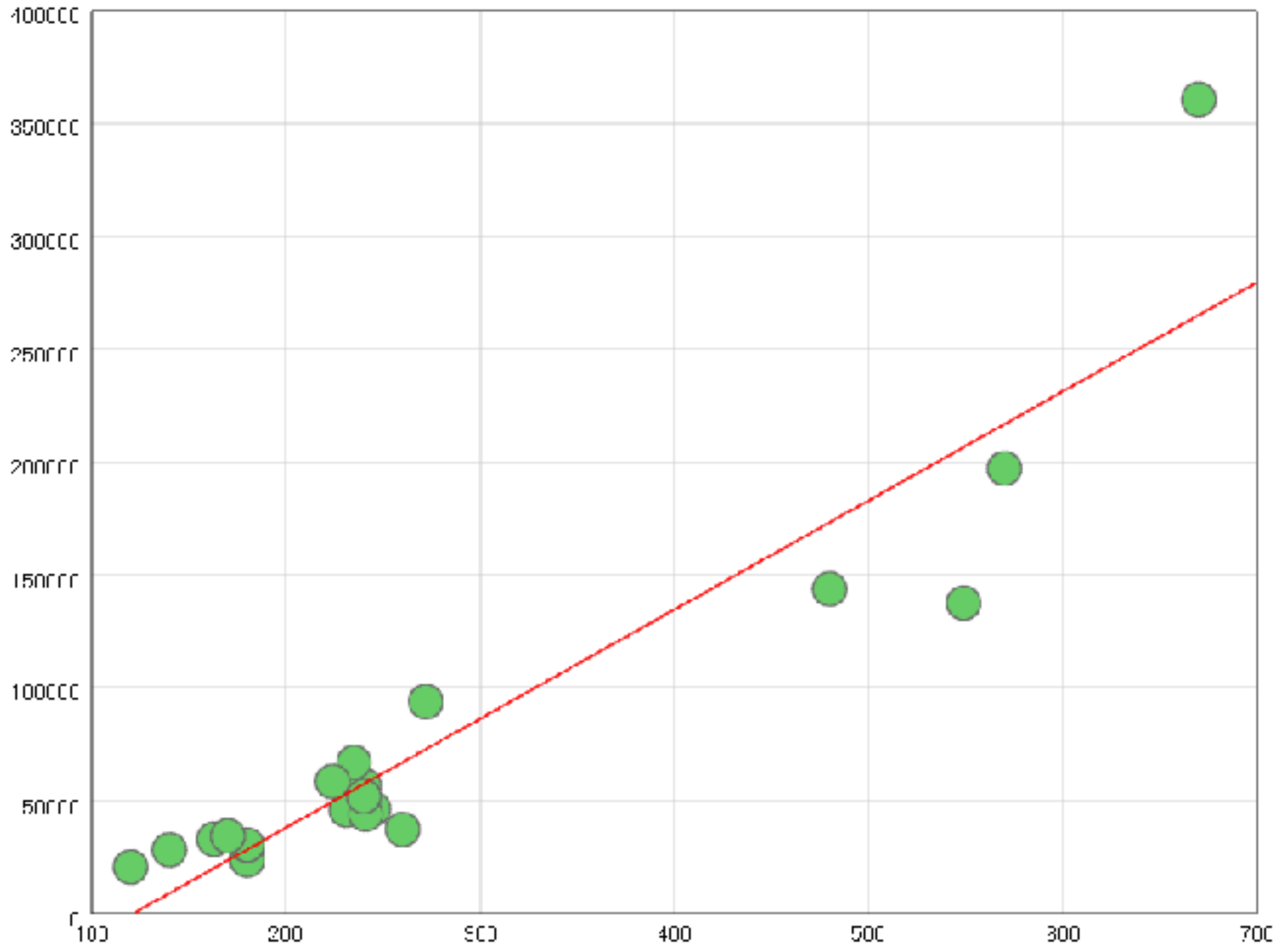
Optimizing the Objective (1)

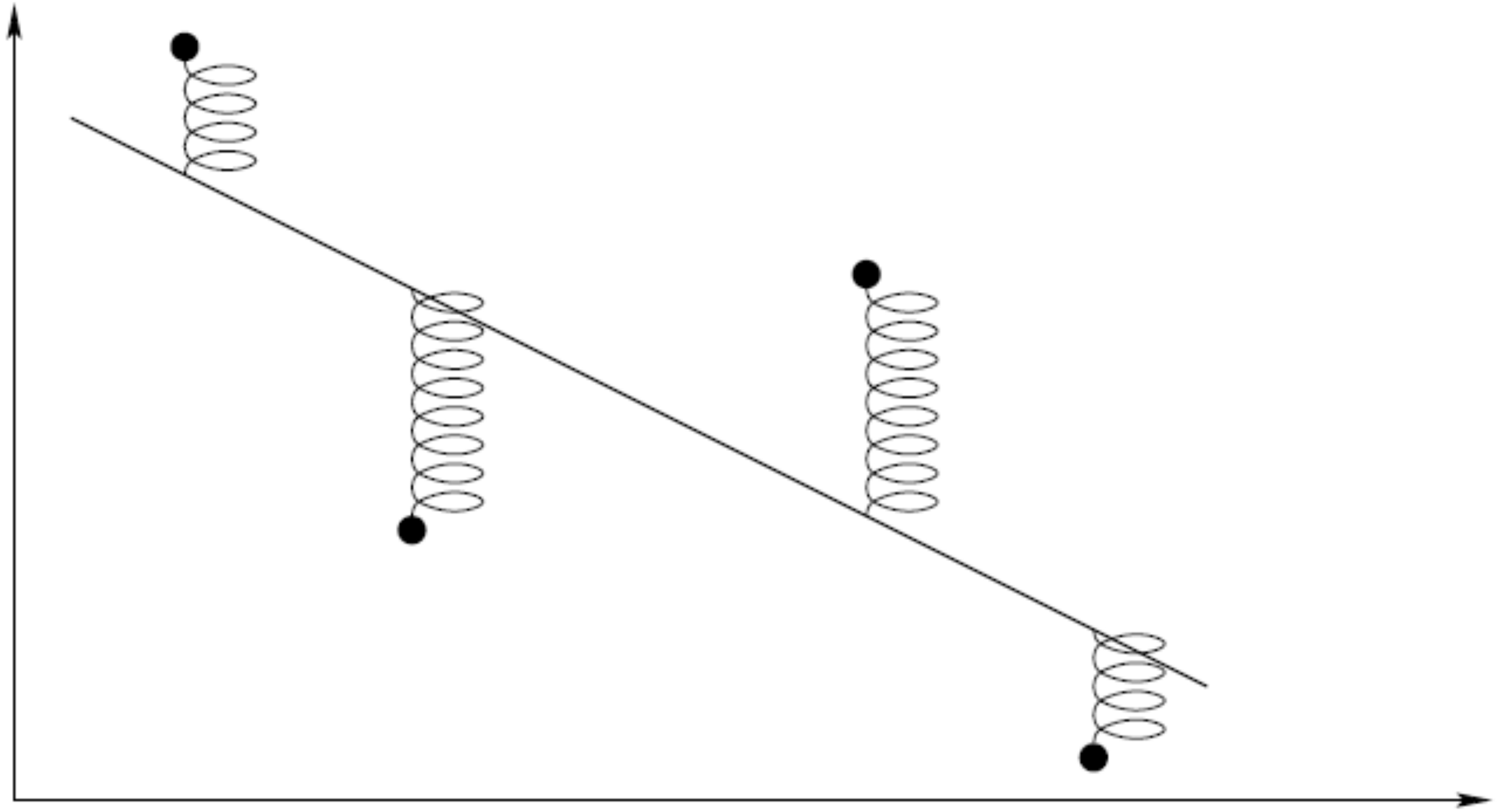
$$Y = \beta_0 + \beta_1 X + \epsilon.$$

Standard loss/cost/objective function measures the squared error between Y and $\hat{Y}$

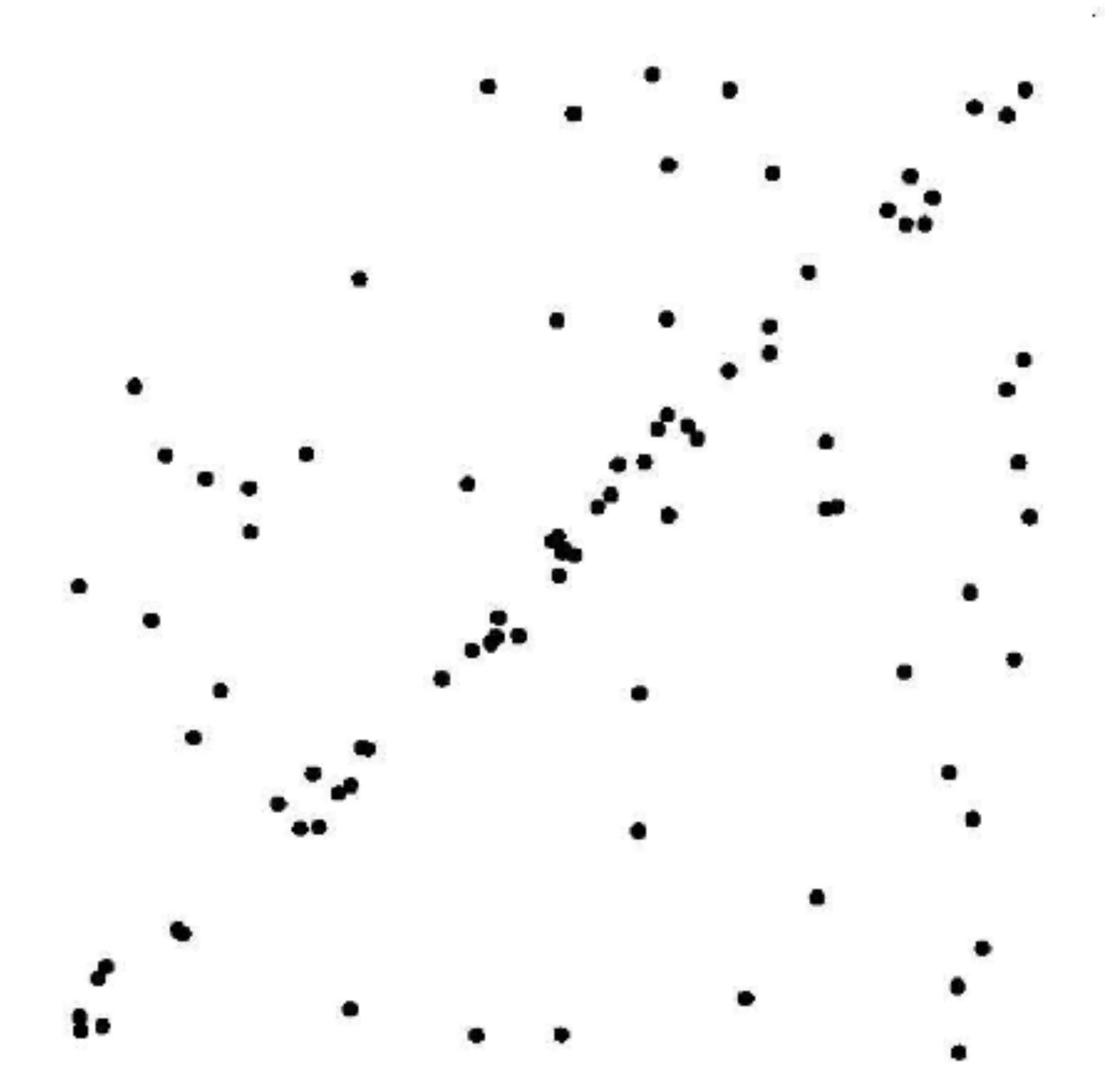
$$l\left(y,\hat{y}\right) = \sum_{i=1}^{N}\left[y_i - \left(\beta_0 + \beta_1 x_i\right)\right]^2$$

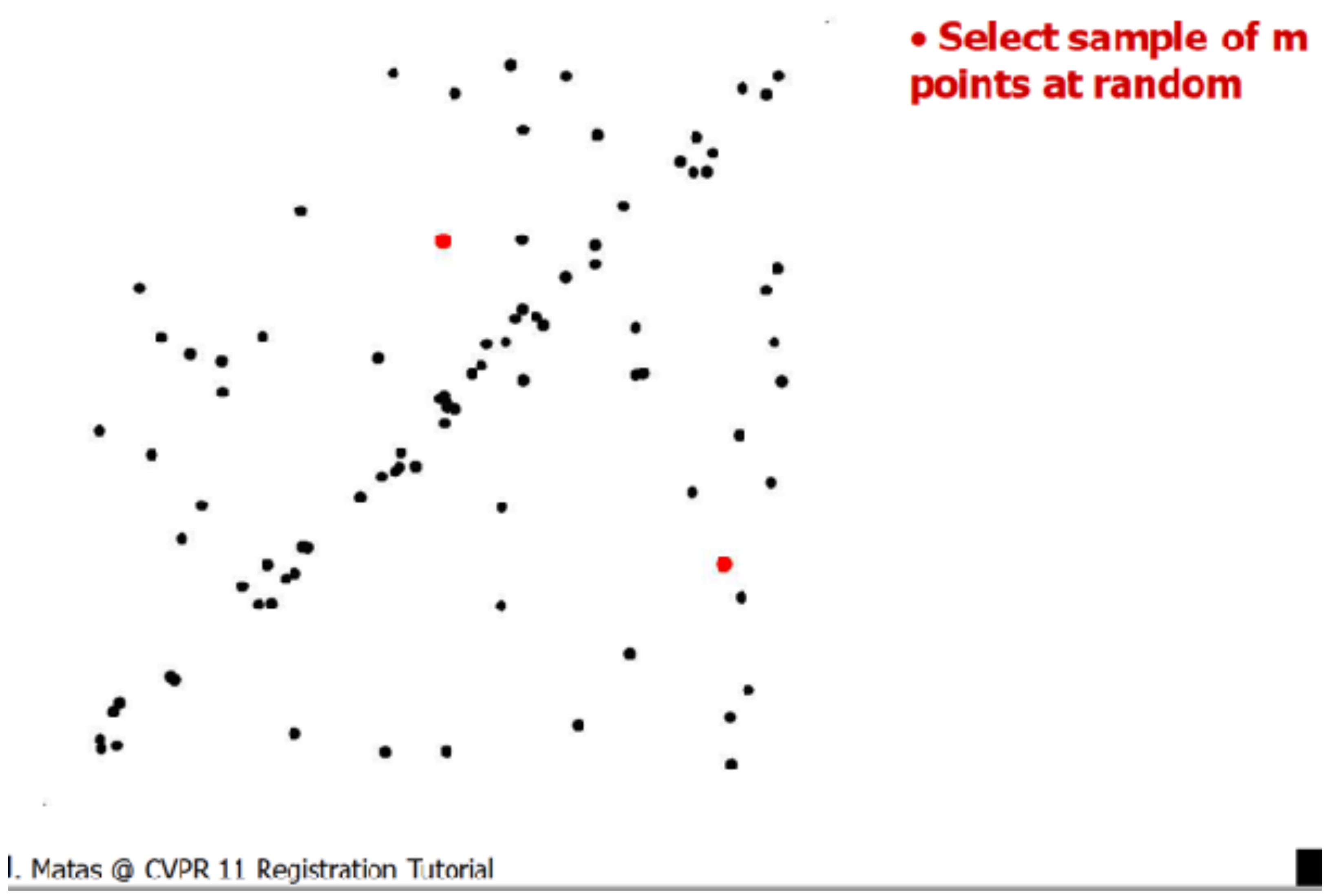
How do we obtain the parameters in general?





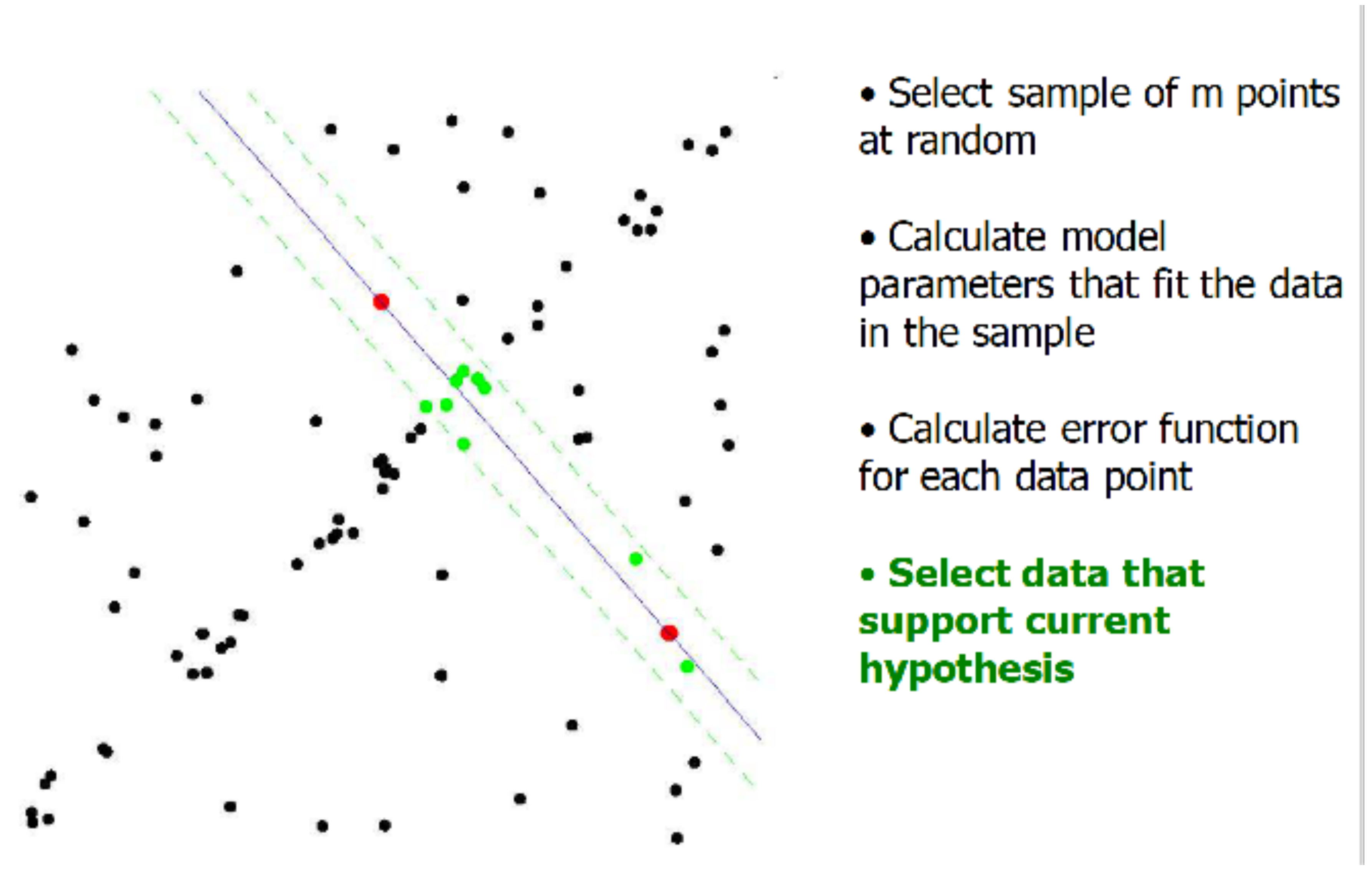
RANSAC



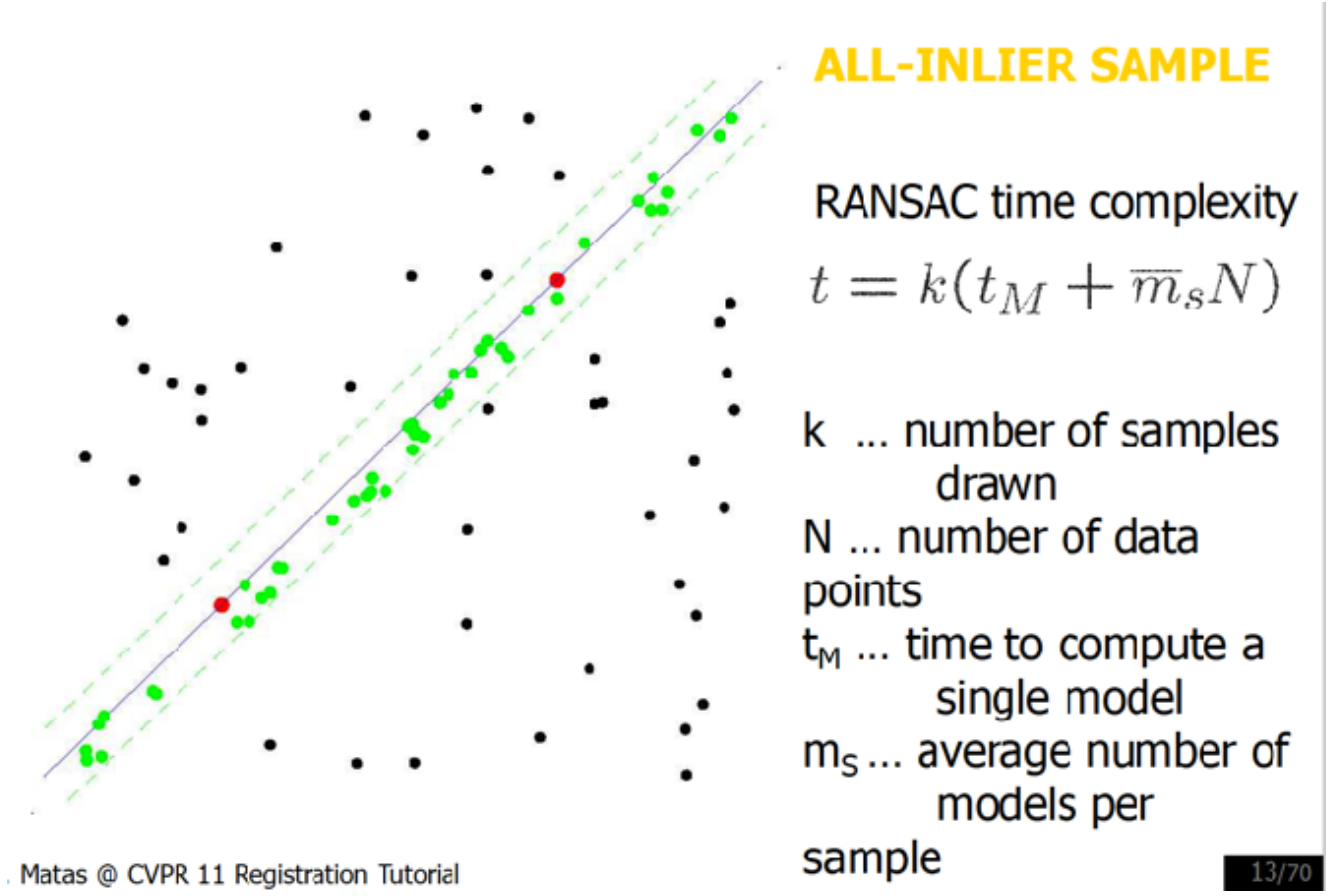


- **Select sample of m points at random**

J. Matas @ CVPR 11 Registration Tutorial



- Select sample of m points at random
- Calculate model parameters that fit the data in the sample
- Calculate error function for each data point
- **Select data that support current hypothesis**



