Bias/Variance Tradeoff

Regularized Regression

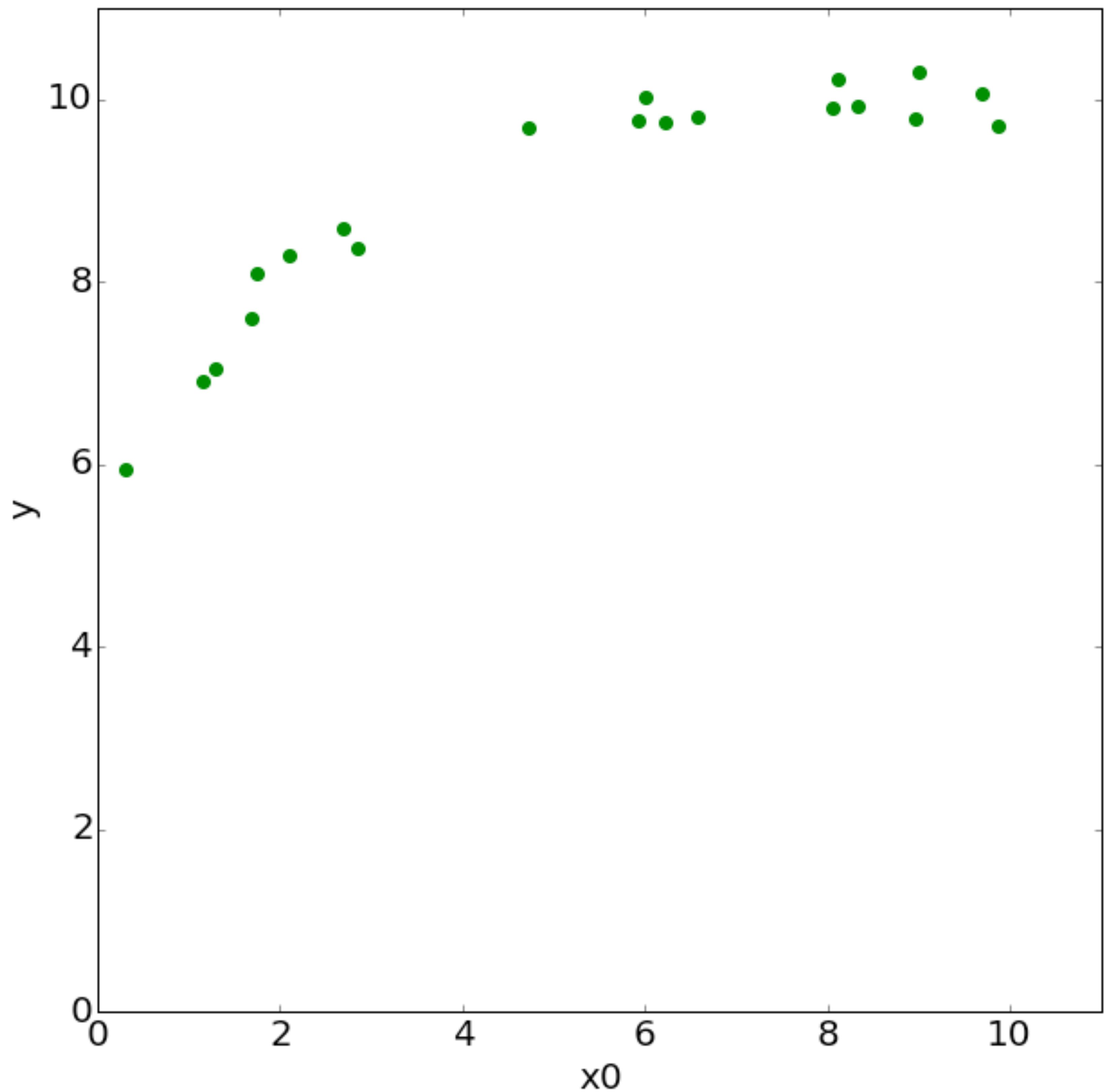
Outline

- Some Shortcomings of Ordinary Linear Regression
- Terms for discussing model error: Under/Over-fitting and Bias/Variance
- Ridge/Lasso: tools for balancing these error types
- When to use Ridge, when to use Lasso

Linear Regression Example

- Data: 20 samples x 10 features
- Predict: y

y	x0	x1	x2	x3	...
9.92	8.33	69.39	578.0	4815.	...
8.58	2.69	7.26	19.54	52.64	...
8.07	1.75	3.06	5.35	9.36	...
8.29	2.11	4.46	9.41	19.86	...
...	...	...	...	...	...



10
8
6
4
2
0
y
0
2
4
6
8
10
x0

Recall: Linear Regression

Assume: $$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \epsilon$$

Noise term taken to be ~ N(0, σ^2)

Generate Model Fits: $$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^{p} X_j \hat{\beta}_j$$

or equivalently in matrix form:

$$\hat{Y} = X^T \hat{\beta}$$

(by adding zeroth term for intercept: β_0)

Recall: Linear Regression

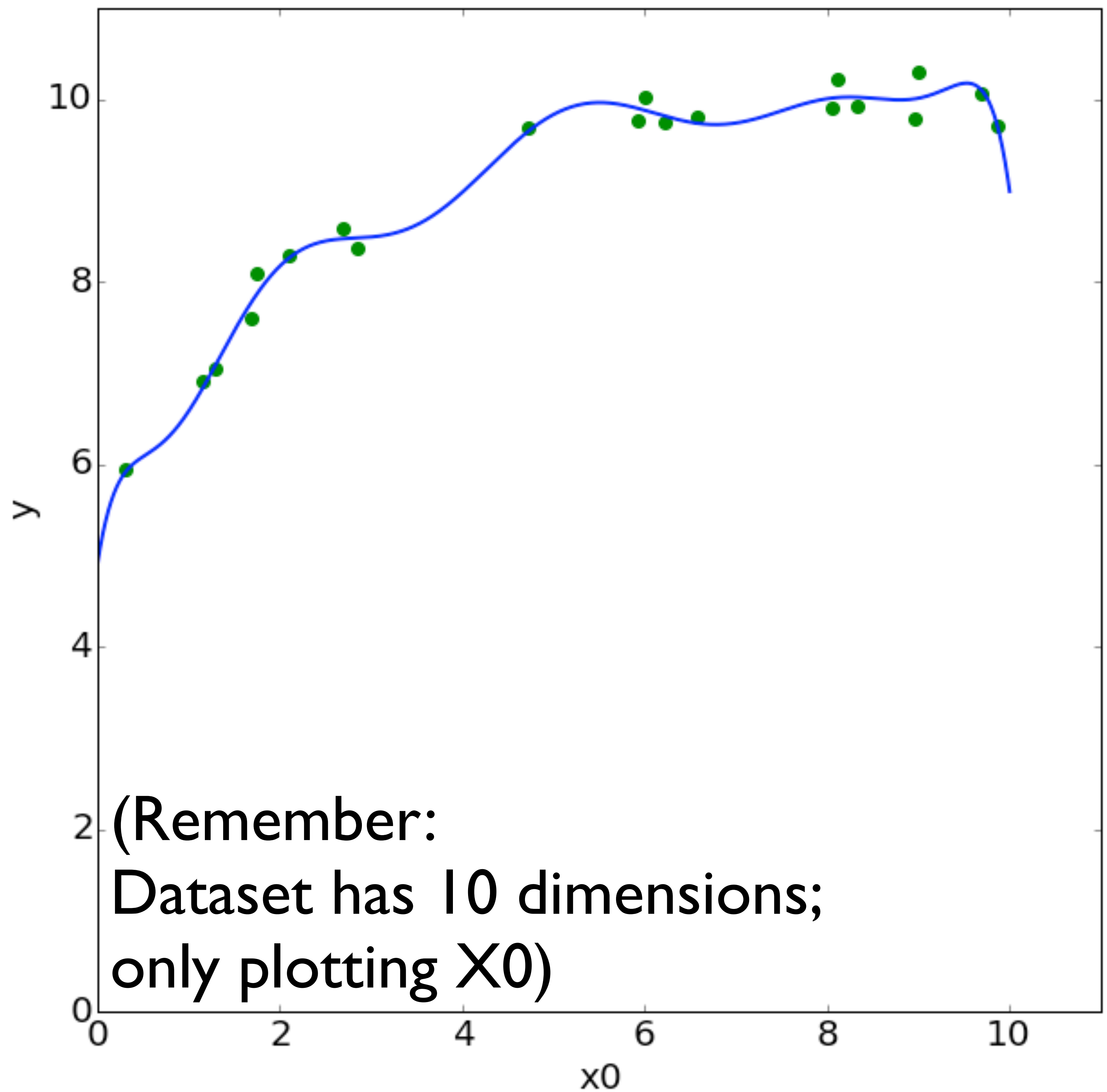
Minimize:

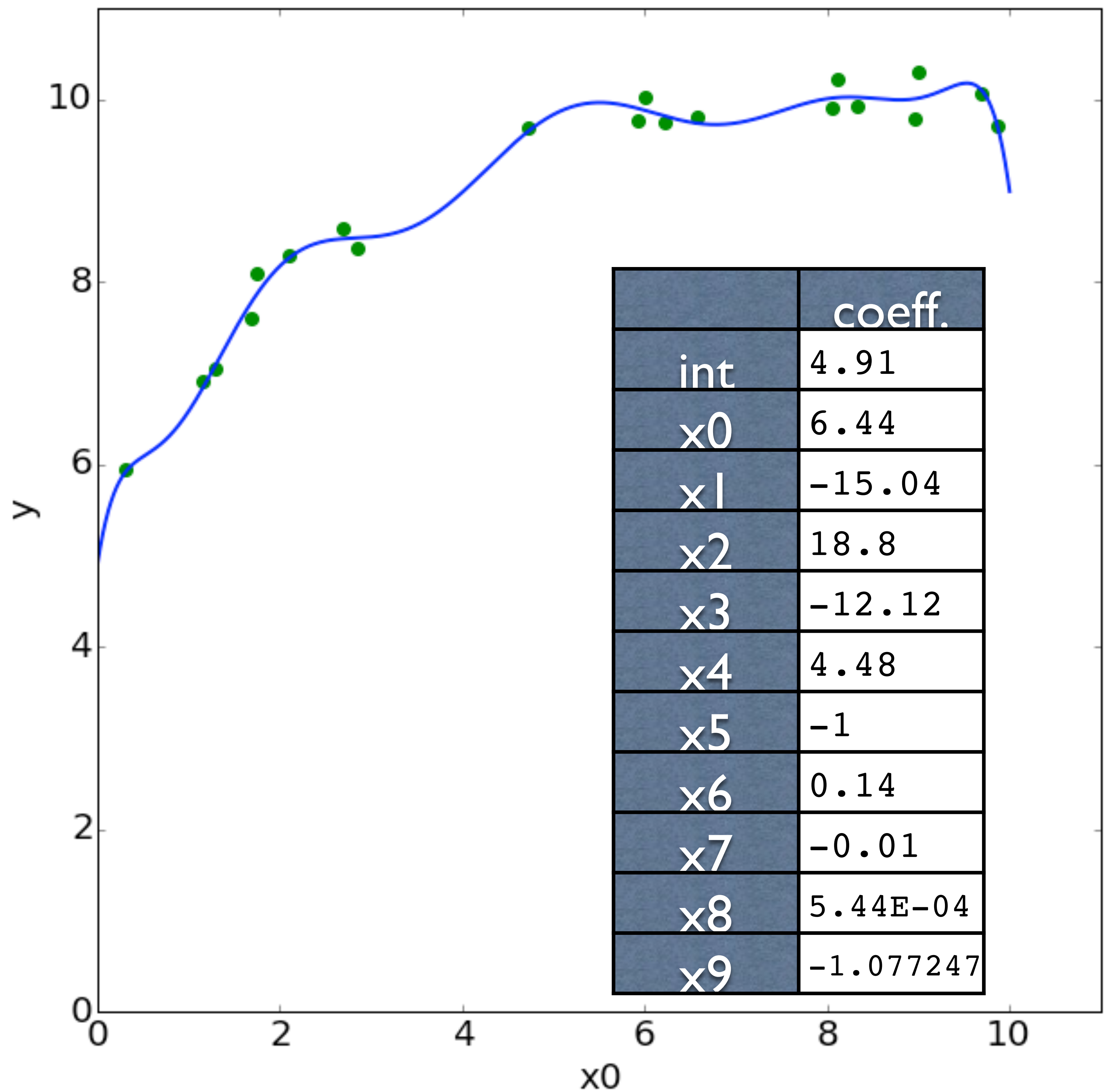
$$\begin{aligned}\mathrm{RSS}(\beta) &= \sum_{i=1}^{N}(y_i - f(x_i))^2 \\ &= \sum_{i=1}^{N}\Big(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\Big)^2\end{aligned}$$

matrix version:

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

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

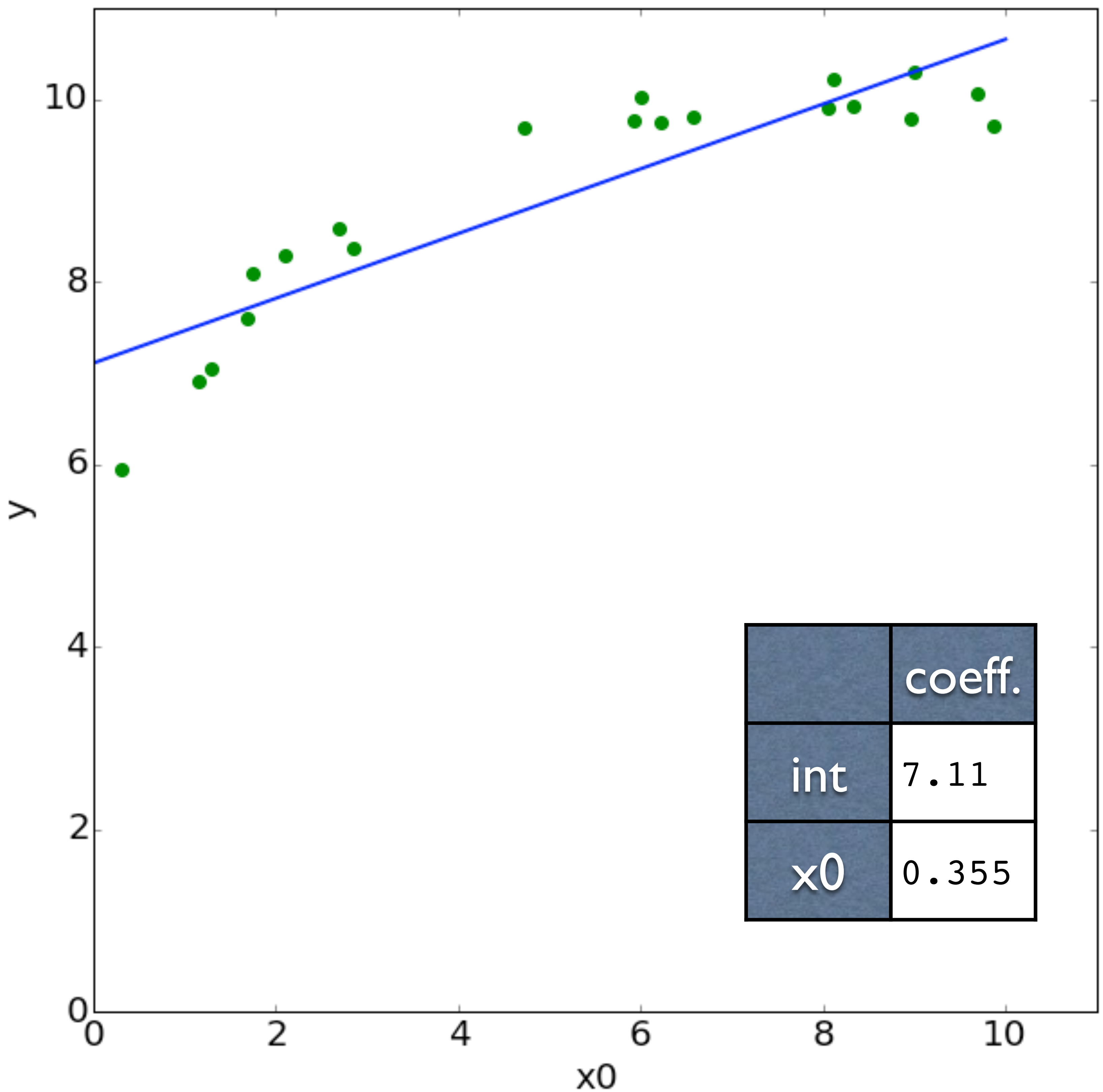

y
x0
0
2
4
6
8
10
(Remember:
Dataset has 10 dimensions;
only plotting X0)



	coeff.
int	4.91
x0	6.44
x1	-15.04
x2	18.8
x3	-12.12
x4	4.48
x5	-1
x6	0.14
x7	-0.01
x8	5.44E-04
x9	-1.077247