ALL-INLIER SAMPLE

RANSAC time complexity

$$t = k(t_M + \overline{m}_s N)$$

k … number of samples drawn

N … number of data points

t_M … time to compute a single model

m_S … average number of models per sample

Matas @ CVPR 11 Registration Tutorial

13/70

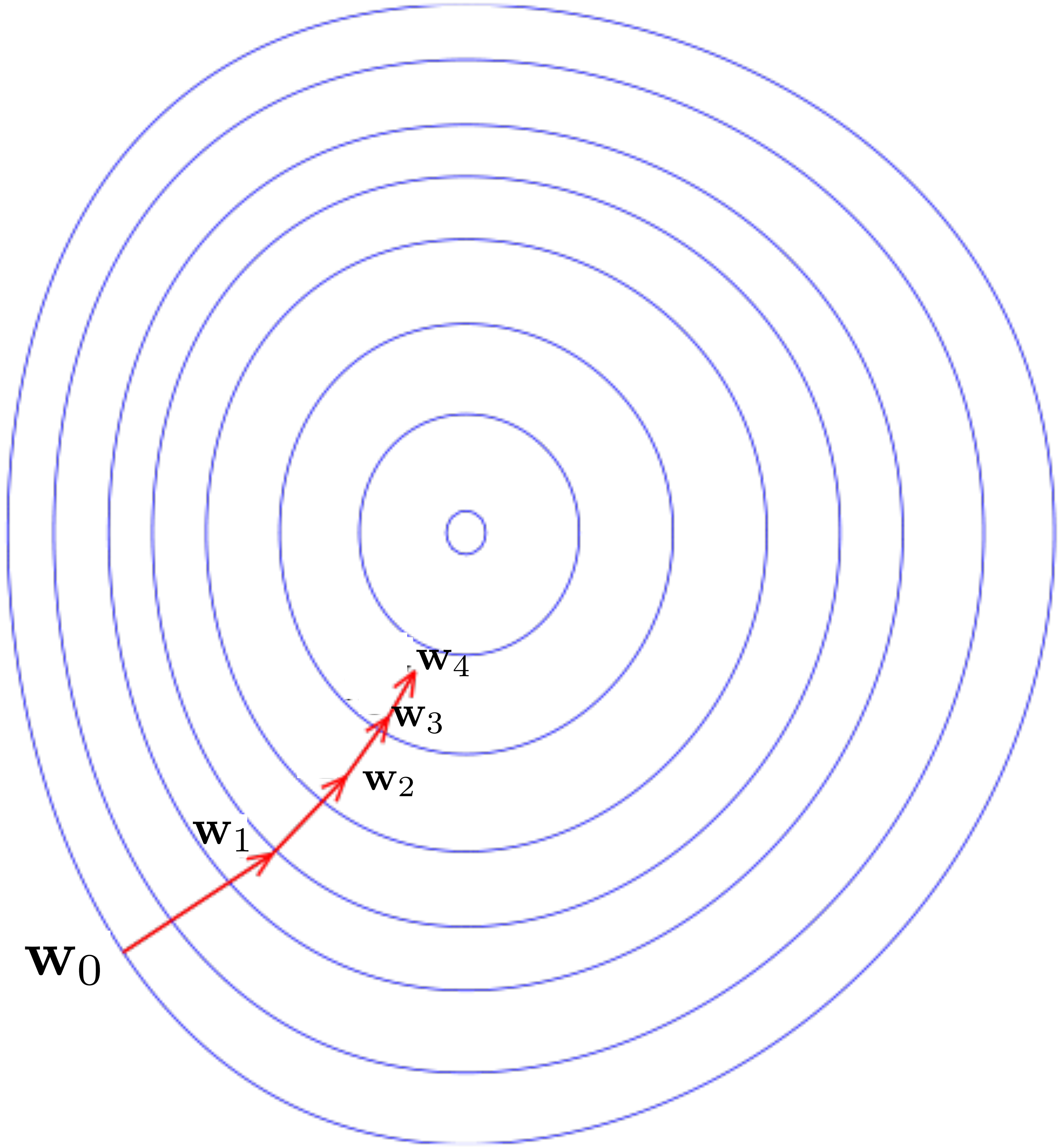
Optimizing the Objective (2)

1, Closed form solution

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

2, Most straightforward solution: gradient descent



(1) initialize **w** (e.g., randomly)
(2) repeatedly update **w** by gradient

$$\mathbf{w} = [\beta_0, \beta_1]^T$$

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda \frac{\partial l}{\partial \mathbf{w}}$$

λ is the learning rate

3, Two ways to generalize this for all examples in training set:

(1) Batch updates : sum or average updates across every example n , then change the parameter values

(2) Stochastic/online updates: update the parameters for each training case in turn, according to its own gradients

Insight of Linear Model

Polynomial Regression $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$

Bias-Variance Decomposition

assume: $y_i = f\left(x_i\right) + \epsilon_i$ for some function f and assume we have a “leaner” that make a training set $\mathcal{D}$

$\epsilon_i \sim N\left(0, \theta^2\right)$

Insight of Linear Model

Polynomial Regression $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ldots + \beta_d x_i^d + \epsilon_i$

Bias-Variance Decomposition

assume: $y_i = f(x_i) + \epsilon_i$ for some function f and assume we have a "leaner" that make a training set $\mathcal{D}$

$$\epsilon_i \sim N\left(0, \theta^2\right)$$

Then for a new example (x_i, y_i) the error averaged over training sets is

$$E[(y_i - \hat{f}(x_i))^2] = Bias[\hat{f}(x_i)]^2 + Var[\hat{f}(x_i)] + \theta^2$$

"Irreducible error": best we can hope for given the noise level.

Expected error due to having wrong model.

Where $Bias[\hat{f}(x_i)] = E[\hat{f}(x_i)] - f(x_i)$,

$$Var[\hat{f}(x_i)] = E[(\hat{f}(x_i) - E[\hat{f}(x_i)])^2]$$

How sensitive is the model to the particular training set?

Supervised Learning Pipeline (Prepare for the Projects)

1, Given a training set X and y, with i.i.d assumption (training and test data drawn from same distribution), if we have an explicit test set to approximate test error:

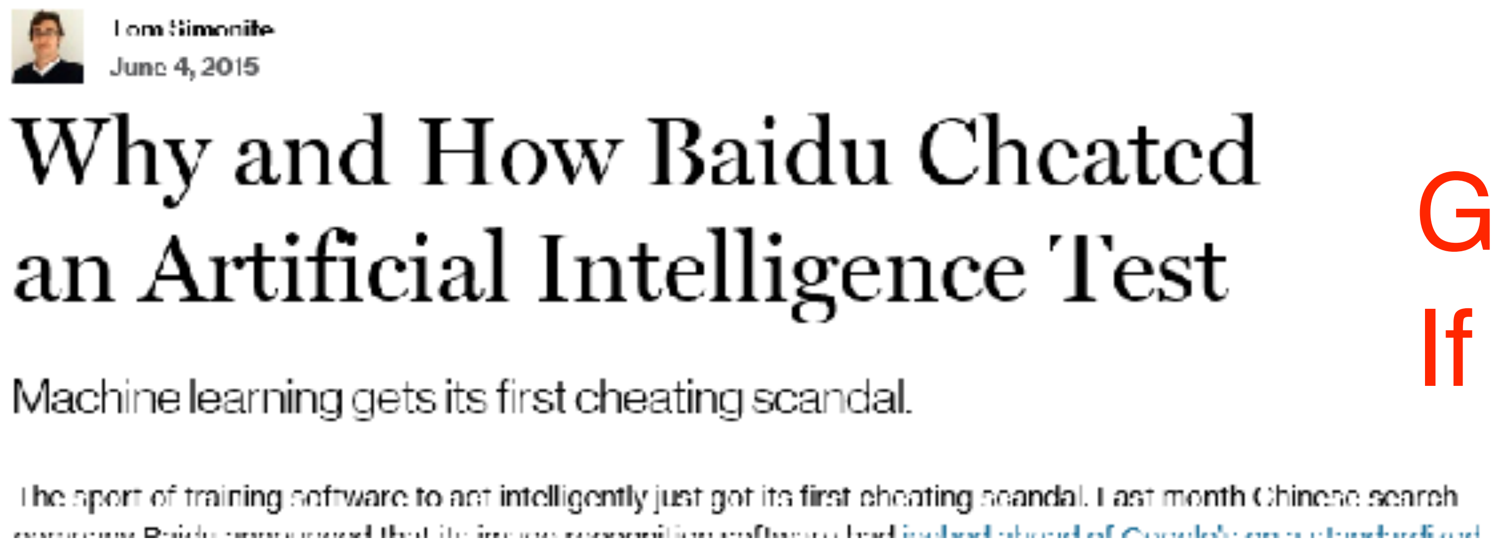
2, What if we don't have an explicit test set?

Possible training procedures if you only have a training set:

(1). Randomly split training set into "train" and "validate" set.

(2). Train model based on train set.

(3). Report validate set accuracy with this model.



Tom Simonite
June 4, 2015

Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

The sport of training software to act intelligently just got its first cheating scandal. Last month Chinese search

Golden rule: this test set cannot influence training in any way.
If you violate golden rule, you can overfit to the test data。

Supervised Learning Pipeline (Prepare for the Projects)

1, Given a training set X and y, with i.i.d assumption (training and test data drawn from same distribution), if we have an explicit test set to approximate test error:

Data:
X, y, X_{test}, y_{test}

1. Train:
$model = fit(X, y)$

2. Predict test set labels
$\hat{y} = predict(model, X_{test})$

3. Evaluate
$error = diff(\hat{y}, y_{test})$

2, What if we don't have an explicit test set?

Possible training procedures if you only have a training set:

(1). Randomly split training set into "train" and "validate" set.

(2). Train model based on train set.

(3). Report validate set accuracy with this model.

Tom Simonite
June 4, 2015

Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

The sport of training software to act intelligently just got its first cheating scandal. Last month Chinese search

Golden rule: this test set cannot influence training in any way.
If you violate golden rule, you can overfit to the test data。

Supervised Learning Pipeline (Prepare for the Projects)

1, Given a training set X and y, with i.i.d assumption (training and test data drawn from same distribution), if we have an explicit test set to approximate test error:

Data: X, y, X_{test}, y_{test}

1. Train: $model = fit(X, y)$

2. Predict test set labels $\hat{y} = predict(model, X_{test})$

3. Evaluate $error = diff(\hat{y}, y_{test})$

2, What if we don't have an explicit test set?

Possible training procedures if you only have a training set:

(1). Randomly split training set into "train" and "validate" set.

(2). Train model based on train set.

(3). Report validate set accuracy with this model.

$$X = \left[\begin{array}{c} \text{train} \\ \hdashline \text{validate} \end{array}\right] \quad y = \left[\begin{array}{c} \text{train} \\ \hdashline \text{validate} \end{array}\right]$$

Tom Simonite
June 4, 2015

Why and How Baidu Cheated an Artificial Intelligence Test

Machine learning gets its first cheating scandal.

The sport of training software to act intelligently just got its first cheating scandal. Last month Chinese search

Golden rule: this test set cannot influence training in any way.
If you violate golden rule, you can overfit to the test data。

What if we don’t have an explicit test set?(1)

Possible training procedures if you only have a training set.

1. Randomly split training set into “train” and “validate” set.
2. Train 10 models based on train set (e.g., 10 different bases)
3. Choose one with highest accuracy on validate set.
4. Report validate set accuracy with this model.

We should be a little skeptical of this accuracy:

–We violated golden rule on validation set:

•Approximation of test error was used to choose model.

–But we probably not overfitting much: only 10 models considered.

1. Randomly split training set into “train” and “validate” set.
2. Train 1 billion models based on train set.
3. Choose one with highest accuracy on validate set.
4. Report validate set accuracy with this model.

•We should be a very skeptical of this accuracy:

–We badly violated golden rule on validation set:

•High chance of overfitting to validation set.

What if we don’t have an explicit test set?(2)

Possible training procedures if you only have a training set.

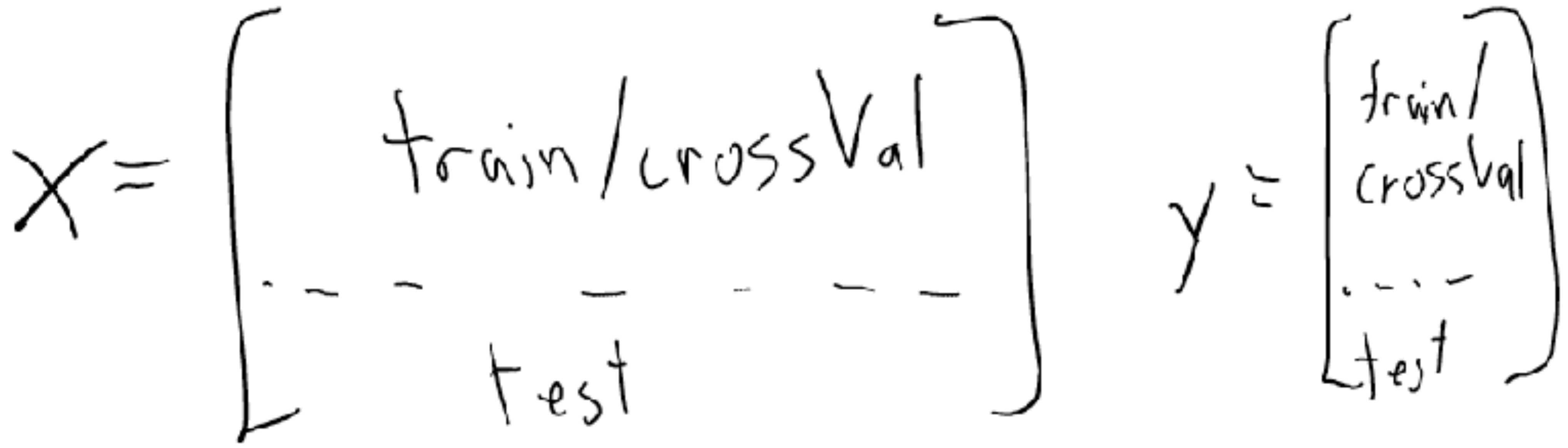
1. Randomly split training set into “train”, “validate”, and “test” set.
2. Train 1 billion models based on train set.
3. Choose one with highest accuracy on validate set.
4. Report test set accuracy with this model.

- We can trust this accuracy is reasonable.

–We might still overfit to validate set, but test set not used during training.

–Proper cross-validation procedure:

- Randomly split data into “train/crossValidate” and “test” set.
- Choose model with lowest cross-validation error on “train/crossValidate” set.
- Report error on “test” set which did not influence final model.



How to do Cross-Validation?

k-fold Cross Validation to estimate a tuning parameter λ

Arrange the training examples in a random order.

Divide the data into K roughly equal parts

1	2	3	4	5
Validation	Train	Train	Train	Train

do this for many values of λ and choose the value of λ that makes $CV(\lambda)$ smallest.

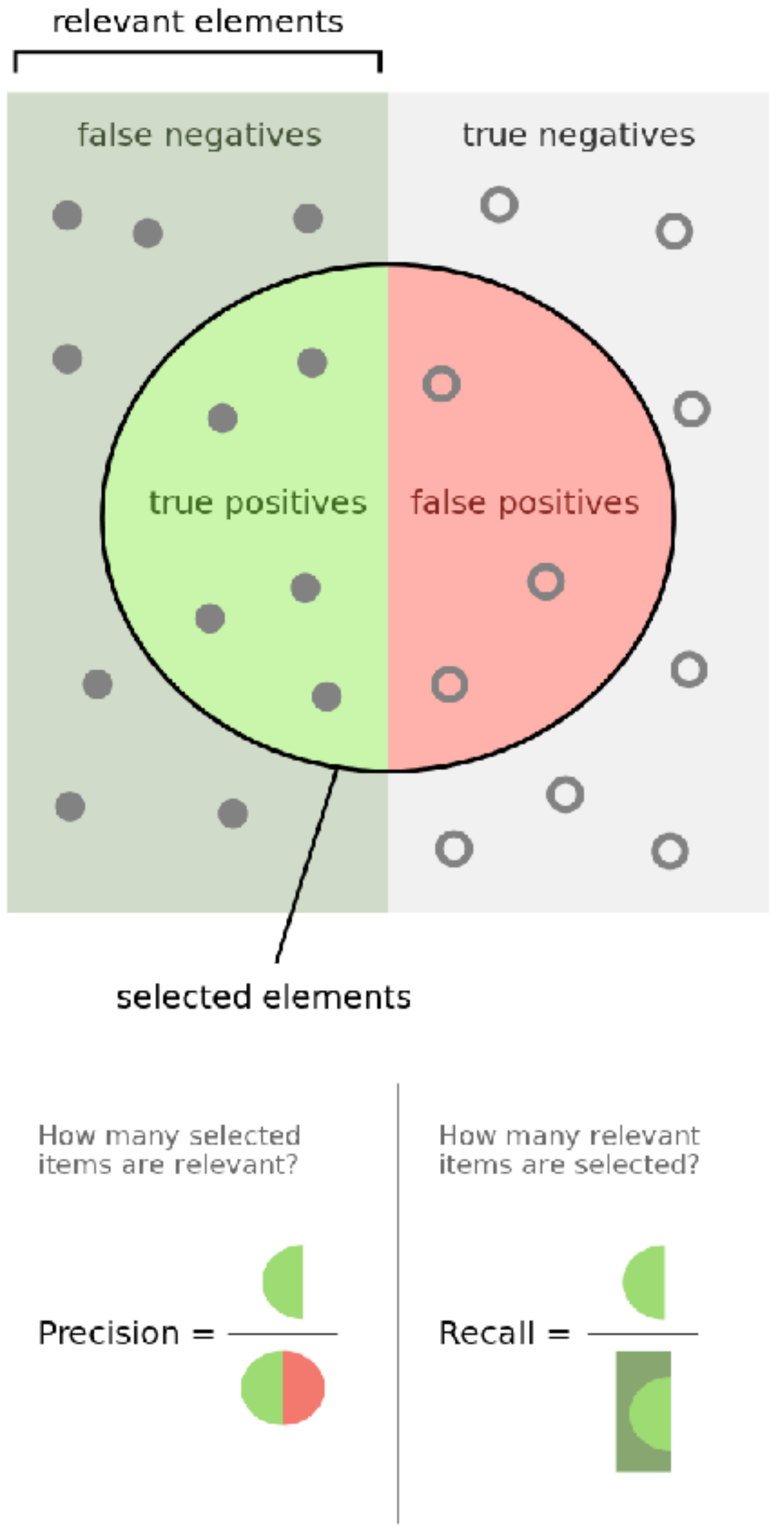
for each $k = 1, 2, \ldots K$, fit the model with parameter λ to the other $K - 1$ parts, giving $\hat{\beta}^{-k}(\lambda)$ and compute its error in predicting the kth part:

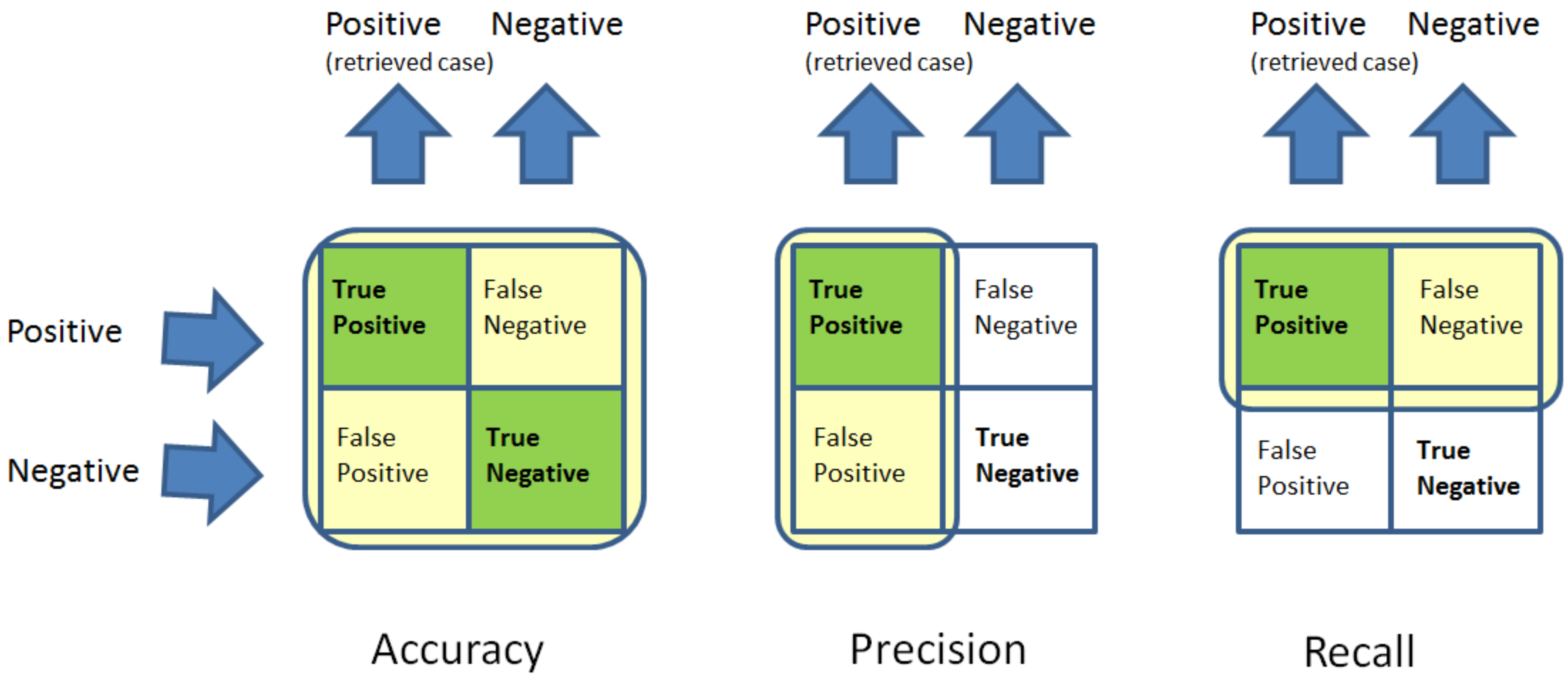
$E_k(\lambda) = \sum_{i \in kth\ part}(y_i - \mathbf{x}_i\hat{\beta}^{-k}(\lambda))^2$.

This gives the cross-validation error

$$CV(\lambda) = \frac{1}{K}\sum_{k=1}^{K} E_k(\lambda)$$

Errors of Different Kinds





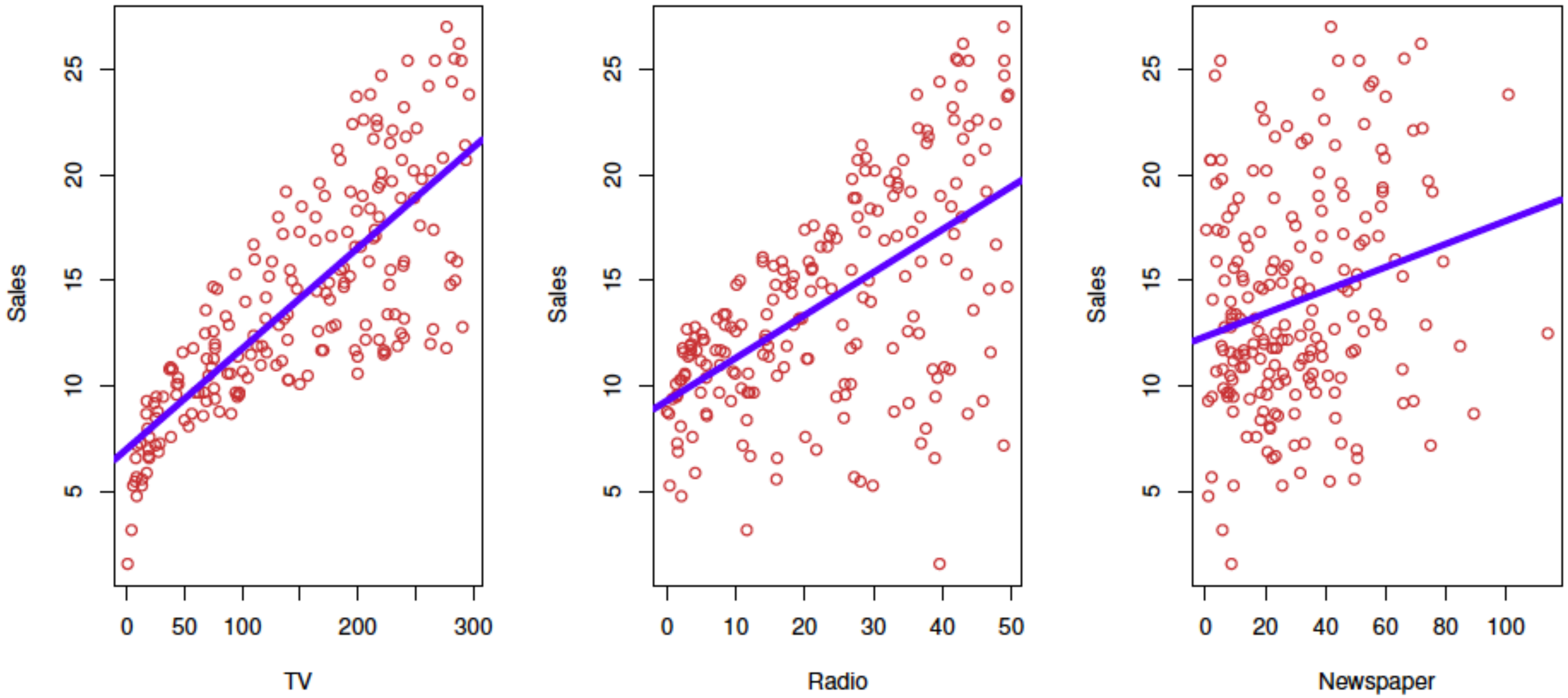
真实情况（ground-truth)	预测结果	
	正例	反例
正例	TP（真正例）	FN（假反例）
反例	FP（假正例）	TN（真反例）

Confusion Matrix

Multiple Linear Regression

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon,$$

$$\texttt{sales} = \beta_0 + \beta_1 \times \texttt{TV} + \beta_2 \times \texttt{radio} + \beta_3 \times \texttt{newspaper} + \epsilon.$$



Interpreting Regression Coefficients

β_j

Interpreting Regression Coefficients

- The ideal scenario is when the predictors are uncorrelated --- a balanced design:

- Each coefficient can be estimated and tested separately.

- Interpretations such as "a unit change in X_j is associated with a β_j change in Y , while all the other variables stay fixed", are possible.

- Correlations amongst predictors cause problems:

- The variance of all coefficients tends to increase, sometimes dramatically

- Interpretations become hazardous --- when X_j changes, everything else changes.

Multiple Linear Regression

Sec 3.2 of “The Elements of Statistical Learning”

$$X^T = (X_1, X_2, \ldots, X_p), \qquad f(X) = \beta_0 + \sum_{j=1}^{p} X_j\beta_j.$$

Least squares to minimize the residual sum of squares:

$$\begin{aligned}\text{RSS}(\beta) &= \sum_{i=1}^{N}(y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N}\Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2.\end{aligned}$$

$$\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta).$$

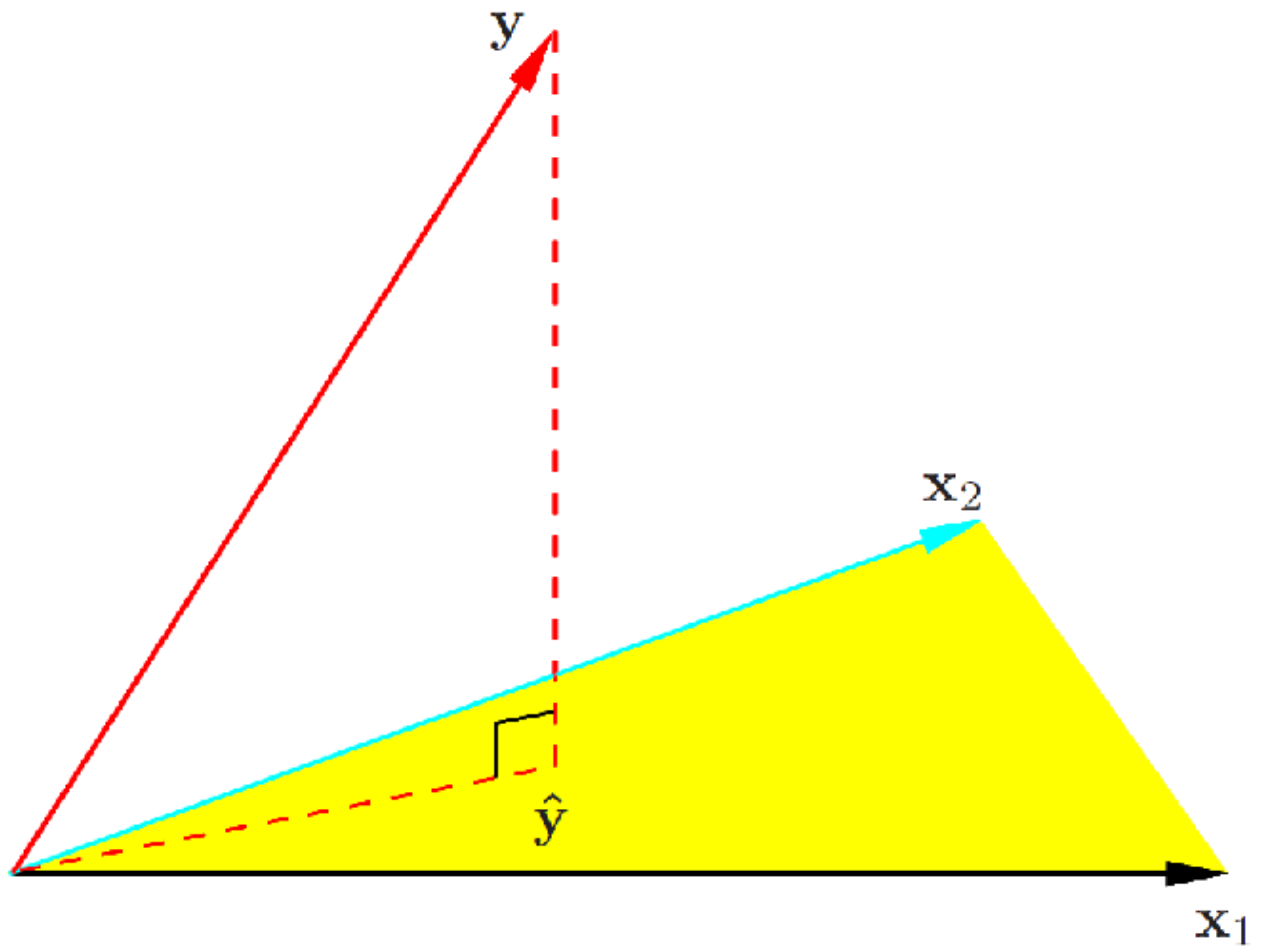
$$\frac{\partial \text{RSS}}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

$$\frac{\partial^2 \text{RSS}}{\partial\beta\partial\beta^T} = 2\mathbf{X}^T\mathbf{X}.$$

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},$$



Geometric interpretation.

Multiple Linear Regression

Sec 3.2 of “The Elements of Statistical Learning”

$$X^T = (X_1, X_2, \ldots, X_p), \qquad f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$

Least squares to minimize the residual sum of squares:

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2. \end{aligned}$$

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta).$$

$$\frac{\partial \mathrm{RSS}}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

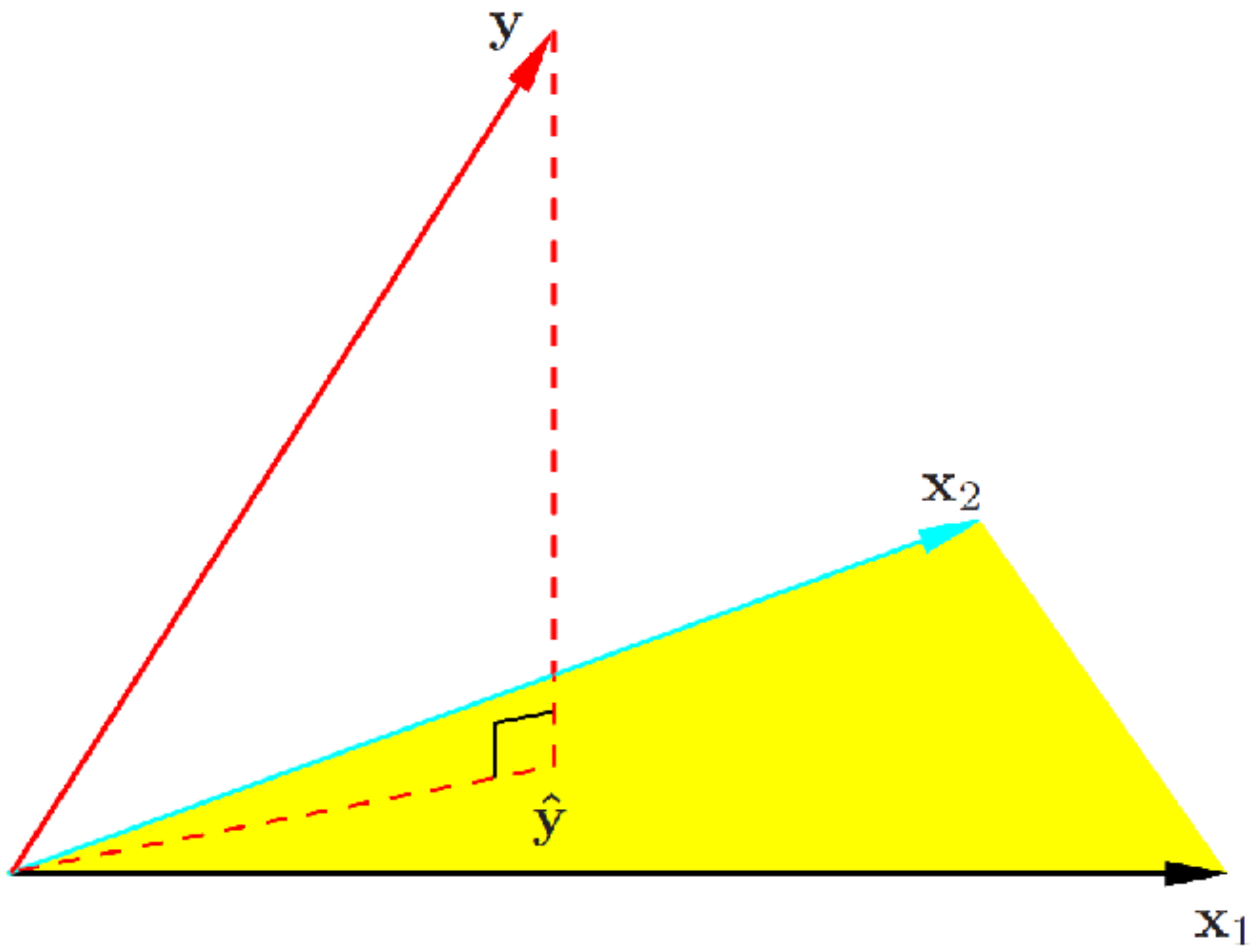
$$\frac{\partial^2 \mathrm{RSS}}{\partial \beta \partial \beta^T} = 2\mathbf{X}^T\mathbf{X}.$$

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \boxed{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}\mathbf{y},$$

Projection (Hat) matrix: $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$



Geometric interpretation.

Multiple Linear Regression

Sec 3.2 of “The Elements of Statistical Learning”

$$X^T = (X_1, X_2, \ldots, X_p), \qquad f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$

RSS denotes the **empirical risk** over the training set. It doesn’t assure the predictive performance over all inputs of interest.

Least squares to minimize the residual sum of squares:

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2. \end{aligned}$$

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta).$$

$$\frac{\partial \mathrm{RSS}}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

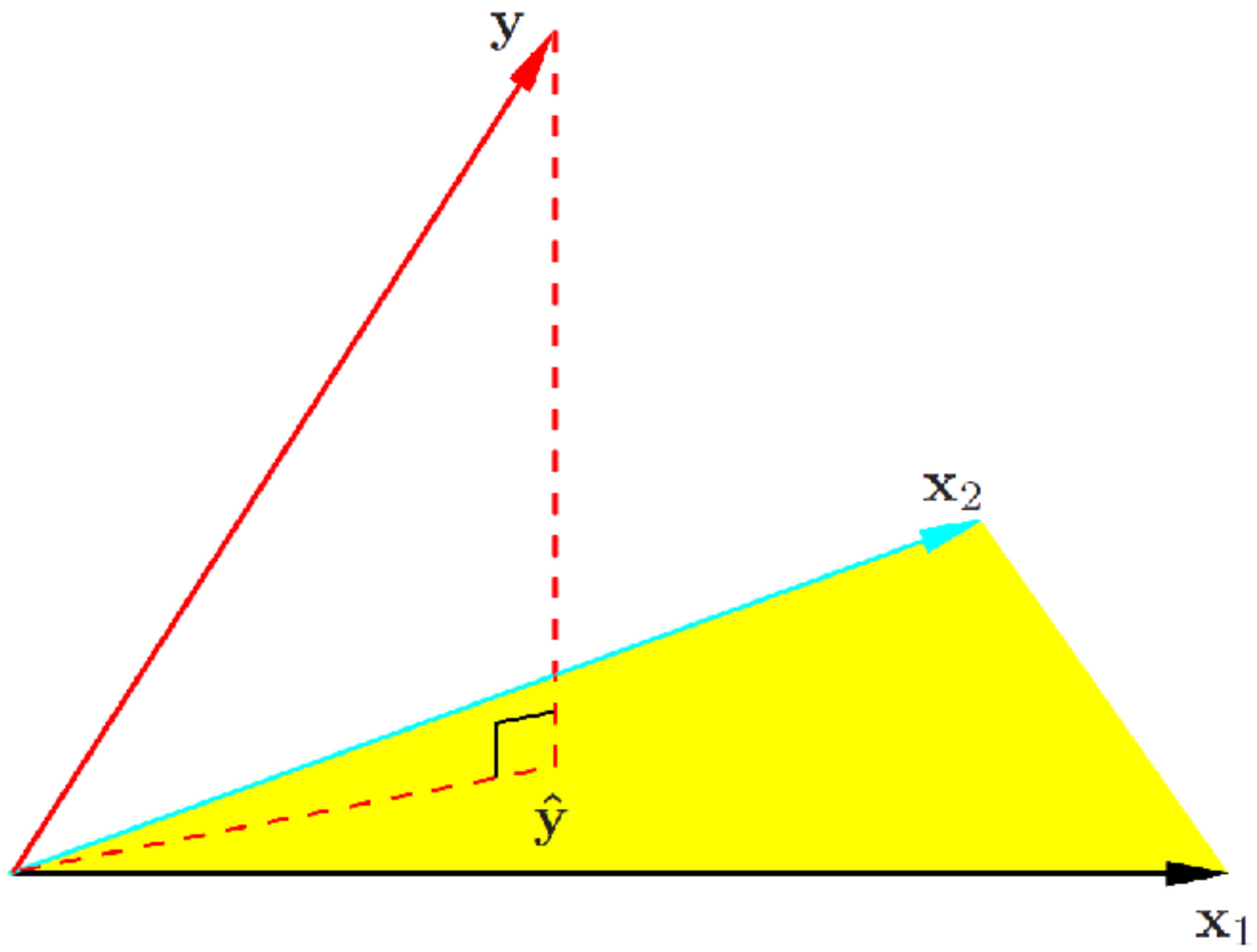
$$\frac{\partial^2 \mathrm{RSS}}{\partial \beta \partial \beta^T} = 2\mathbf{X}^T\mathbf{X}.$$

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \boxed{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}\mathbf{y},$$

Projection (Hat) matrix: $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$



Geometric interpretation.