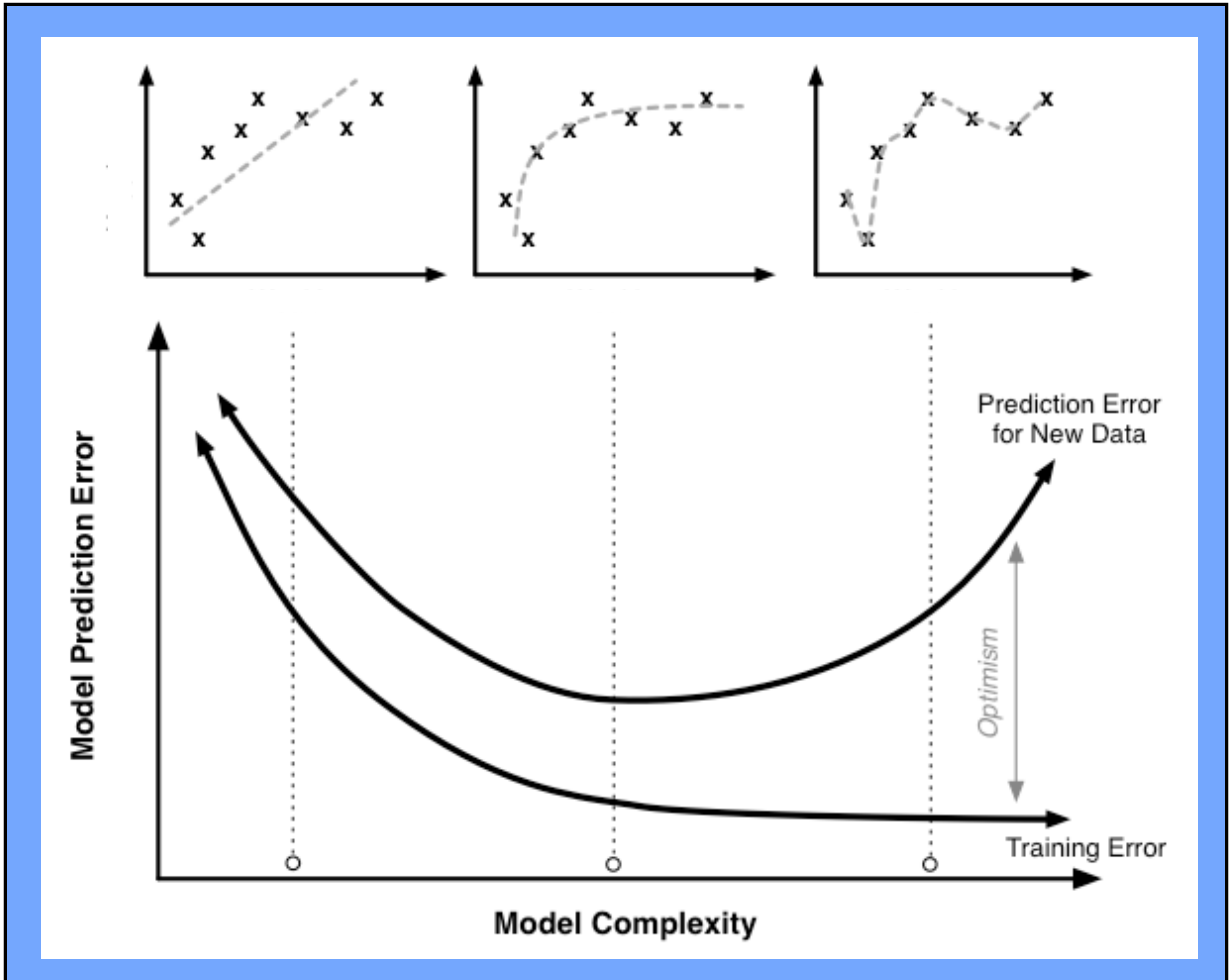
Subset Selection

- From this morning: Iteratively check combinations of features to eliminate some.
  - In this toy example, let's take it to extremes and eliminate all our features except x0…



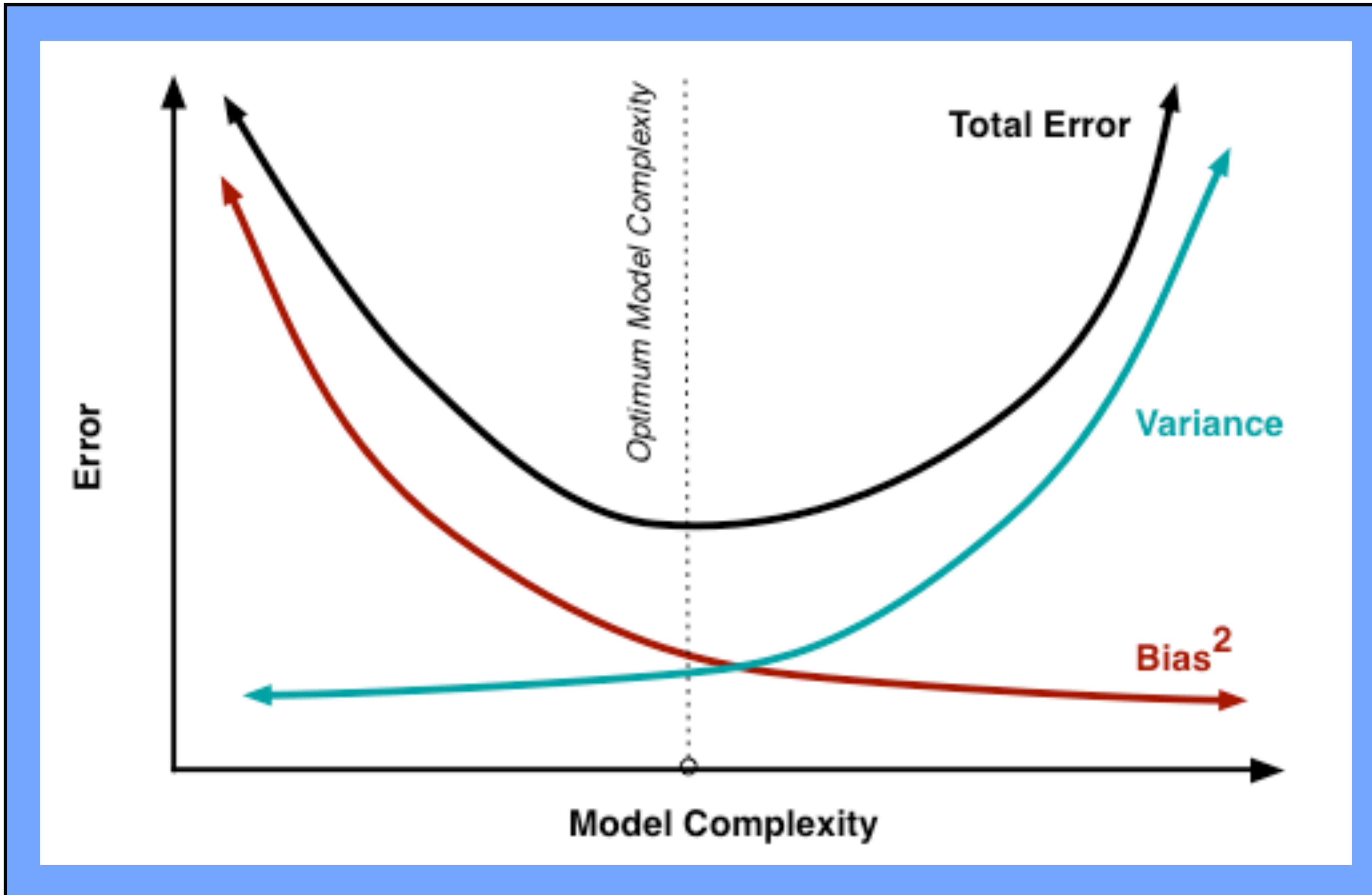
	coeff.
int	7.11
x0	0.355

Under/Over-Fitting



Source: scott.fortmann-roe.com

Bias/Variance



Source: scott.fortmann-roe.com

Bias/Variance

Trained model: $\hat{y}(x)$

Assuming: $y = f(x) + \epsilon$

where f(x) is the "ground truth"