Multiple Linear Regression

Sec 3.2 of "The Elements of Statistical Learning"

$$X^T = (X_1, X_2, \ldots, X_p), \qquad f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$

RSS denotes the **empirical risk** over the training set. It doesn't assure the predictive performance over all inputs of interest.

Least squares to minimize the residual sum of squares:

$$\begin{aligned} \mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2. \end{aligned}$$

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta).$$

$$\frac{\partial \mathrm{RSS}}{\partial \beta} = -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta)$$

$$\frac{\partial^2 \mathrm{RSS}}{\partial \beta \partial \beta^T} = 2\mathbf{X}^T\mathbf{X}.$$

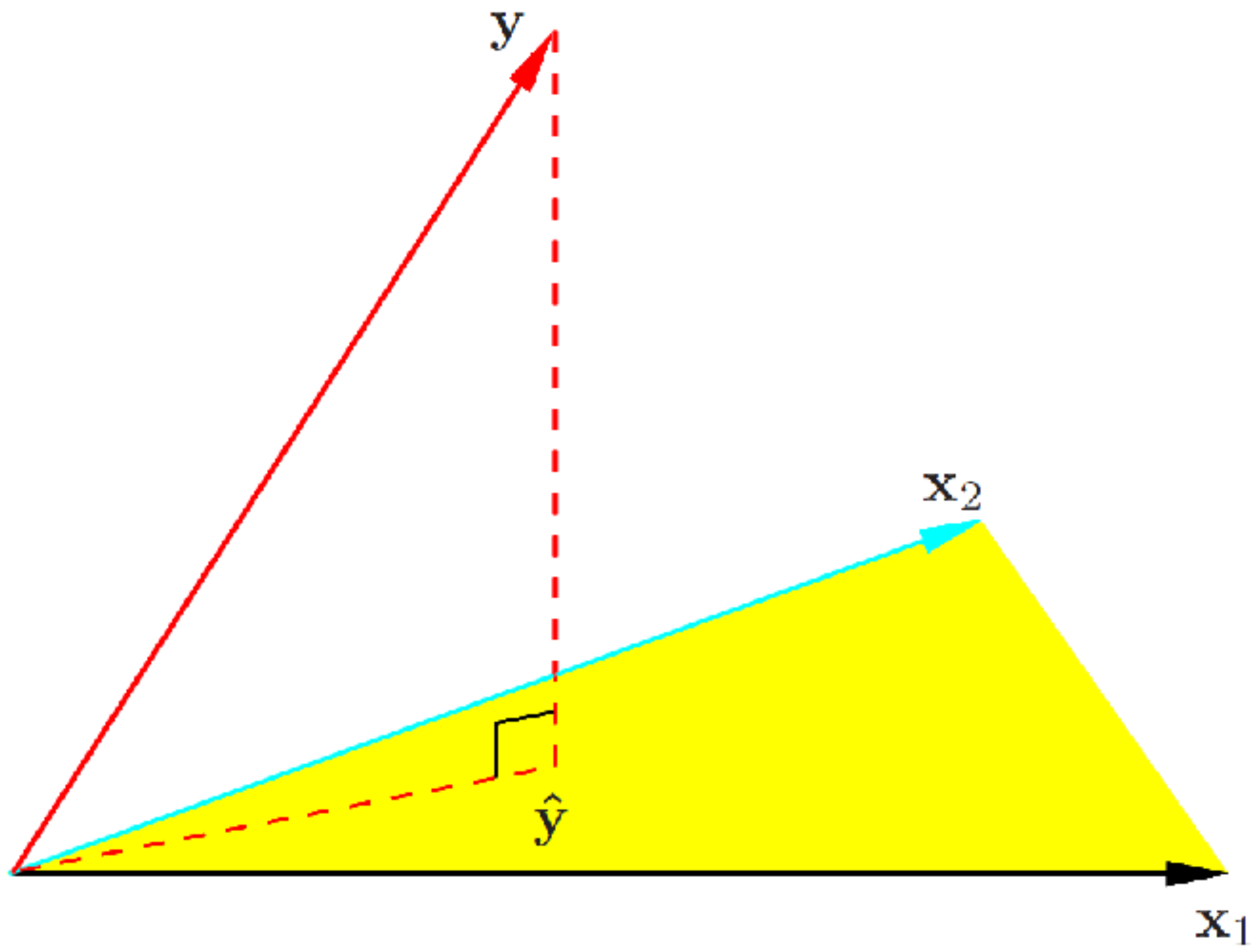
Note that: For a unique solution, the matrix X^TX must be full rank.

$$\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \boxed{\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T}\mathbf{y},$$

Projection (Hat) matrix: $\mathbf{H} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$



Geometric interpretation.

Multiple Linear Regression

Sec 3.2 of “The Elements of Statistical Learning”

$$X^T = (X_1, X_2, \ldots, X_p), \qquad f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j.$$

RSS denotes the **empirical risk** over the training set. It doesn’t assure the predictive performance over all inputs of interest.

Least squares to minimize the residual sum of squares:

$$\begin{aligned}\mathrm{RSS}(\beta) &= \sum_{i=1}^{N} (y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N} \Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \Big)^2.\end{aligned}$$

$$\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta).$$

$$\frac{\partial \mathrm{RSS}}{\partial \beta} = -2\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta)$$

$$\frac{\partial^2 \mathrm{RSS}}{\partial \beta \partial \beta^T} = 2\mathbf{X}^T \mathbf{X}.$$

Note that: For a unique solution, the matrix X^TX must be full rank.

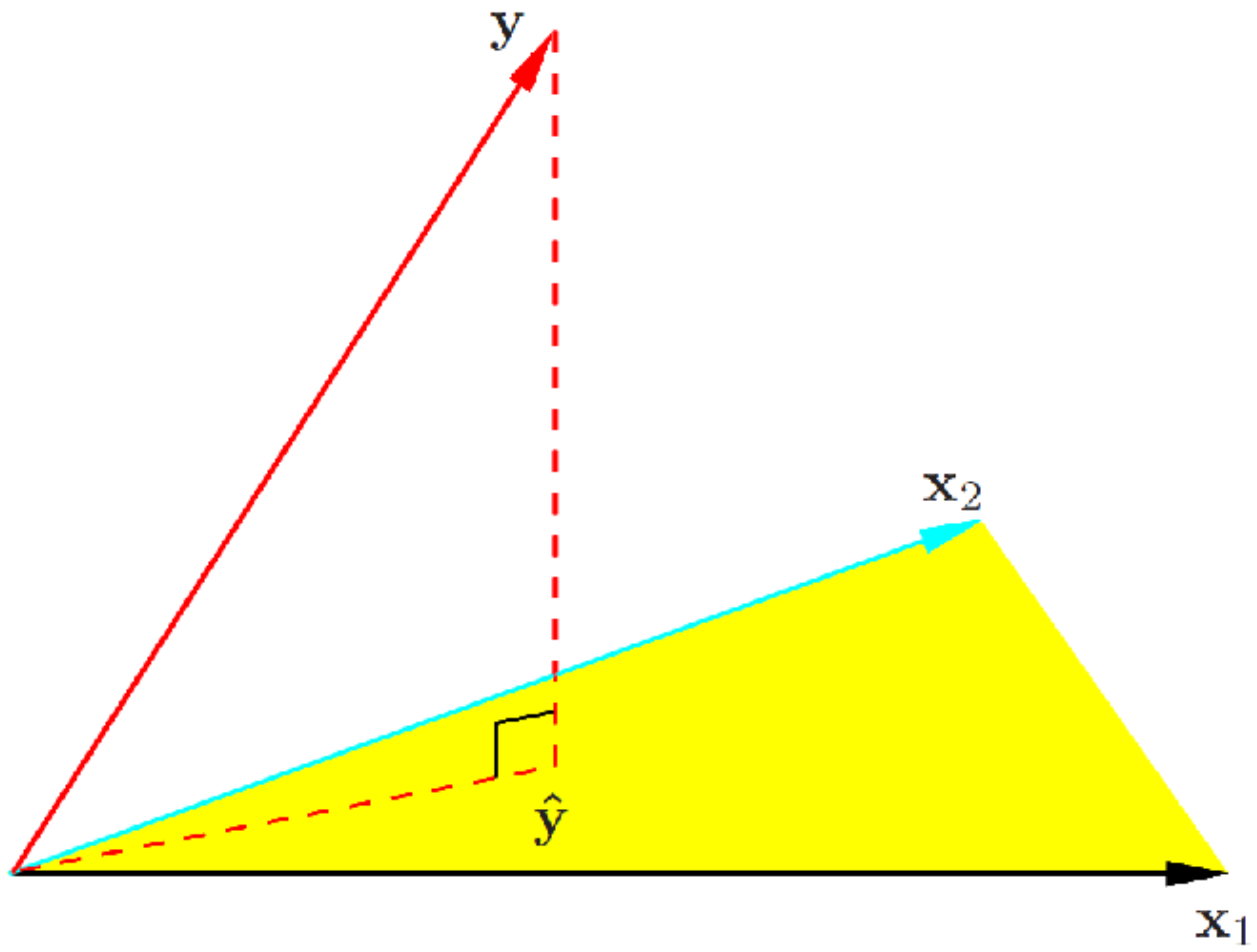
$$\mathbf{X}^T (\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

$$\hat{\mathbf{y}} = \mathbf{X}\hat{\beta} = \boxed{\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T} \mathbf{y},$$

Orthogonal Projection of Y on the space spanned by the columns of X.

Projection (Hat) matrix: $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$



Geometric interpretation.

Some Important Questions of Multiple Linear Regression

- Is at least one of the predictors $X_1, X_2, \ldots, X_p$ useful in predicting the response?

F-statistic: $$F = \frac{(\mathrm{TSS} - \mathrm{RSS})/p}{\mathrm{RSS}/(n-p-1)} \sim F_{p,n-p-1}$$

Hypothesis test

one parameter : t-test

two or more parameters: F-test

- How well does the model fit the data?

$$\mathrm{RSE} = \sqrt{\frac{1}{n-p-1}\mathrm{RSS}}, \quad R^2 = \mathrm{Cor}(Y, \hat{Y})^2$$

p-values considered harmful (page 212-213, Murphy’s book)

Summary of Linear Model

Optional subtitle

- Despite its simplicity, the linear model has distinct advantages in terms of its interpretability and often shows good predictive performance.
- Generalizations of the Linear Model:
  - Classification problems: logistic regression, support vector machines
  - Non-linearity: kernel smoothing, splines and generalized additive models; nearest neighbor methods.
  - Interactions: Tree-based methods, bagging, random forests and boosting (these also capture non-linearities);
  - Regularized fitting: Ridge regression and lasso;

Chap 2 - Linear Regression(2)

Linear Model Selection and Regularisation

—ref: Chap 6.1, 6.2, [James,2013]

1.Subset Selection;

2.Shrinkage Methods

- Ridge Regression
- The Lasso

We need Alternatives instead of Least Squares

Optional subtitle

- Prediction Accuracy: especially when p > n, to control the variance. [Example: homework]
- Model interpretability: By removing irrelevant features —that is, by setting the corresponding coefficient estimates to zero— we can obtain a model that is more easily interpreted.

Three methods to perform feature selection:

We need Alternatives instead of Least Squares

Optional subtitle

- Prediction Accuracy: especially when p > n, to control the variance. [Example: homework]
- Model interpretability: By removing irrelevant features —that is, by setting the corresponding coefficient estimates to zero— we can obtain a model that is more easily interpreted.

Three methods to perform feature selection:

- Subset Selection. We identify a subset of the ***p*** predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.

We need Alternatives instead of Least Squares

Optional subtitle

- Prediction Accuracy: especially when p > n, to control the variance. [Example: homework]
- Model interpretability: By removing irrelevant features —that is, by setting the corresponding coefficient estimates to zero— we can obtain a model that is more easily interpreted.

Three methods to perform feature selection:

- Subset Selection. We identify a subset of the ***p*** predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.
- Shrinkage. We fit a model involving all ***p*** predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance and can also perform variable selection.

We need Alternatives instead of Least Squares

Optional subtitle

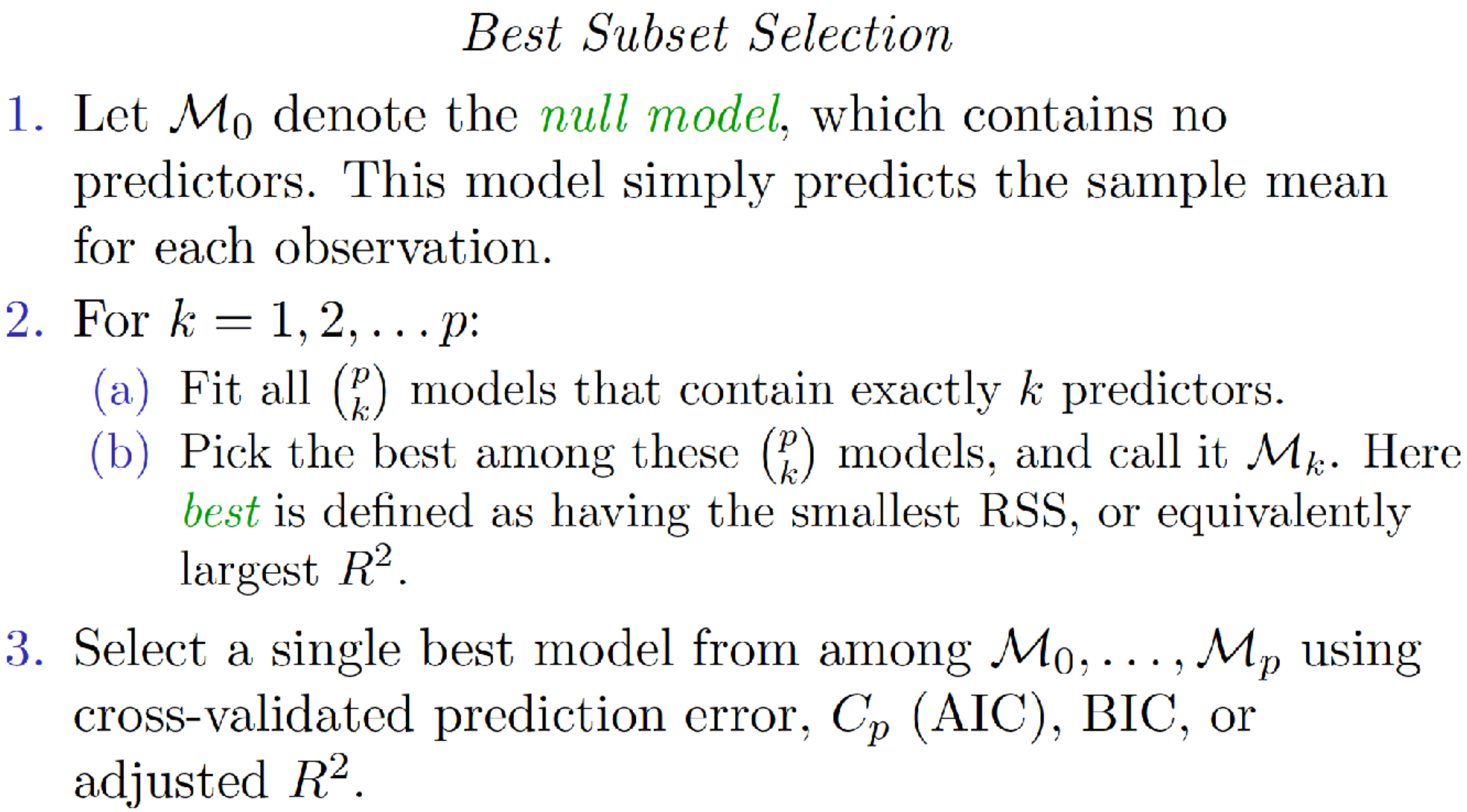
- Prediction Accuracy: especially when p > n, to control the variance. [Example: homework]
- Model interpretability: By removing irrelevant features —that is, by setting the corresponding coefficient estimates to zero— we can obtain a model that is more easily interpreted.

Three methods to perform feature selection:

- Subset Selection. We identify a subset of the ***p*** predictors that we believe to be related to the response. We then fit a model using least squares on the reduced set of variables.
- Shrinkage. We fit a model involving all ***p*** predictors, but the estimated coefficients are shrunken towards zero relative to the least squares estimates. This shrinkage (also known as regularization) has the effect of reducing variance and can also perform variable selection.
- Dimension Reduction. We project the p predictors into a M-dimensional subspace, where M < p. This is achieved by computing M different linear combinations, or projections, of the variables. Then these M projections are used as predictors to t a linear regression model by least squares.

Subset Selection — Best Subset Selection

also ref Chap 3.3 [Hastie 2011]



Best Subset Selection

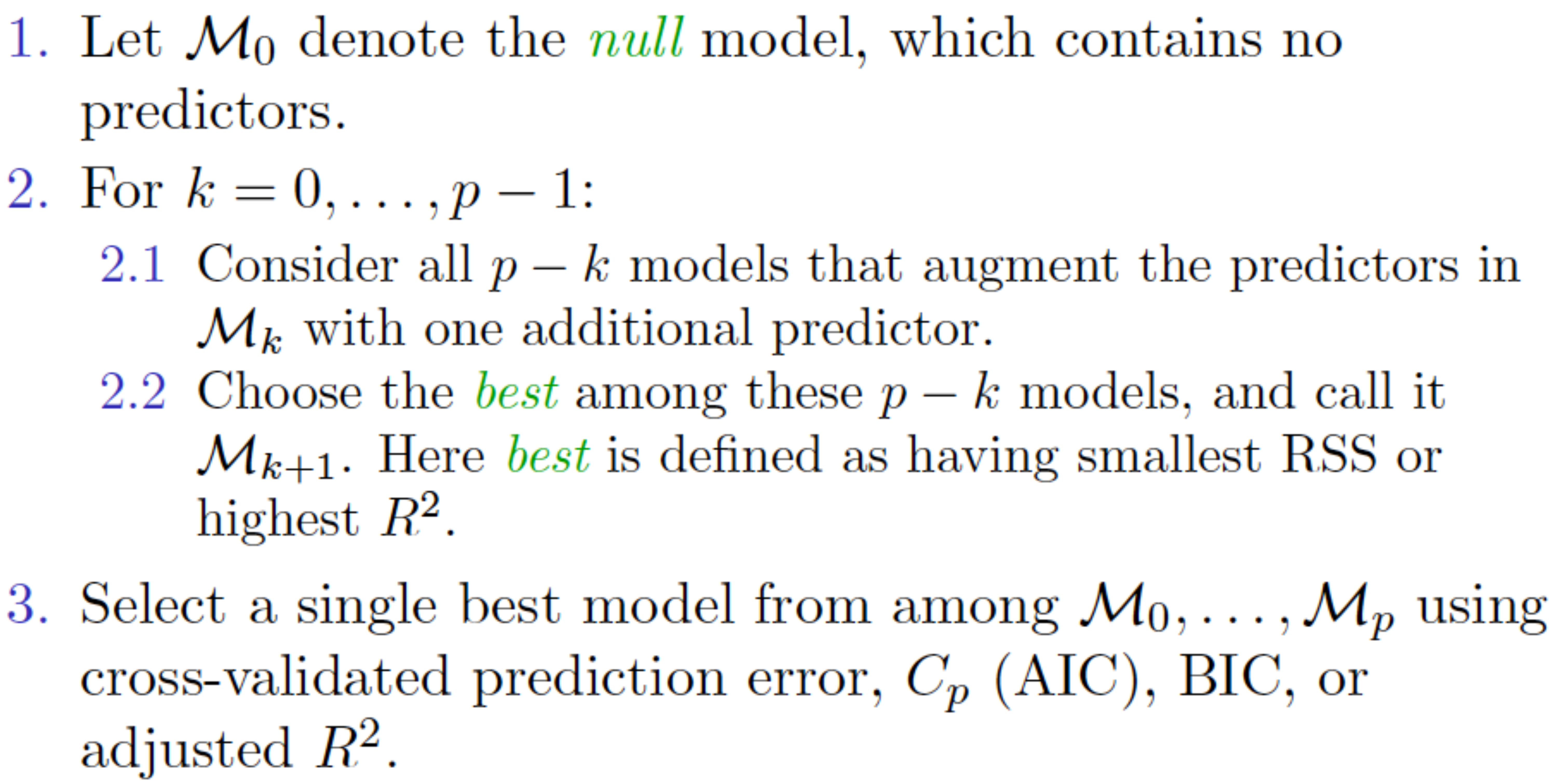
1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Stepwise Selection

- For computational reasons, best subset selection cannot be applied with very large p. *Why not?*
- Best subset selection may also suffer from statistical problems when p is large: larger the search space, the higher the chance of finding models that look good on the training data, even though they might not have any predictive power on future data.
- Thus an enormous search space can lead to *overfitting* and high variance of the coefficient estimates.
- For both of these reasons, *stepwise* methods, which explore a far more restricted set of models, are attractive alternatives to best subset selection.

Forward Stepwise Selection

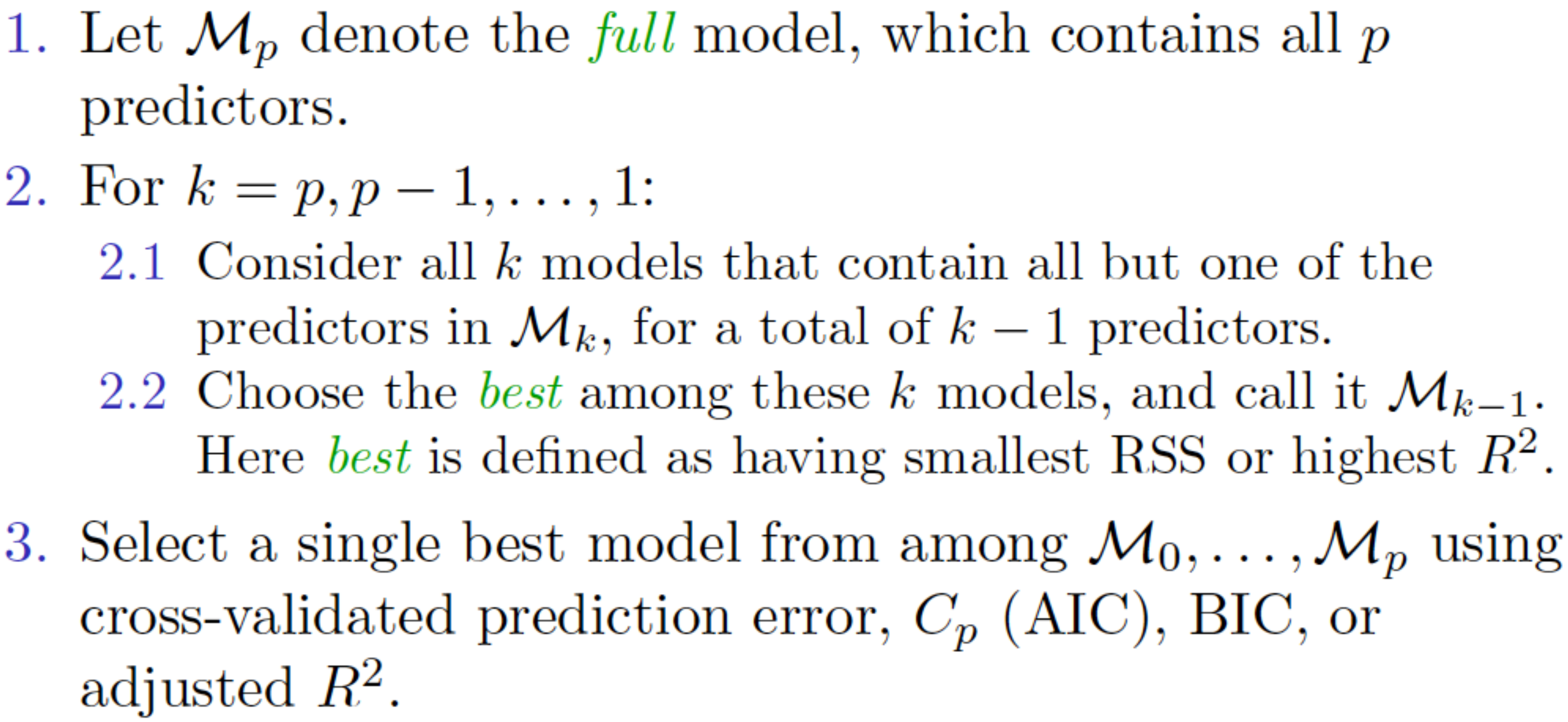
- Forward stepwise selection begins with a model containing no predictors, and then adds predictors to the model, one-at-a-time, until all of the predictors are in the model.



1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   2.1 Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   2.2 Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Backward Stepwise Selection

- Backward stepwise selection begins with the full least squares model containing all p predictors, and then iteratively removes the least useful predictor, one-at-a-time.



```
1. Let $\mathcal{M}_p$ denote the *full* model, which contains all $p$
   predictors.
2. For $k = p, p-1, \ldots, 1$:
   2.1 Consider all $k$ models that contain all but one of the
       predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   2.2 Choose the *best* among these $k$ models, and call it $\mathcal{M}_{k-1}$.
       Here *best* is defined as having smallest RSS or highest $R^2$.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using
   cross-validated prediction error, $C_p$ (AIC), BIC, or
   adjusted $R^2$.
```

Choosing the Optimal Model

1, AIC, BIC, Cp, and adjusted R2;

- C_p, AIC, and BIC all have rigorous theoretical justifications
- 该课程对此不做要求

2, Cross-Validation.

- Cross Validation has an advantage relative to AIC, BIC, Cp, and adjusted R2, in that it provides a direct estimate of the test error, and makes fewer assumptions about the true underlying model.
- 需要自己动手实现相应的代码。

Shrinkage Methods(1)

- Ridge Regression

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

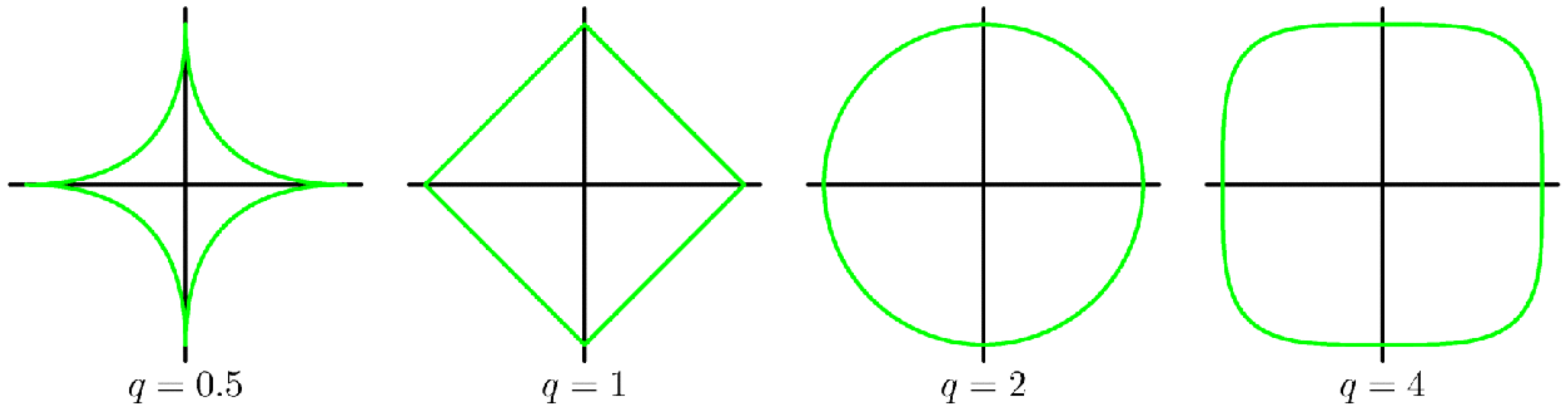
where $\lambda \geq 0$ is a *tuning parameter*, to be determined separately.

- Lasso

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| \;=\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

Shrinkage Methods in Matrix Form

$$\underset{\beta}{\operatorname{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_q$$



$\parallel \beta \parallel_0 = \sharp\sigma(\beta)$ $\sharp$ $\sigma(\beta)$ β

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j{}^2}.$$

Note: (1) tuning the parameter λ is very important.

$$\parallel \beta \parallel_q = \left(\sum_{i=1}^{p} |\beta_i|^q \right)^{\frac{1}{q}}$$

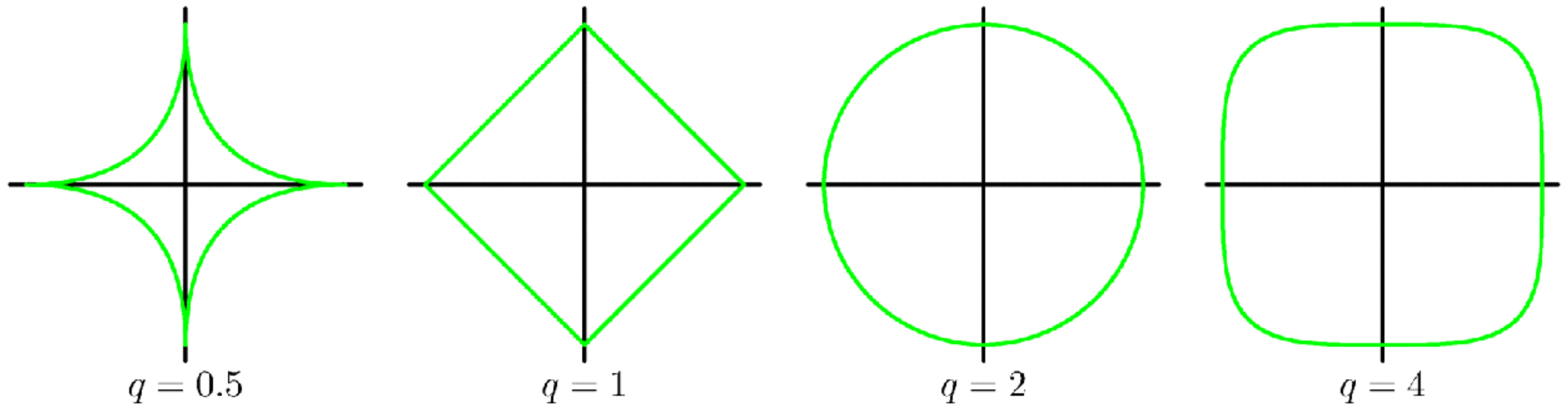
[1] Mila Nikolova, Description of the minimizers of least squares regularized with ℓ0-norm. Uniqueness of the global minimizer, SIAM J. IMAGING SCIENCE 2013.

[2] Yiyuan She, and Art B. Owen, Outlier Detection Using Nonconvex Penalized Regression, 2011. Journal of the American Statistical Association

[3] Yanwei Fu et al. Robust Subjective Visual Property Prediction from Crowdsourced Pairwise Labels. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2016

Shrinkage Methods in Matrix Form

$$\underset{\beta}{\operatorname{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_q$$



q=0, L$_0$-norm; —> finding the minimiser is NP-hard computational problem. (the Eq. is nonconvex).

- L$_0$-norm has closed form solution [1].
- it is defined in Eq(6.10) of textbook. i.e., $\parallel \beta \parallel_0 = \sharp\sigma(\beta)$, $\sharp$ stands for cardinality; $\sigma(\beta)$ is the support of β

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} {\beta_j}^2}.$$

Note: (1) tuning the parameter λ is very important.

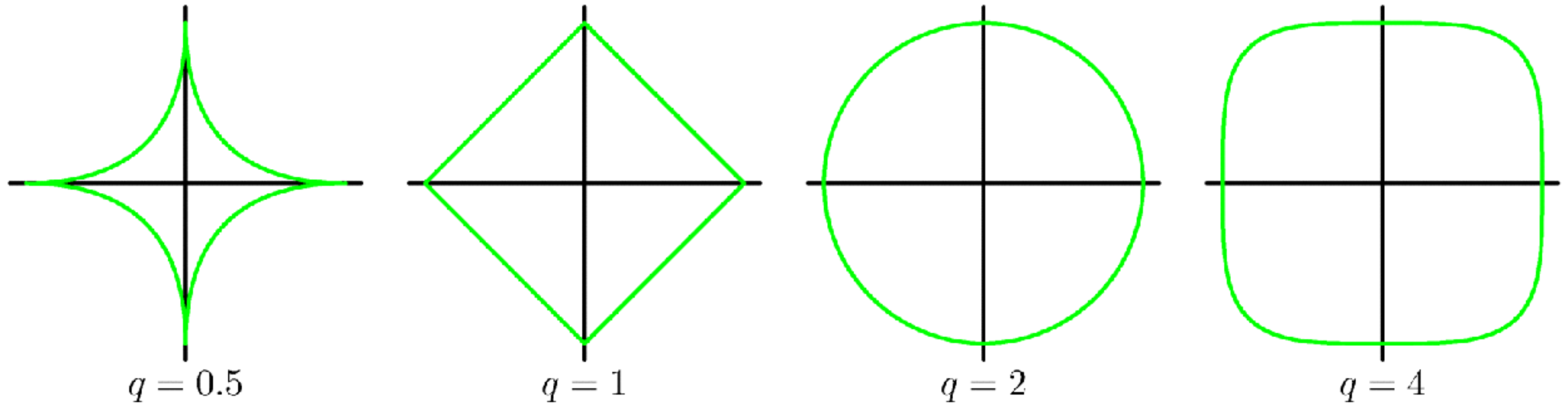
$$\parallel \beta \parallel_q = \left(\sum_{i=1}^{p} |\beta_i|^q \right)^{\frac{1}{q}}$$

[1] Mila Nikolova, Description of the minimizers of least squares regularized with ℓ0-norm. Uniqueness of the global minimizer, SIAM J. IMAGING SCIENCE 2013.

[2] Yiyuan She, and Art B. Owen, Outlier Detection Using Nonconvex Penalized Regression, 2011. Journal of the American Statistical Association

[3] Yanwei Fu et al. Robust Subjective Visual Property Prediction from Crowdsourced Pairwise Labels. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2016

Shrinkage Methods in Matrix Form



$$\underset{\beta}{\text{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_q$$

q=0, L0-norm; —> finding the minimiser is NP-hard computational problem. (the Eq. is nonconvex).

- L0-norm has closed form solution [1].
- it is defined in Eq(6.10) of textbook. i.e., $\parallel \beta \parallel_0 = \sharp\sigma(\beta)$, $\sharp$ stands for cardinality; $\sigma(\beta)$ is the support of β

q<1, **hard-thresholding**

$$\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j{}^2}.$$

Note: (1) tuning the parameter λ is very important.

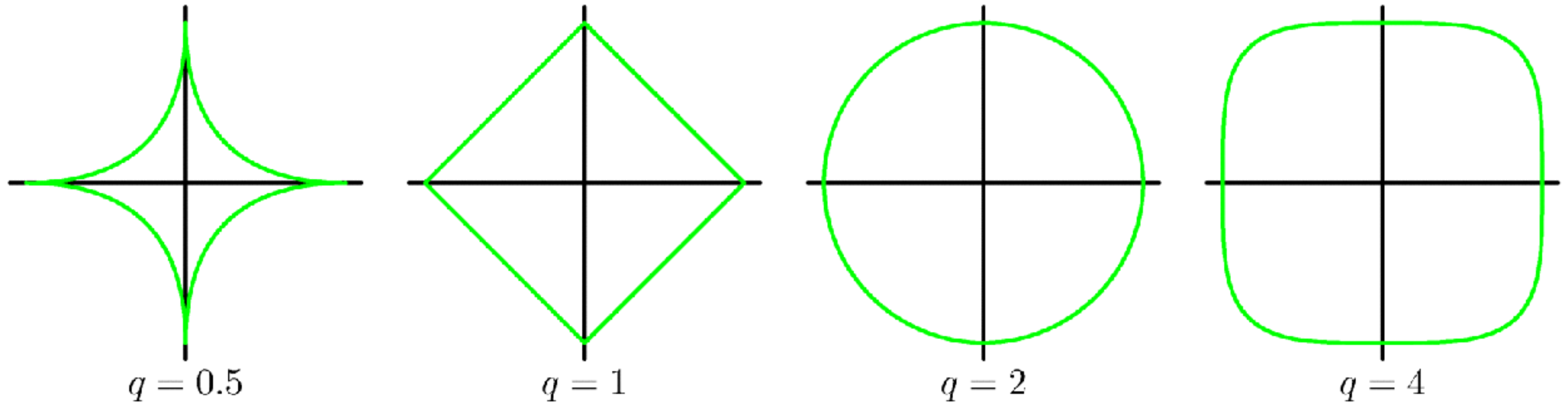
$$\parallel \beta \parallel_q = \left(\sum_{i=1}^{p} |\beta_i|^q \right)^{\frac{1}{q}}$$

[1] Mila Nikolova, Description of the minimizers of least squares regularized with ℓ0-norm. Uniqueness of the global minimizer, SIAM J. IMAGING SCIENCE 2013.

[2] Yiyuan She, and Art B. Owen, Outlier Detection Using Nonconvex Penalized Regression, 2011. Journal of the American Statistical Association

[3] Yanwei Fu et al. Robust Subjective Visual Property Prediction from Crowdsourced Pairwise Labels. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2016

Shrinkage Methods in Matrix Form



$$\underset{\beta}{\operatorname{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_q$$

q=0, L_0-norm; —> finding the minimiser is NP-hard computational problem. (the Eq. is nonconvex).

- L_0-norm has closed form solution [1].
- it is defined in Eq(6.10) of textbook. i.e., $\parallel \beta \parallel_0 = \sharp\sigma(\beta)$, $\sharp$ stands for cardinality; $\sigma(\beta)$ is the support of β

q<1, **hard-thresholding**

q=1, L_1-norm —> Lasso (convex), a.k.a., **soft-thresholding**.

$$\|\beta\|_2 = \sqrt{\textstyle\sum_{j=1}^{p} {\beta_j}^2}.$$

Note: (1) tuning the parameter λ is very important.

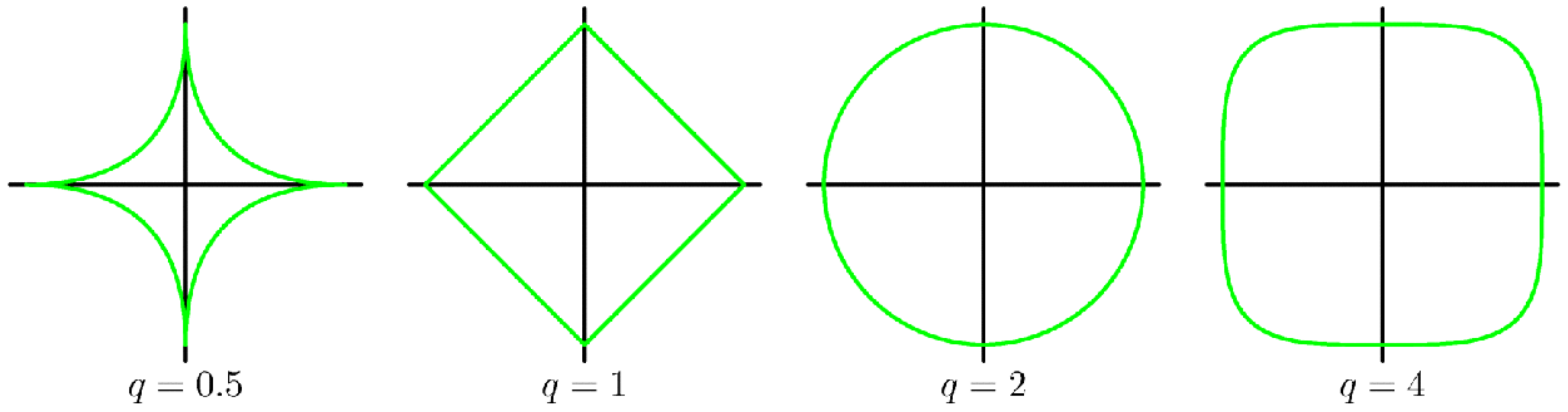
$$\parallel \beta \parallel_q = \left(\sum_{i=1}^{p} |\beta_i|^q \right)^{\frac{1}{q}}$$

[1] Mila Nikolova, Description of the minimizers of least squares regularized with ℓ0-norm. Uniqueness of the global minimizer, SIAM J. IMAGING SCIENCE 2013.

[2] Yiyuan She, and Art B. Owen, Outlier Detection Using Nonconvex Penalized Regression, 2011. Journal of the American Statistical Association

[3] Yanwei Fu et al. Robust Subjective Visual Property Prediction from Crowdsourced Pairwise Labels. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2016

Shrinkage Methods in Matrix Form



$$\underset{\beta}{\operatorname{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_q$$

q=0, L$_0$-norm; —> finding the minimiser is NP-hard computational problem. (the Eq. is nonconvex).