Noise: $\epsilon \sim N(0, \sigma^2)$

New observation: $x^\star$, $y^\star = f(x^\star) + \epsilon$

Want to understand: $E[(\hat{y}(x^\star) - y^\star)^2]$

Decomposition of Expected Error

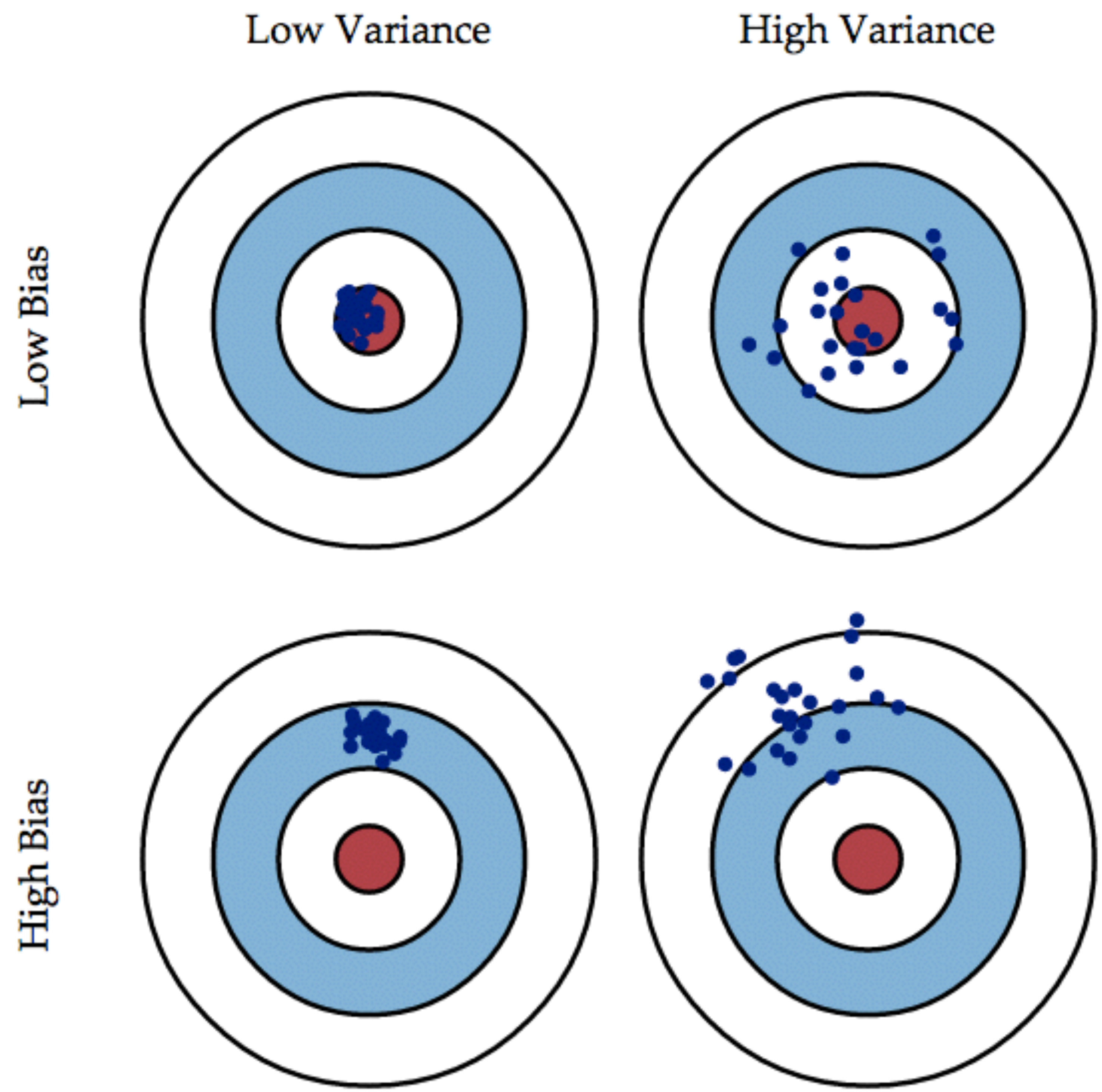
$$E[(\hat{y}(x^\star) - y^\star)^2] = E[(\hat{y}(x^\star) - \overline{\hat{y}(x^\star)})^2]$$ variance

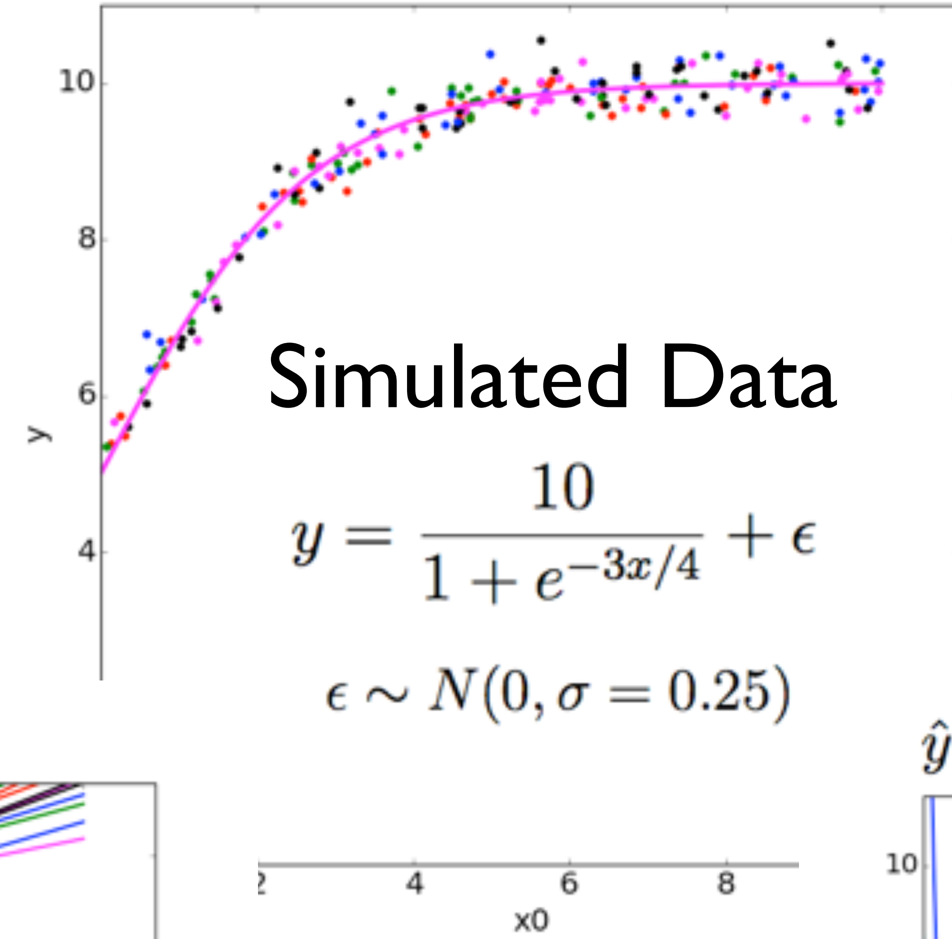
$$+(\overline{\hat{y}(x^\star)} - f(x^\star))^2$$ bias^2