- L$_0$-norm has closed form solution [1].
- it is defined in Eq(6.10) of textbook. i.e., $\parallel \beta \parallel_0 = \sharp\sigma(\beta)$, $\sharp$ stands for cardinality; $\sigma(\beta)$ is the support of β

q<1, **hard-thresholding**

q=1, L$_1$-norm —> Lasso (convex), a.k.a., **soft-thresholding**.

q=2, L$_2$-norm —> Ridge Regression (convex) $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} {\beta_j}^2}$.

Note: (1) tuning the parameter λ is very important.

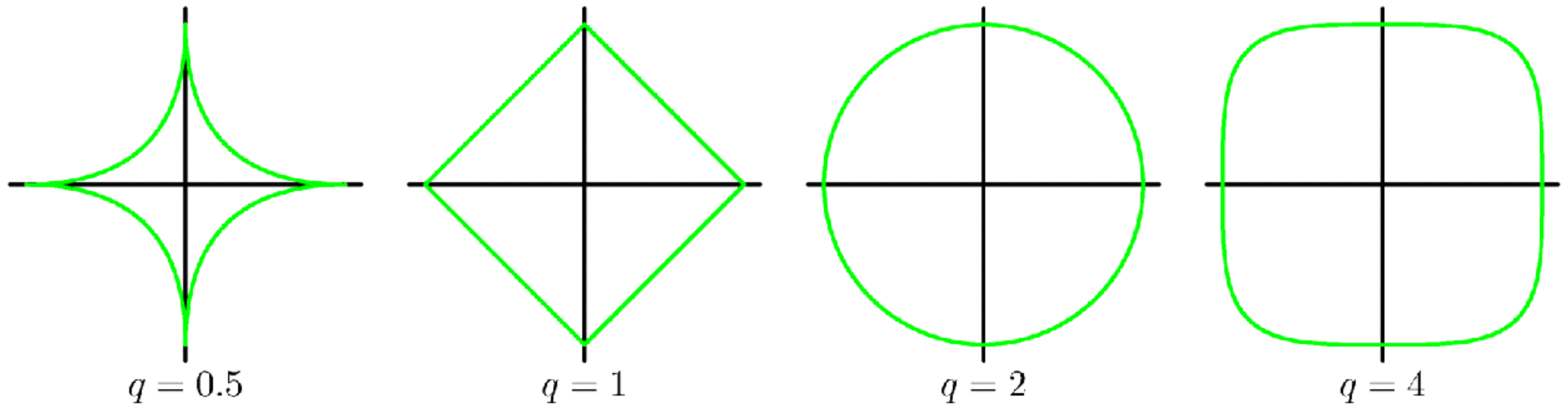
$$\parallel \beta \parallel_q = \left(\sum_{i=1}^{p} |\beta_i|^q \right)^{\frac{1}{q}}$$

[1] Mila Nikolova, Description of the minimizers of least squares regularized with ℓ0-norm. Uniqueness of the global minimizer, SIAM J. IMAGING SCIENCE 2013.

[2] Yiyuan She, and Art B. Owen, Outlier Detection Using Nonconvex Penalized Regression, 2011. Journal of the American Statistical Association

[3] Yanwei Fu et al. Robust Subjective Visual Property Prediction from Crowdsourced Pairwise Labels. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2016

Shrinkage Methods in Matrix Form



$$\underset{\beta}{\operatorname{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_q$$

q=0, L0-norm; —> finding the minimiser is NP-hard computational problem. (the Eq. is nonconvex).

- L0-norm has closed form solution [1].
- it is defined in Eq(6.10) of textbook. i.e., $\parallel \beta \parallel_0 = \sharp\sigma(\beta)$, $\sharp$ stands for cardinality; $\sigma(\beta)$ is the support of β

q<1, **hard-thresholding**

q=1, L1-norm —> Lasso (convex), a.k.a., **soft-thresholding**.

q<=1 used for outlier detection [2,3].

q=2, L2-norm —> Ridge Regression (convex) $\|\beta\|_2 = \sqrt{\sum_{j=1}^{p} \beta_j{}^2}$.

Note: (1) tuning the parameter λ is very important.

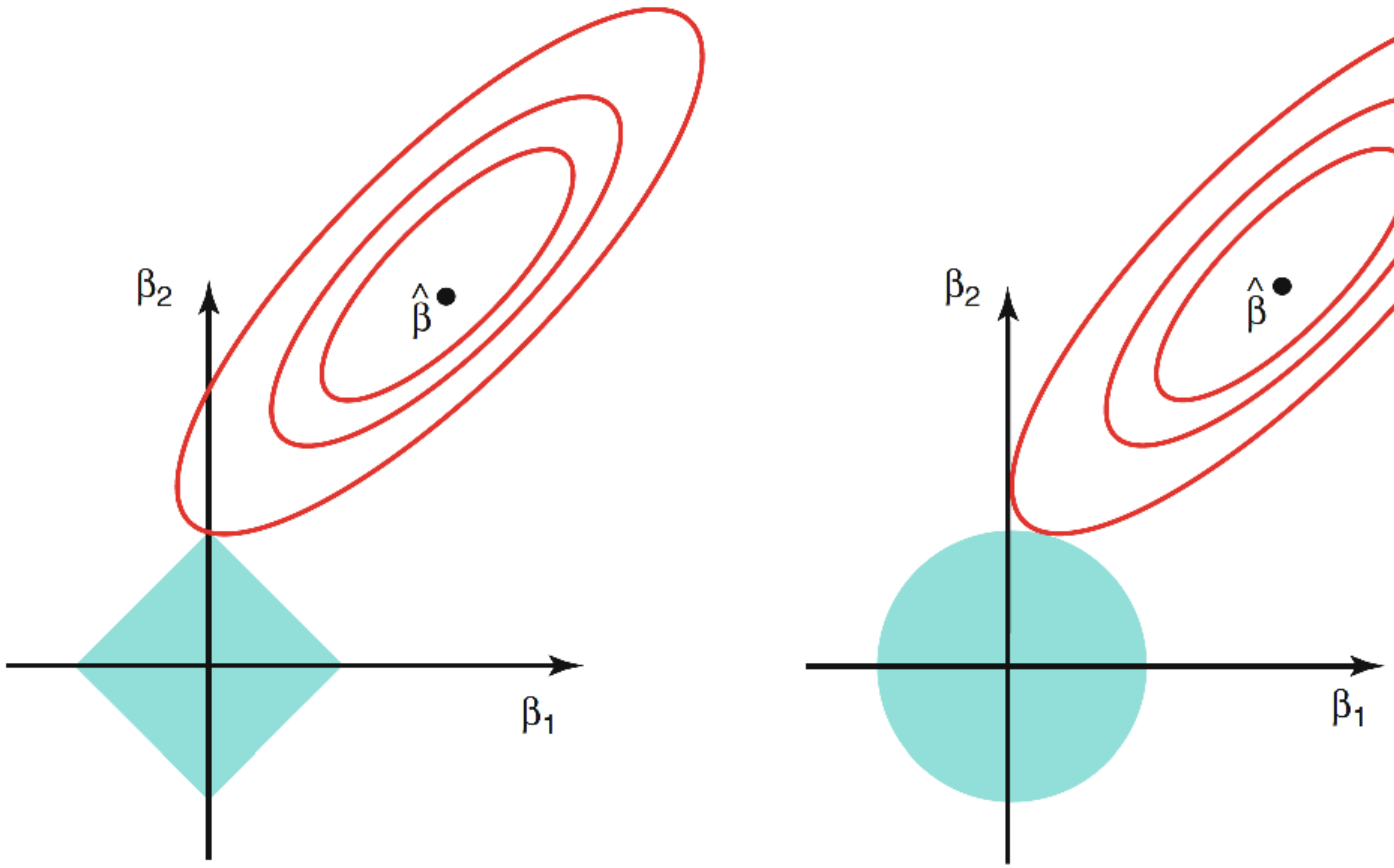
$$\parallel \beta \parallel_q = \left(\sum_{i=1}^{p} |\beta_i|^q \right)^{\frac{1}{q}}$$

[1] Mila Nikolova, Description of the minimizers of least squares regularized with ℓ0-norm. Uniqueness of the global minimizer, SIAM J. IMAGING SCIENCE 2013.

[2] Yiyuan She, and Art B. Owen, Outlier Detection Using Nonconvex Penalized Regression, 2011. Journal of the American Statistical Association

[3] Yanwei Fu et al. Robust Subjective Visual Property Prediction from Crowdsourced Pairwise Labels. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2016

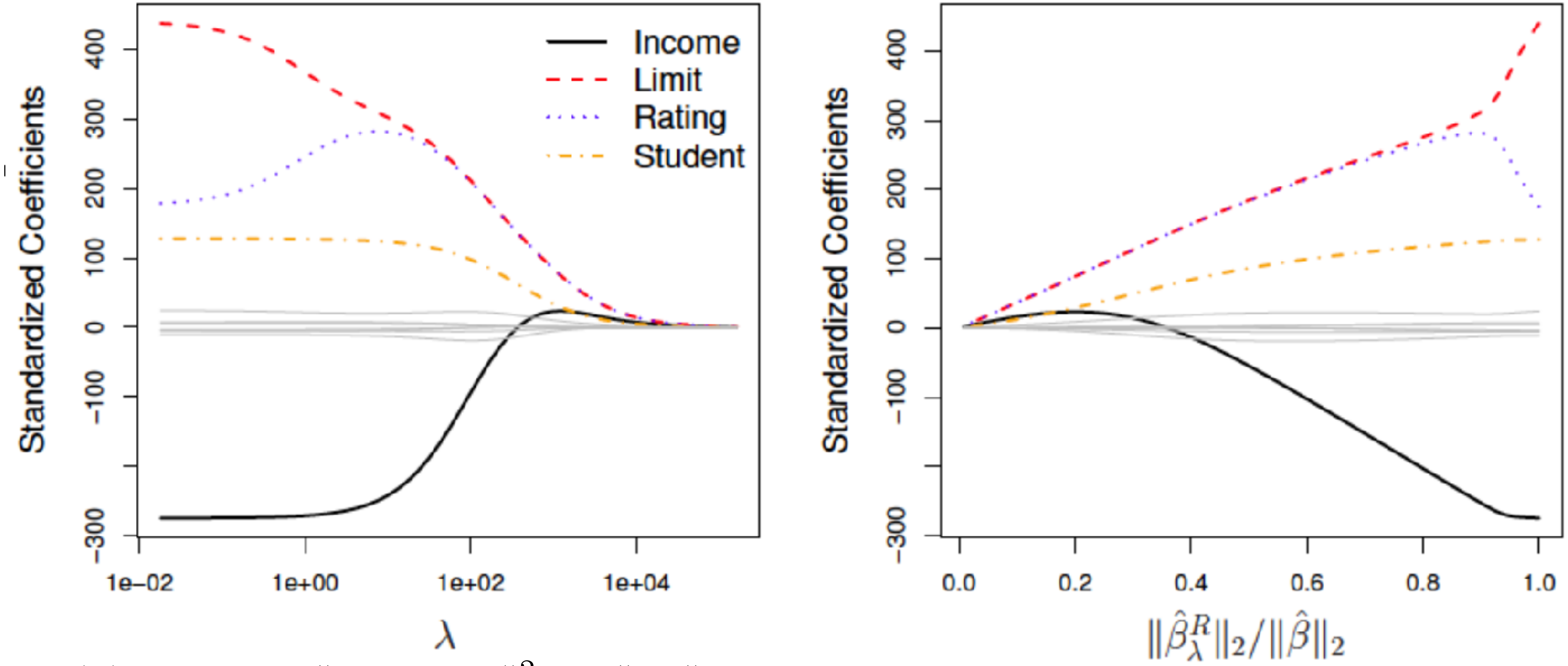
Regularized Least Squares



ridge regression has a circular constraint with no sharp points, this intersection will not generally occur on an axis, and so the ridge regression coefficient estimates will be exclusively non-zero.

However, the lasso constraint has corners at each of the axes, and so the ellipse will OFFEN intersect the constraint region at an axis.

Credit Data Example of Ridge regression



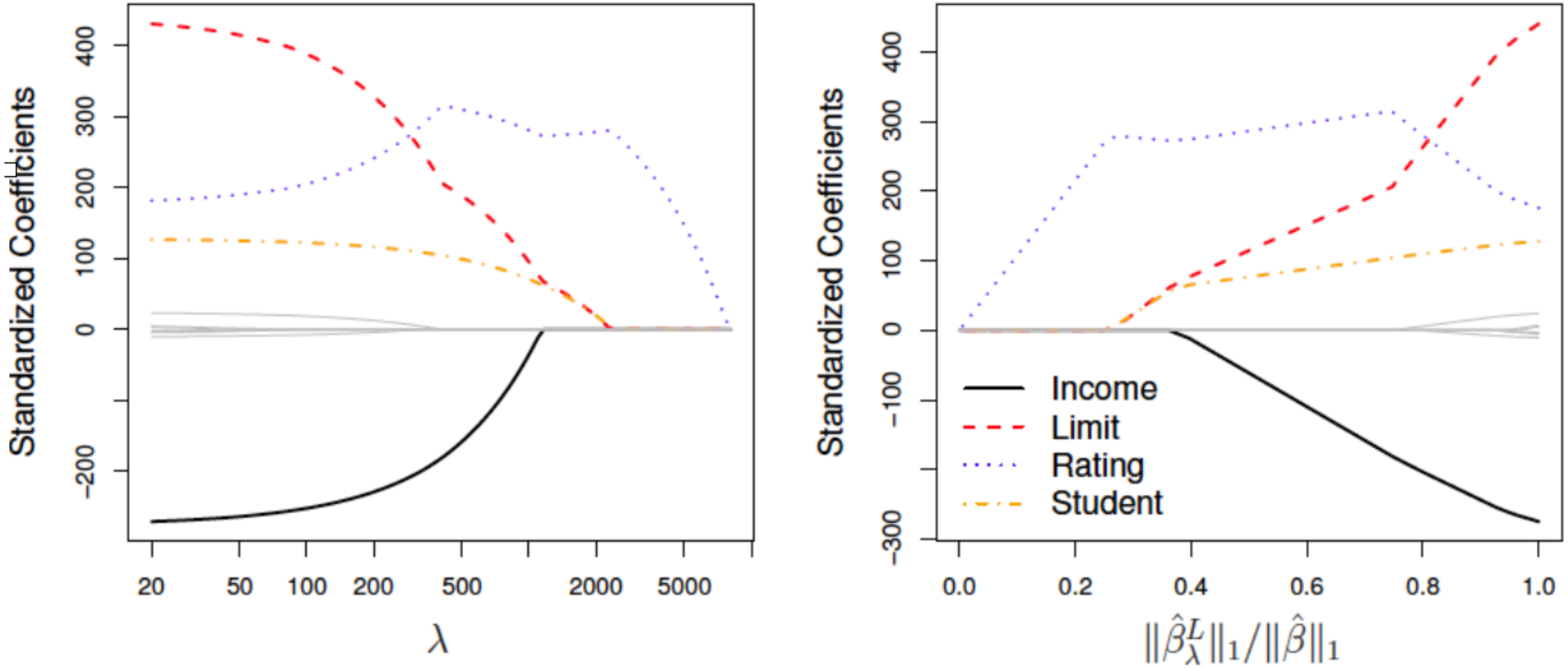
$$\beta(\lambda) = \underset{\beta}{\operatorname{argmin}} \parallel Y - X\beta \parallel_2^2 + \lambda \parallel \beta \parallel_2$$

Therefore, it is best to apply ridge regression after *standardizing the predictors*, using the formula

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

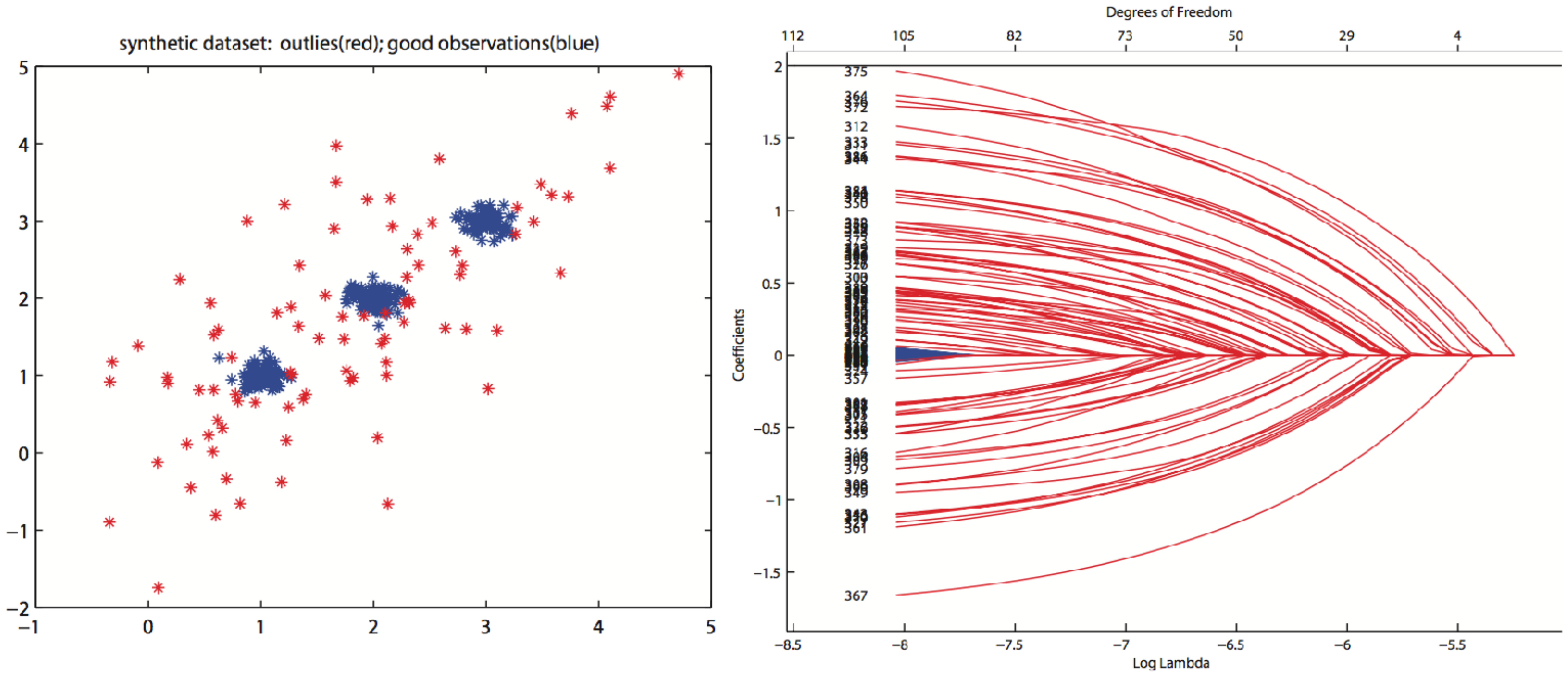
The right-hand panel displays the same ridge coefficient estimates as the left-hand panel, but instead of displaying λ on the x-axis, we now display $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$, where $\hat{\beta}$ denotes the vector of least squares coefficient estimates.

Credit Data Example of Lasso



- However, in the case of the lasso, the L_1 penalty has the effect of forcing some of the coefficient estimates to be exactly equal to zero when the tuning parameter is sufficiently large.
- much like best subset selection, the lasso performs variable selection.
- We say that the lasso yields sparse models I that is, models that involve only a subset of the variables.

Lasso for Outlier Detection by Checking Regularisation Path



Red lines & red points indicate outliers; Blue lines & blue points are inliers. Figures from [3].