$$+E[(y^\star - f(x^\star))^2]$$ noise^2

By: $\overline{Z} = E[Z]$

$$E[(Z - \overline{Z})^2] = E[Z^2] - \overline{Z}^2$$

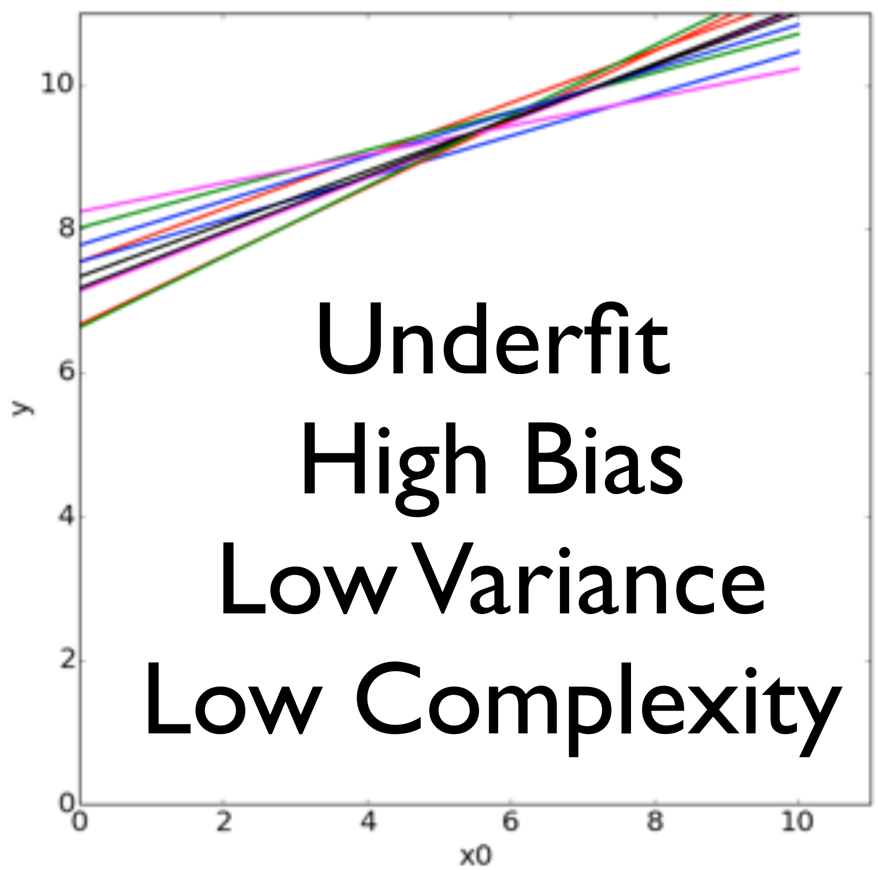

Low Variance
High Variance
Low Bias
High Bias



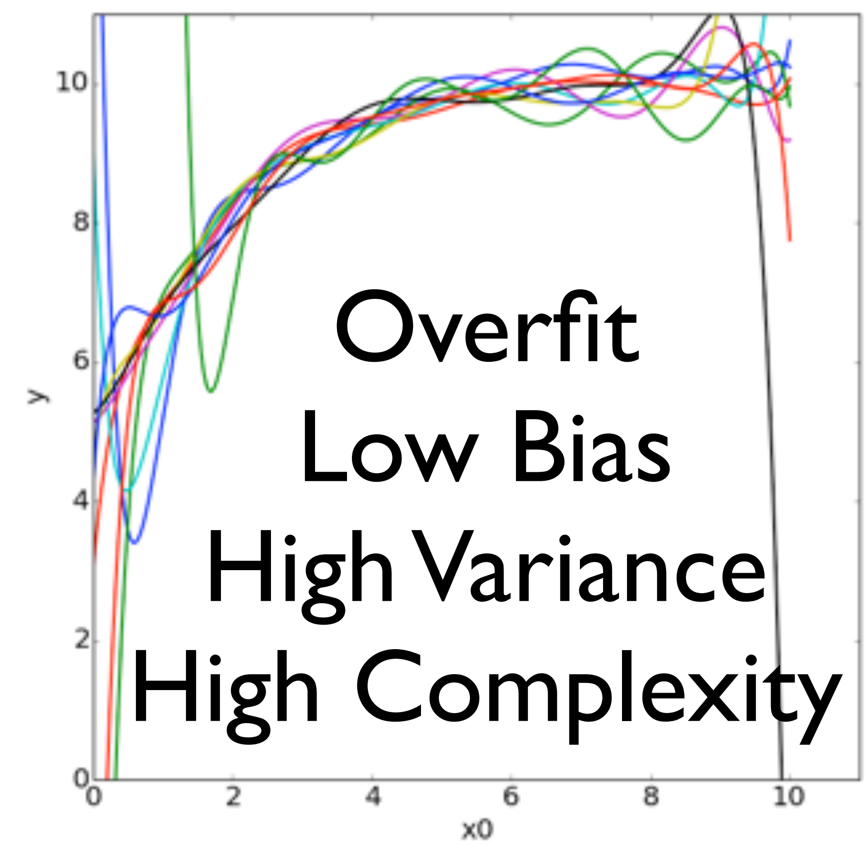
$$y = \frac{10}{1 + e^{-3x/4}} + \epsilon$$

$$\epsilon \sim N(0, \sigma = 0.25)$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$



$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \ldots + \hat{\beta}_{10} x^{10}$$



Bias/Variance Tradeoff

$$Var(\hat{y}(x^\star)) = E[(\hat{y}(x^\star) - \overline{\hat{y}(x^\star)})^2]$$

- Amount by which $\hat{y}$ would change if we had estimated it using a different training set

$$Bias(\hat{y}(x^\star)) = E[(\hat{y}(x^\star)] - f(x^\star)$$

- Difference between expected prediction of our models and true value we are trying to predict

$$Var(\epsilon)$$

- Irreducible error, recall: $y = f(x) + \epsilon$

Ridge Regression

Minimize: $$\sum_{i=1}^{N}\left(y_i-\beta_0-\sum_{j=1}^{p}x_{ij}\beta_j\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

Matrix form: $$(\mathbf{y}-\mathbf{X}\beta)^T(\mathbf{y}-\mathbf{X}\beta)+\lambda\beta^T\beta$$

where λ is the regularization parameter

$$\boxed{\hat{\beta}^{\text{ridge}}=(\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}}$$

where *I* is the identity matrix

(ones on the diagonal, zeros elsewhere)

Ridge Regression

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

- Intuition: Feature selection equivalent to setting some ß to zero. Instead Ridge imposes a penalty on high (squared) ß-values, to favor ones that are near zero.

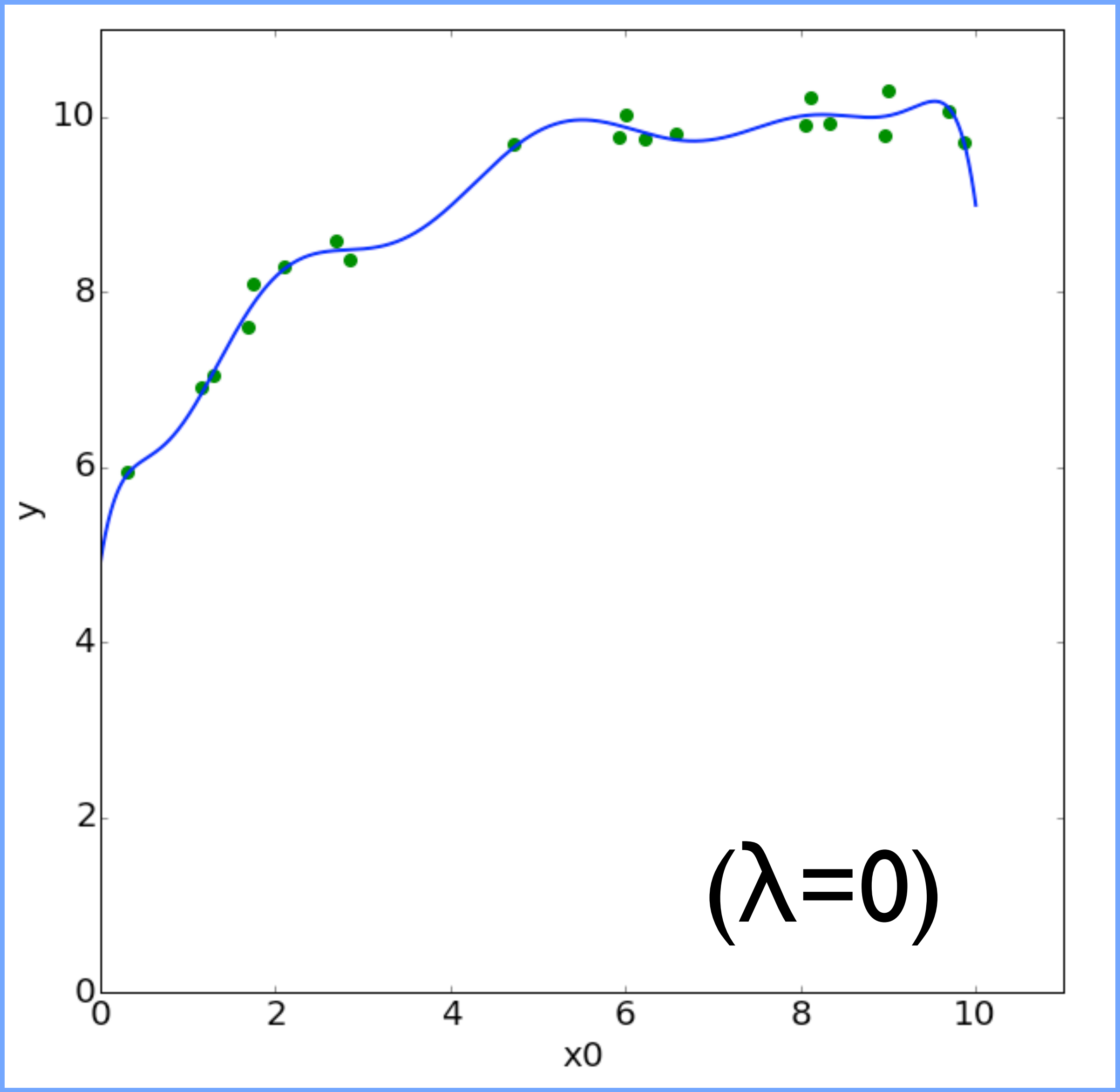
Ridge Regression

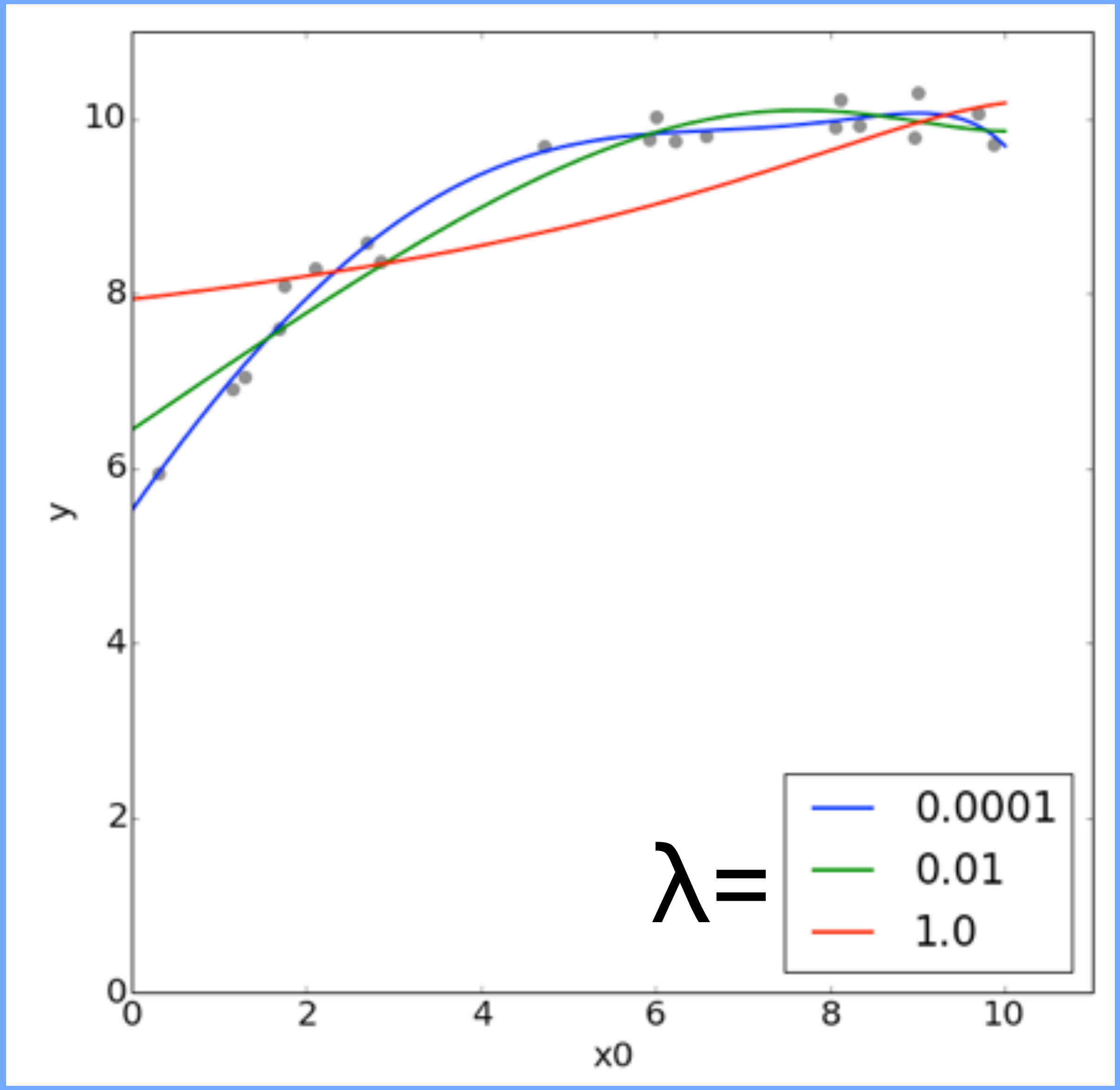
$$\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- Amount of penalty set by $\lambda \geq 0$
- $\lambda = 0$: No Regularization
- $\lambda \rightarrow$ "high", $\beta_j \rightarrow 0$ (Note that β_0 is not penalized, so model will return mean observed y value, as the intercept in this case)