[3] **Yanwei Fu**, De-An Huang, Leonid Sigal, Robust Classification by Pre-conditioned LASSO and Transductive Diffusion Component Analysis,http://arxiv.org/abs/1511.06340

Chap 2 - Linear Regression(2)

Linear Model Selection and Regularisation

1.advanced topics

Alternatives to Squared Error

Huber M-estimator:

$$\min J_h(\Theta) = \rho_\lambda(\delta_0\Theta - Y) \tag{14}$$

where the Huber's loss function $\rho_\lambda(x)$ is defined as

$$\rho_\lambda(x) = \begin{cases} x^2/2, & \text{if } |x| \le \lambda \\ \lambda|x| - \lambda^2/2, & \text{if } |x| > \lambda. \end{cases}$$

Appendix

Gradient Checking

Optional subtitle

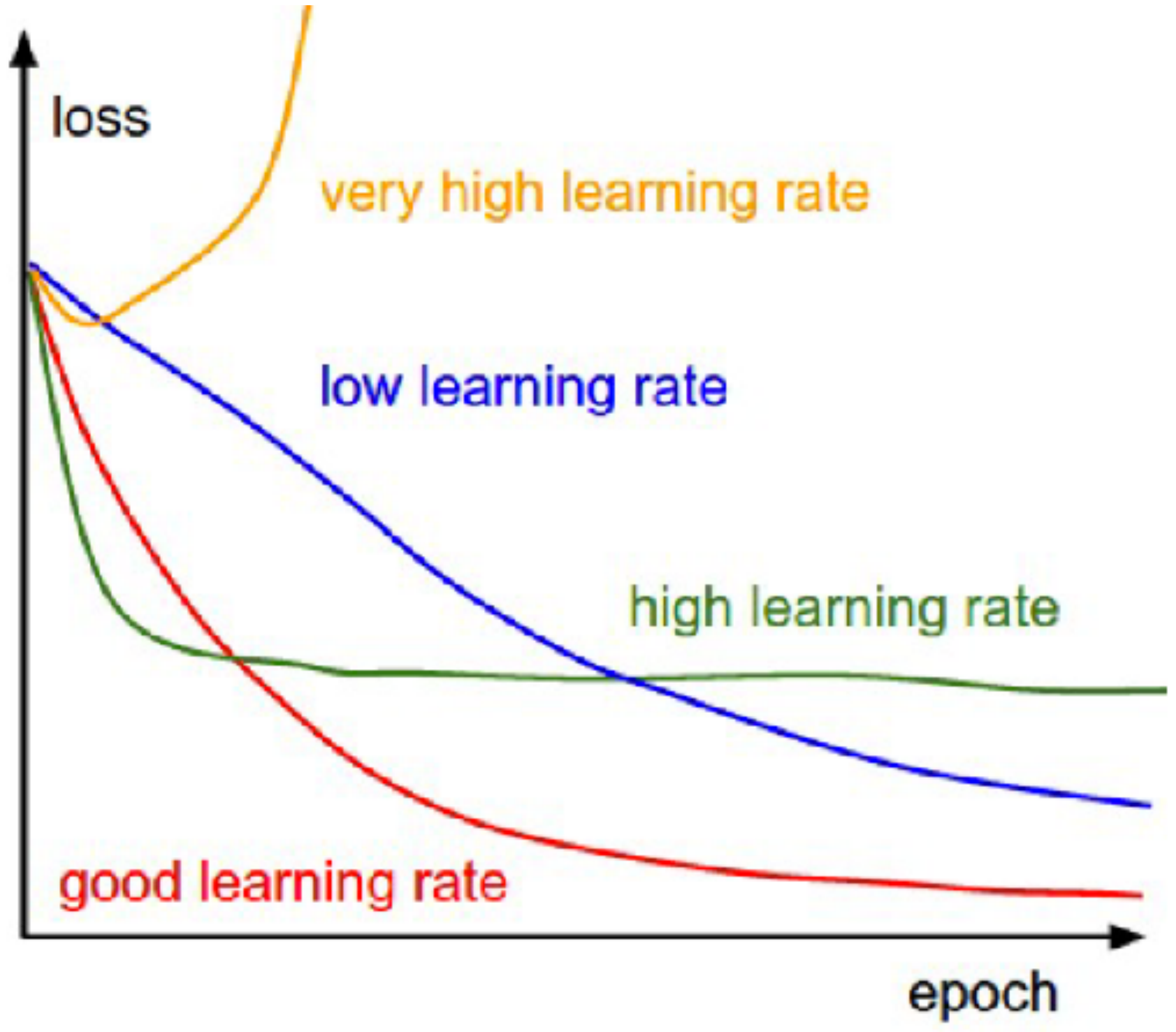
- When implementing the gradient computation for machine learning models, it's often difficult to know if our implementation of f and ∇f is correct.
- We can use finite-differences approximation to the gradient to help:

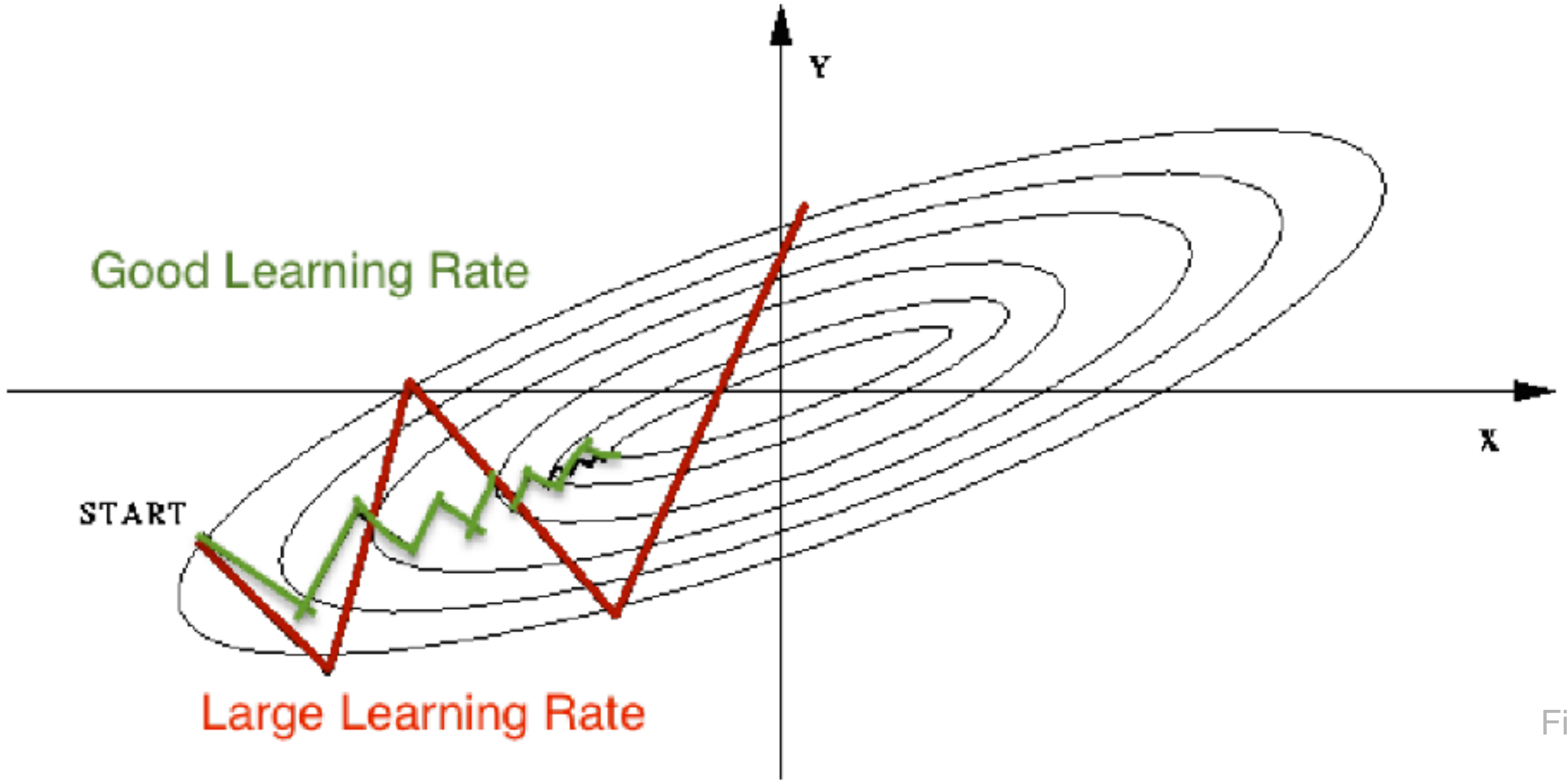
$$\frac{\partial f}{\partial \theta_i} \approx \frac{f((\theta_1, \ldots, \theta_i + \epsilon, \ldots, \theta_n)) - f((\theta_1, \ldots, \theta_i - \epsilon, \ldots, \theta_n))}{2\epsilon}$$

Why don't we always just use the finite differences approximation?

- slow: we need to recompute f twice for each parameter in our model.
- numerical issues

Learning Rate





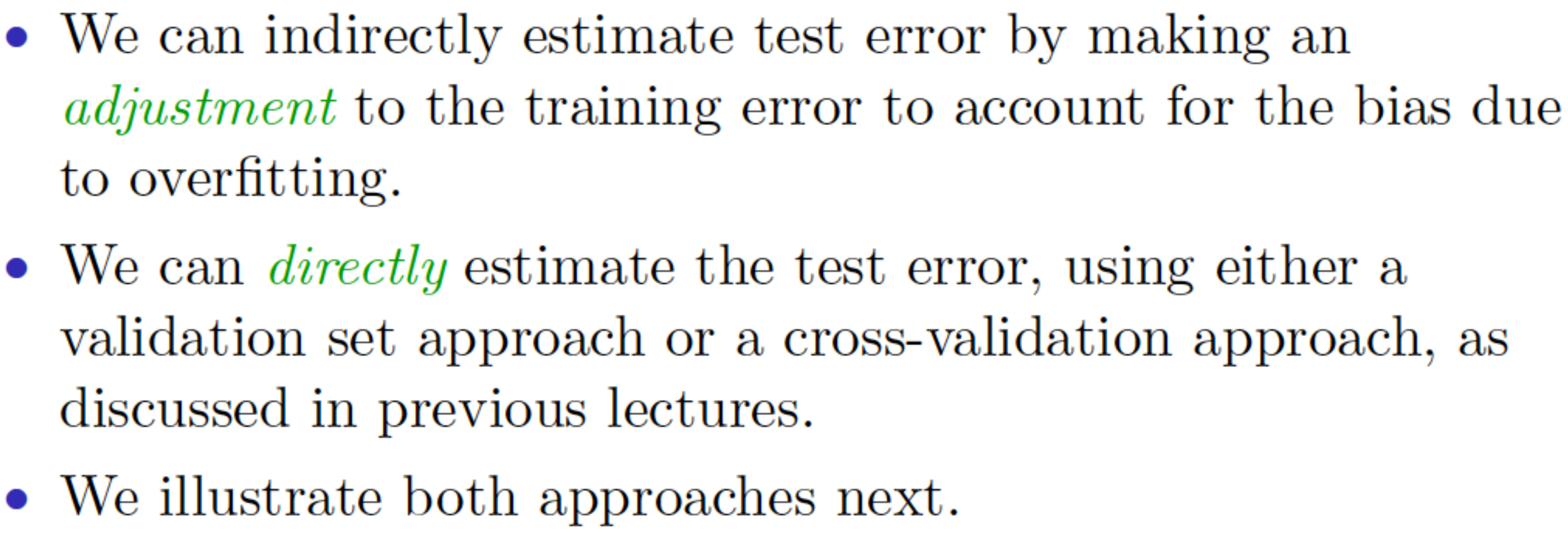
Stopping Rules of Optimisation Algorithms

- Change in objective function value is close to zero: $|f(\theta_{t+1}) - f(\theta_t)| < \epsilon$
- Gradient norm is close to zero: $\|\nabla_\theta f\| < \epsilon$
- Validation error starts to increase (this is called *early stopping*)

First image taken from Andrej Karpathy's Stanford Lectures, second image taken from Wikipedia

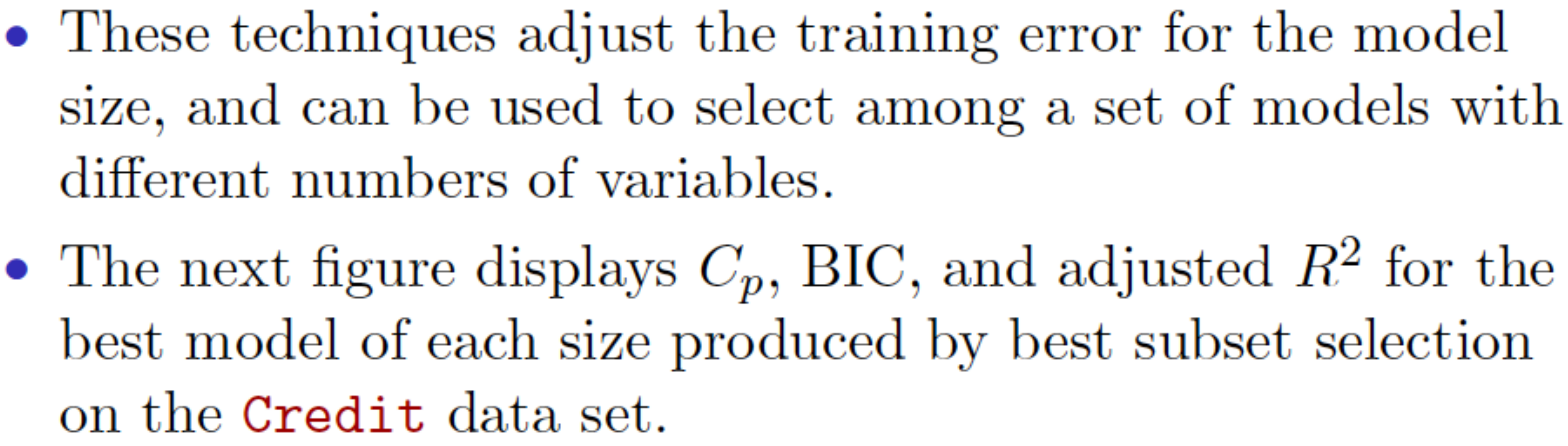
Estimating test error: two approaches

Optional subtitle



- We can indirectly estimate test error by making an *adjustment* to the training error to account for the bias due to overfitting.
- We can *directly* estimate the test error, using either a validation set approach or a cross-validation approach, as discussed in previous lectures.
- We illustrate both approaches next.

C_p, AIC, BIC, and Adjusted R^2



- These techniques adjust the training error for the model size, and can be used to select among a set of models with different numbers of variables.
- The next figure displays C_p, BIC, and adjusted R^2 for the best model of each size produced by best subset selection on the `Credit` data set.

Details

Optional subtitle

- *Mallow's C_p*:

$$C_p = \frac{1}{n}\left(\text{RSS} + 2d\hat{\sigma}^2\right),$$

where d is the total # of parameters used and $\hat{\sigma}^2$ is an estimate of the variance of the error ϵ associated with each response measurement.

- The *AIC* criterion is defined for a large class of models fit by maximum likelihood:

$$\text{AIC} = -2\log L + 2 \cdot d$$

where L is the maximized value of the likelihood function for the estimated model.

- In the case of the linear model with Gaussian errors, maximum likelihood and least squares are the same thing, and C_p and AIC are equivalent. *Prove this.*

$$\text{BIC} = \frac{1}{n}\left(\text{RSS} + \log(n)d\hat{\sigma}^2\right).$$

- Like C_p, the BIC will tend to take on a small value for a model with a low test error, and so generally we select the model that has the lowest BIC value.
- Notice that BIC replaces the $2d\hat{\sigma}^2$ used by C_p with a $\log(n)d\hat{\sigma}^2$ term, where n is the number of observations.
- Since $\log n > 2$ for any $n > 7$, the BIC statistic generally places a heavier penalty on models with many variables, and hence results in the selection of smaller models than C_p. See Figure on slide 19.

Adjusted R^2

- For a least squares model with d variables, the adjusted R^2 statistic is calculated as
$$\text{Adjusted } R^2 = 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)}.$$
where TSS is the total sum of squares.
- Unlike C_p, AIC, and BIC, for which a *small* value indicates a model with a low test error, a *large* value of adjusted R^2 indicates a model with a small test error.
- Maximizing the adjusted R^2 is equivalent to minimizing $\frac{\text{RSS}}{n-d-1}$. While RSS always decreases as the number of variables in the model increases, $\frac{\text{RSS}}{n-d-1}$ may increase or decrease, due to the presence of d in the denominator.
- Unlike the R^2 statistic, the adjusted R^2 statistic *pays a price* for the inclusion of unnecessary variables in the model. See Figure on slide 19.

Maximum Likelihood Estimation

Maximum likelihood estimate (MLE) in an abstract setting:

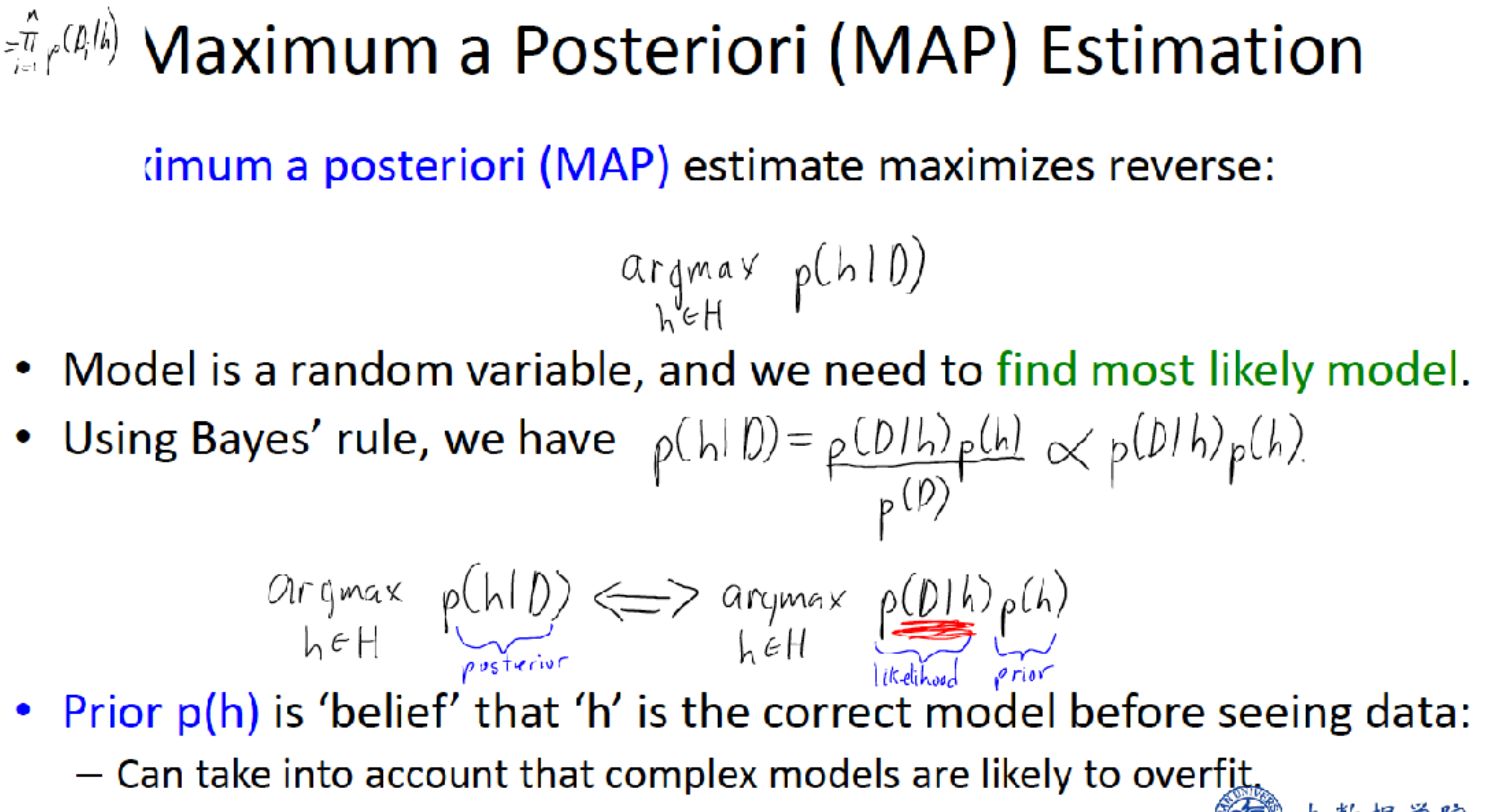
- We have a dataset ‘D’.
- We want to pick a model ‘h’ from among set of models H.
- We define the likelihood as a probability density p(D | h).
- We choose the model ‘h’ that maximizes the likelihood:

$$\boxed{\hat{h} = \underset{h \in H}{\operatorname{argmax}}\ p(D \mid h)}$$

- If the data consists of ‘n’ IID samples ‘D_i’, then we equivalently have:

$$\hat{h} = \underset{h \in H}{\operatorname{argmax}} \prod_{i=1}^{n} p(D_i \mid h)$$ Since independence implies $p(D \mid h) = \prod_{i=1}^{n} p(D_i \mid h)$

- MLE has appealing properties as n -> ∞ (take STAT 560/561)