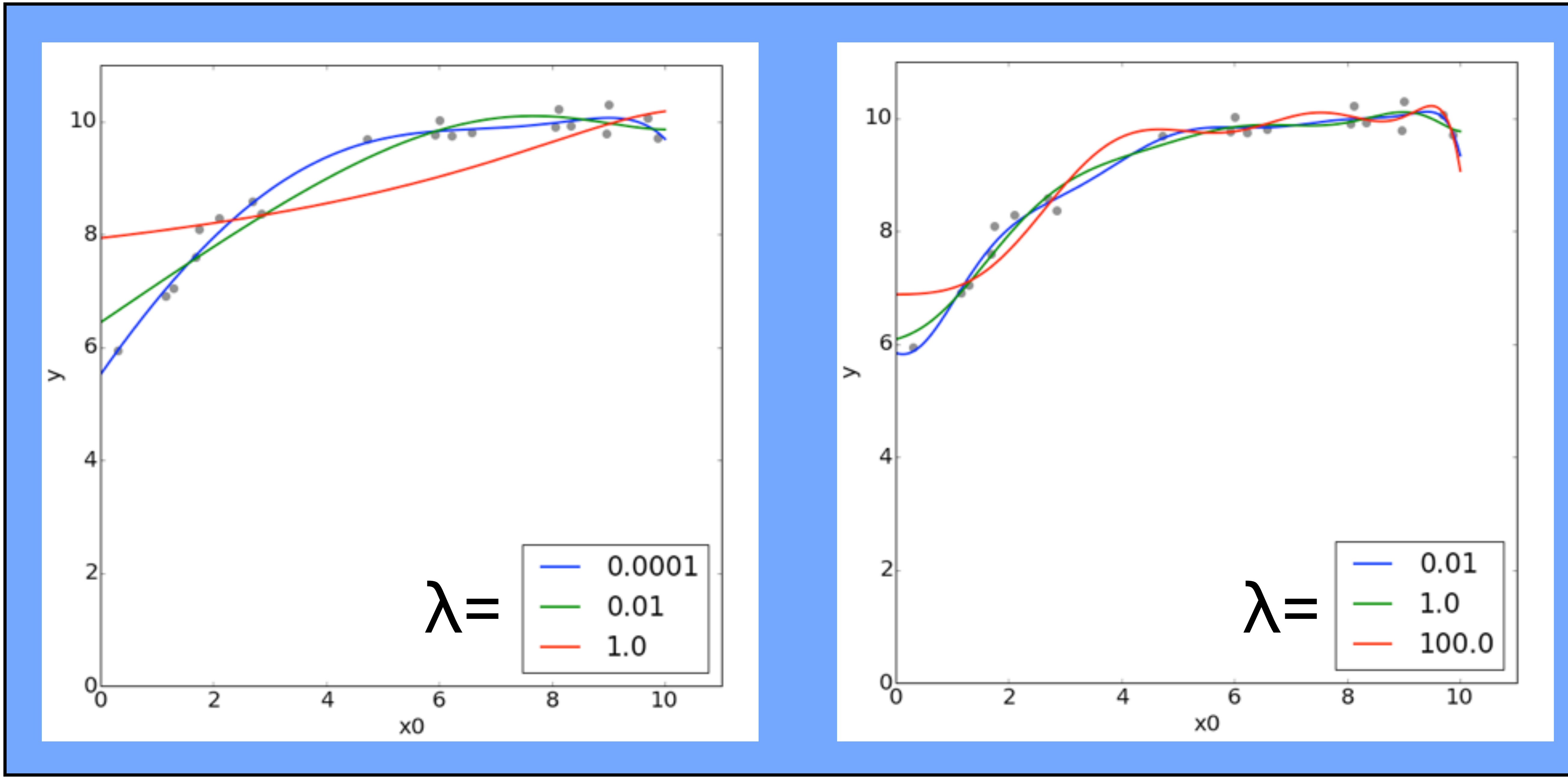
Linear Regression

Ridge Regression



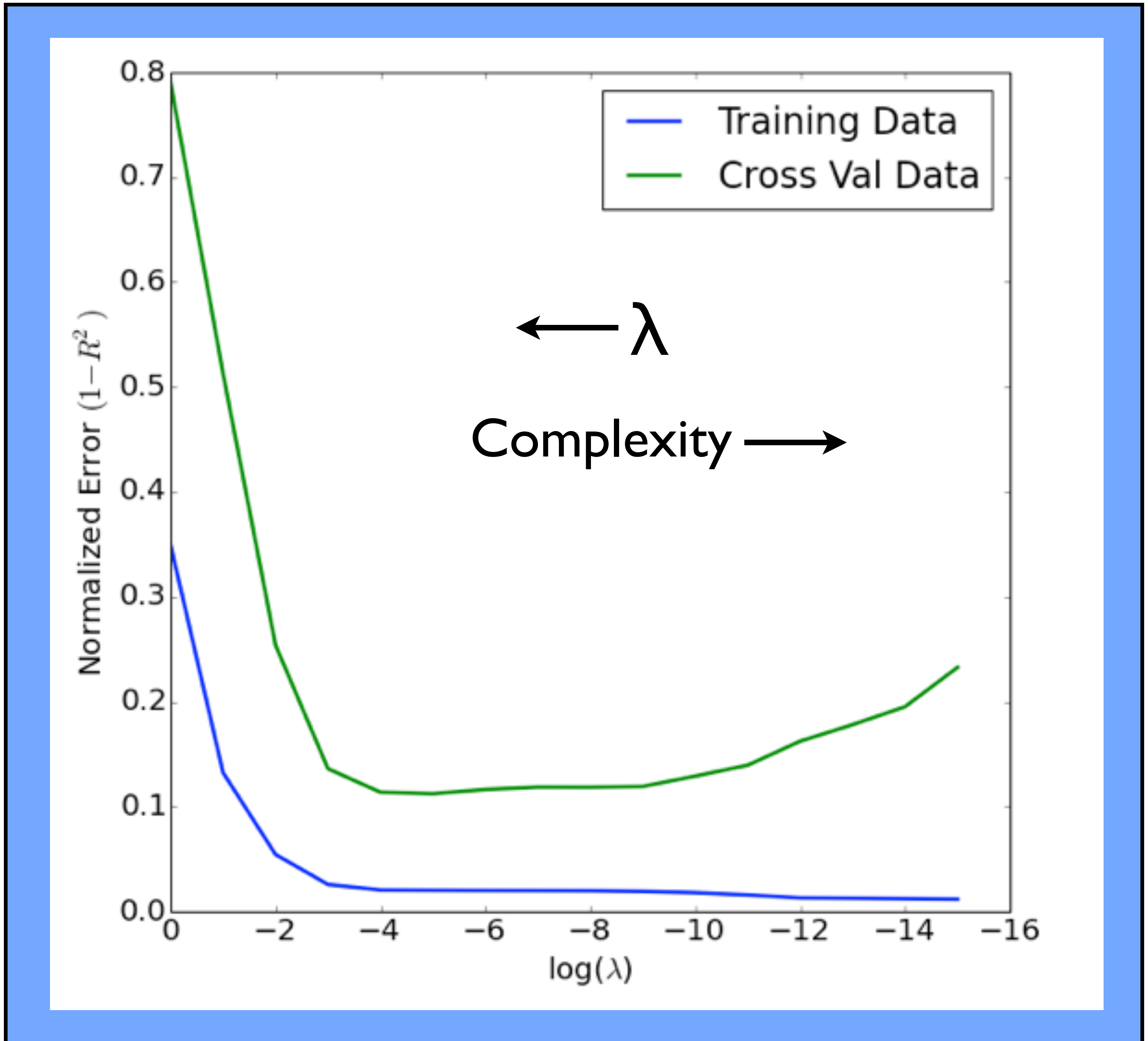


Normalized Data | Non-Normalized Data



Single value for λ assumes features are on the same scale!!

Pick λ with Cross Validation



Lasso Regression

Minimize: $$\sum_{i=1}^{N}\left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j\right)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

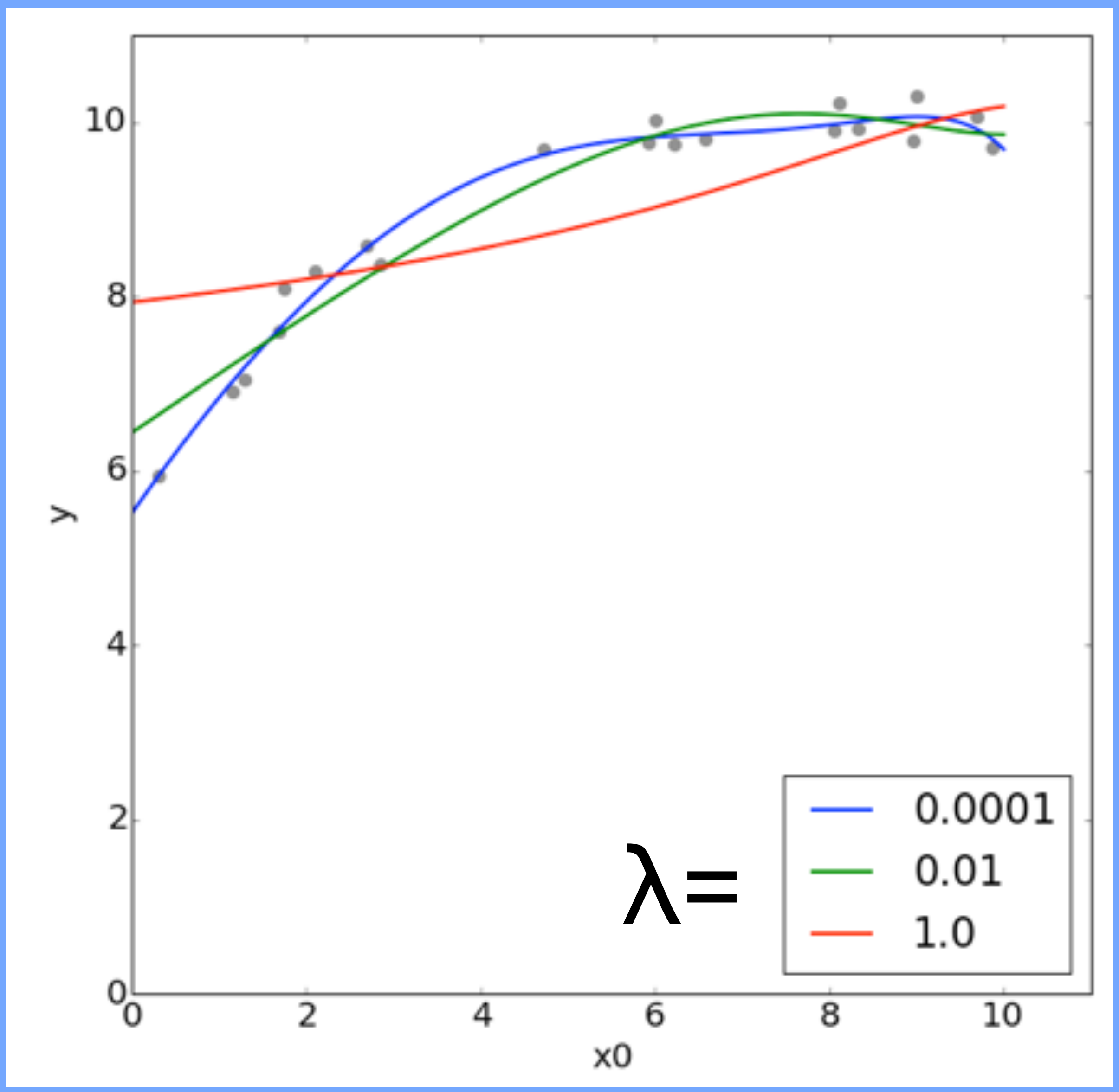
(No matrix form)

Same rules as Ridge apply:

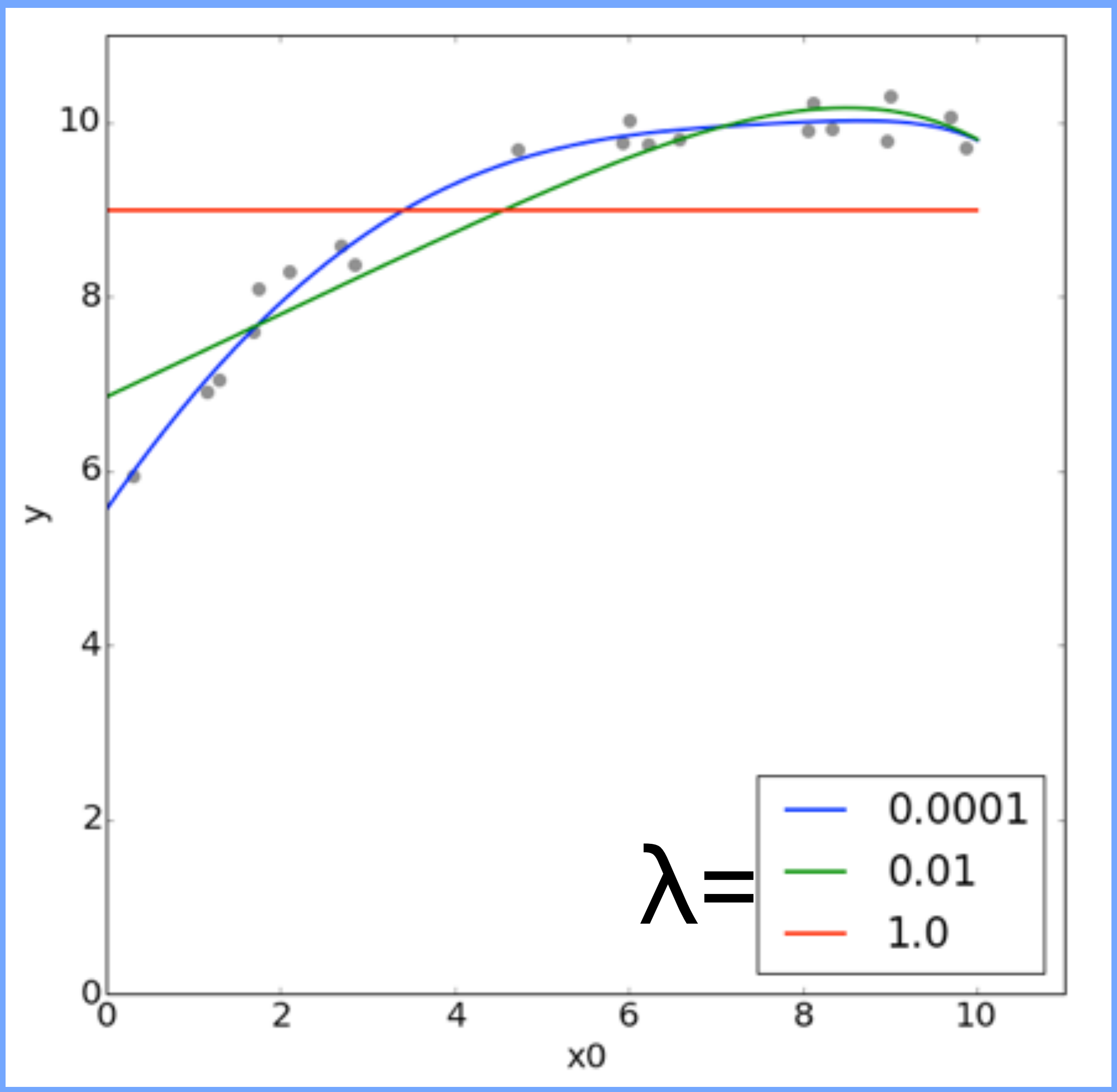
Normalize data

Cross validate to pick λ

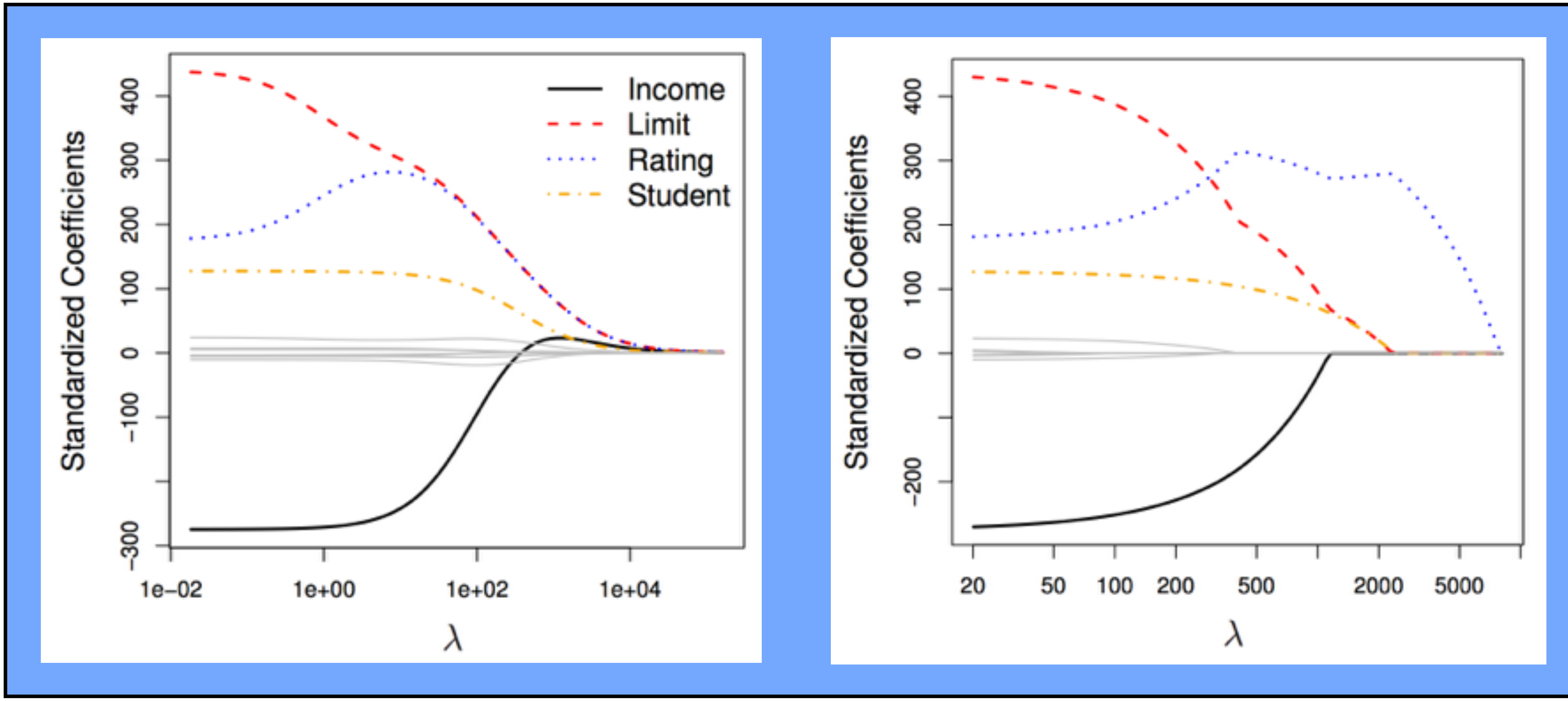
Ridge Regression



Lasso Regression



Ridge vs. Lasso



- Neither dominate
- Lasso's tendency to set coefficients exactly equal to zero:
  - Useful for feature selection and/or when response is a function of few predictors

scikit-learn

- `sklearn.linear_model.`**`LinearRegression`**(...)
- **`...Ridge`**(*alpha=1.0, ...)*
- **`...Lasso`**(*alpha=1.0, ...)*

Questions

- What is training error? validation error? test error?
- What are the steps to cross-validation?
  - How would you use it to compare say p different models?
- Same question as above, except with K-fold cross-validation
- What is the Bias-Variance tradeoff?
  - What happens with Bias and Variance at low levels of complexity?
  - What happens with Bias and Variance at high levels of complexity?
- How do Ridge and Lasso attempt to win at the Bias-Variance tradeoff?
  - What's being penalized exactly?