Maximum a Posteriori (MAP) Estimation

ximum a posteriori (MAP) estimate maximizes reverse:

$$\underset{h \in H}{\operatorname{argmax}}\ p(h \mid D)$$

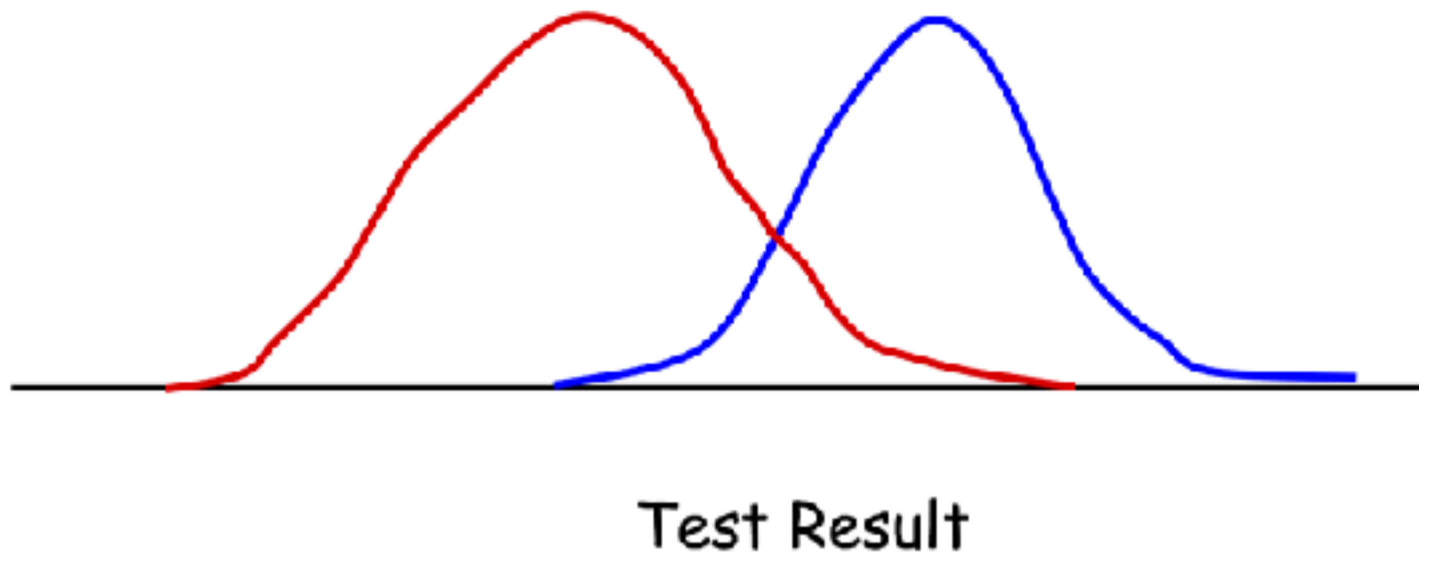
- Model is a random variable, and we need to find most likely model.
- Using Bayes’ rule, we have $p(h \mid D) = \frac{p(D \mid h)\, p(h)}{p(D)} \propto p(D \mid h)\, p(h)$

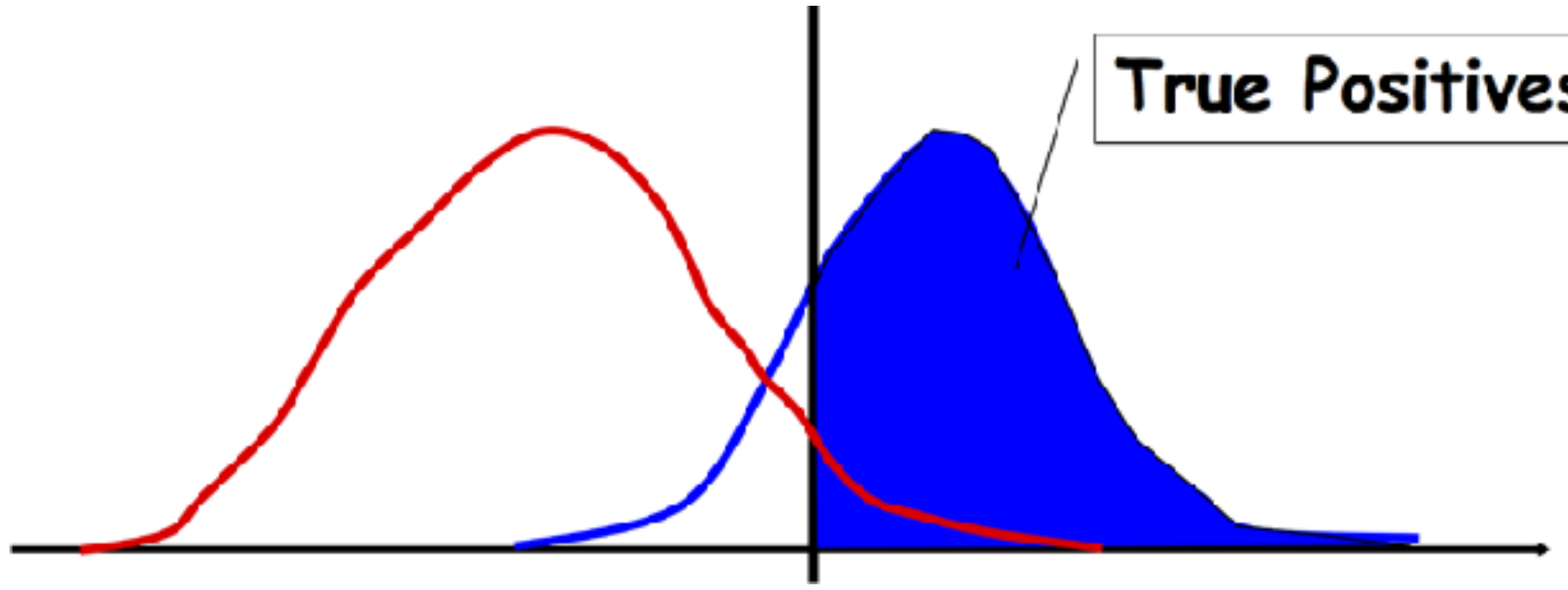
$$\underset{h \in H}{\operatorname{argmax}}\ \underbrace{p(h \mid D)}_{\text{posterior}} \Longleftrightarrow \underset{h \in H}{\operatorname{argmax}}\ \underbrace{p(D \mid h)}_{\text{likelihood}}\ \underbrace{p(h)}_{\text{prior}}$$

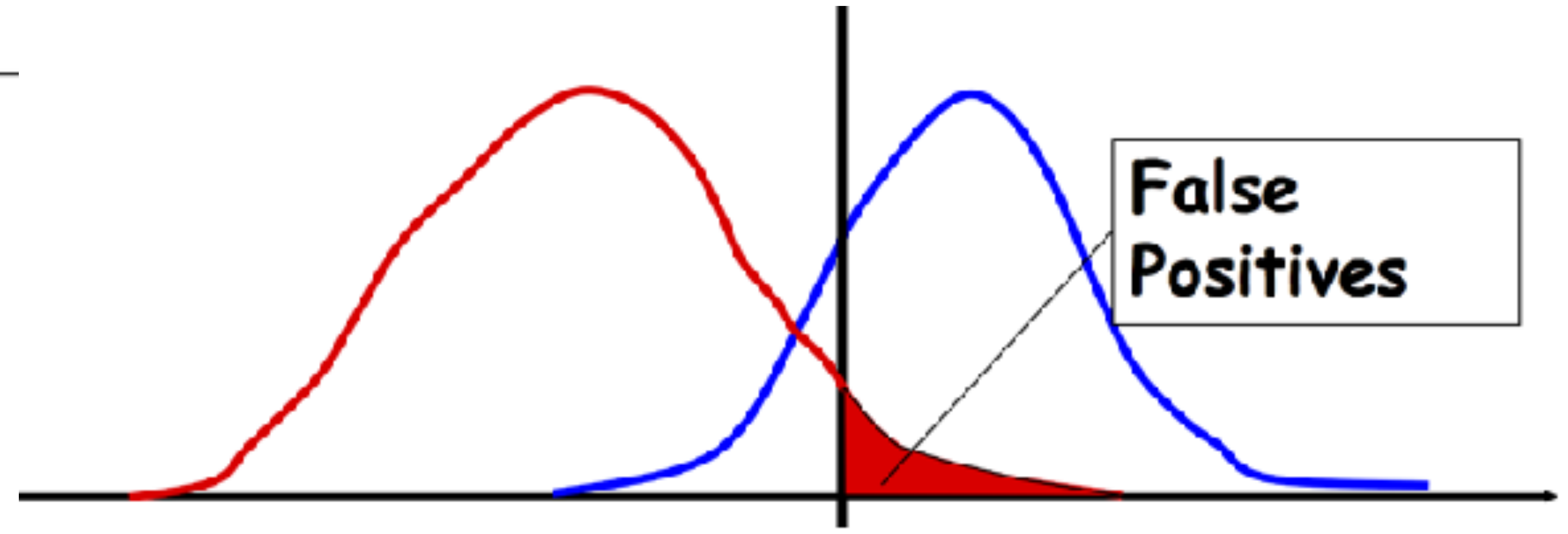
- Prior p(h) is ‘belief’ that ‘h’ is the correct model before seeing data:
  - Can take into account that complex models are likely to overfit.

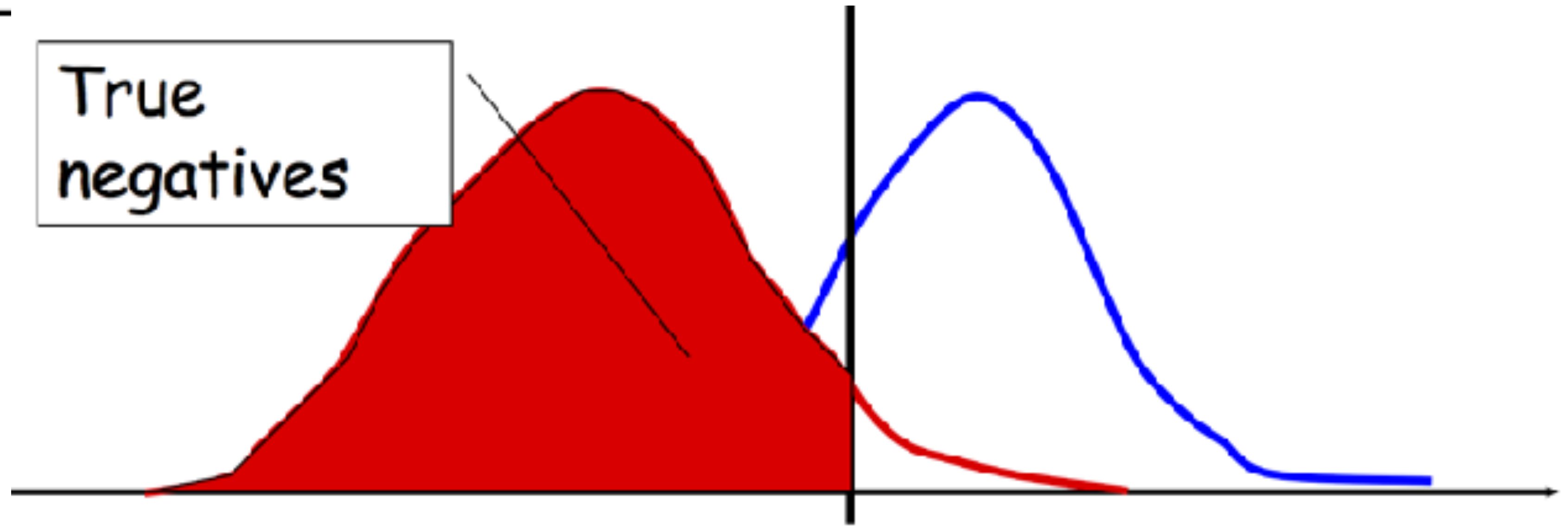
大数据学院 School of Data Science

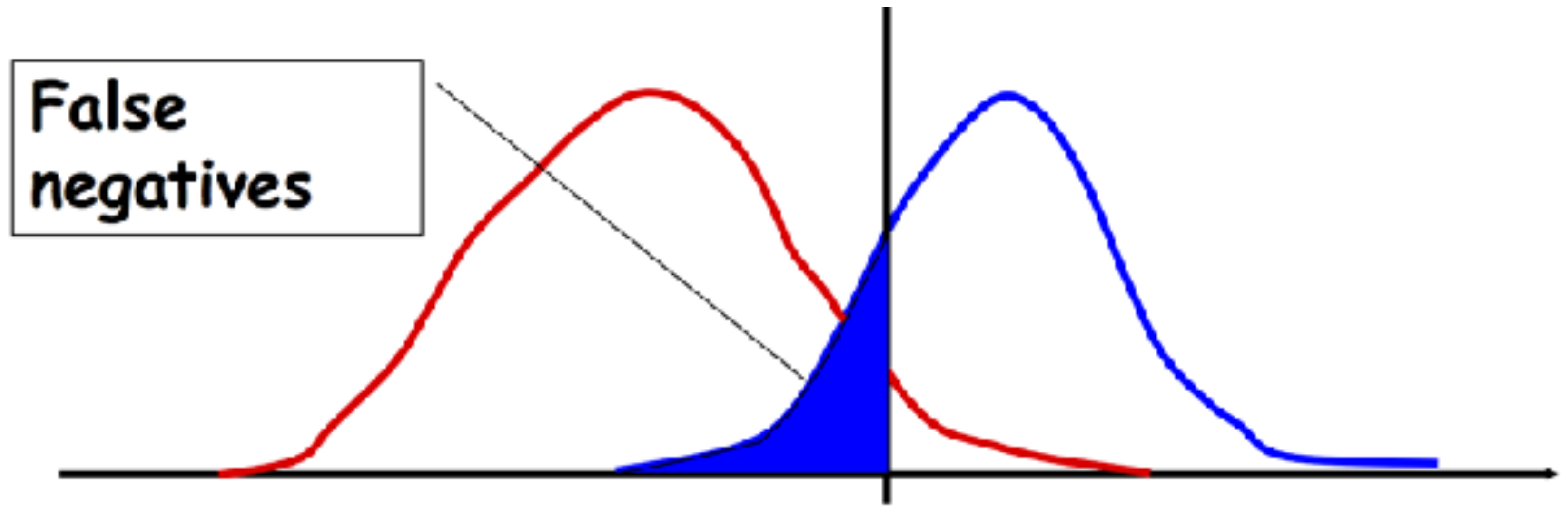
ROC

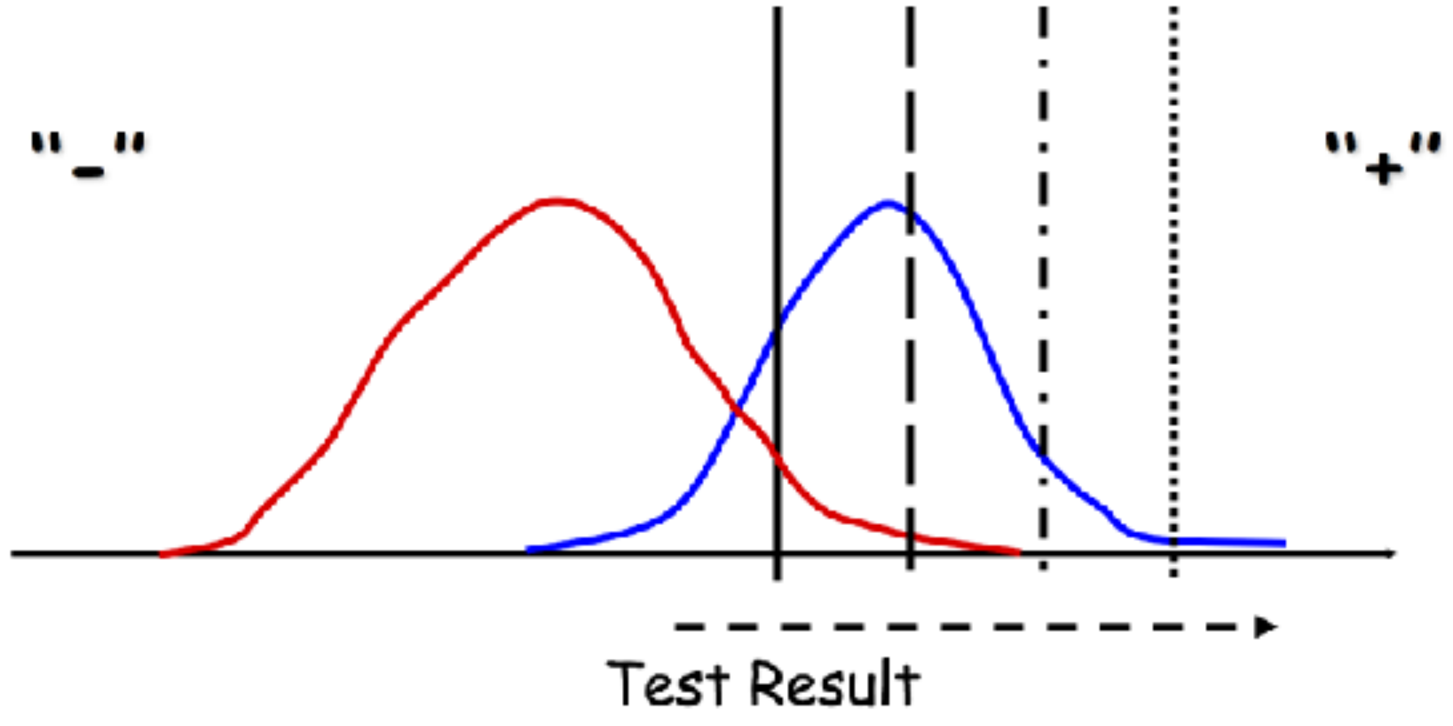

Test Result

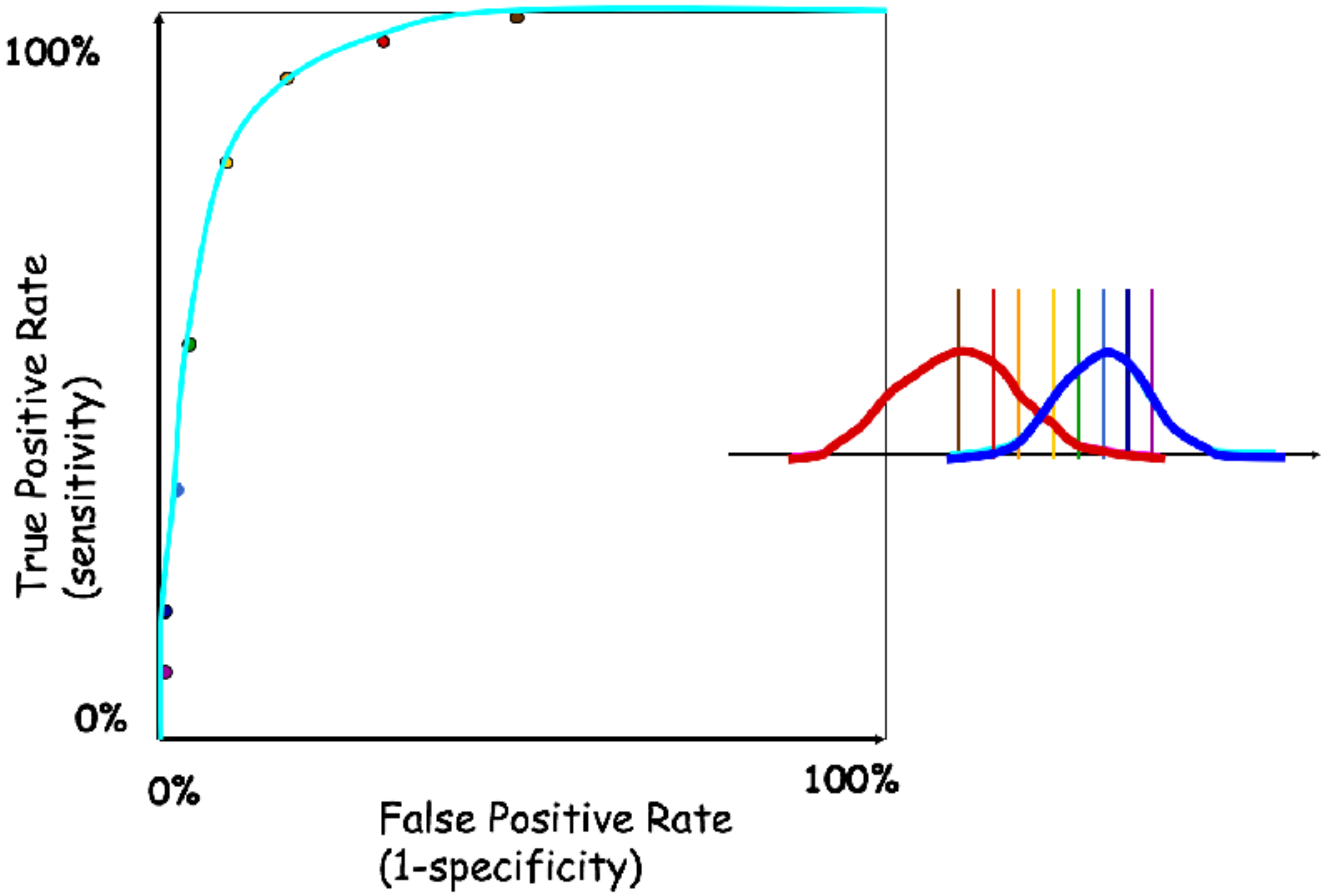

True Positives


False Positives


True negatives


False negatives


"-"
"+"
Test Result


100%
0%
True Positive Rate (sensitivity)
0%
100%
False Positive Rate (1-specificity)