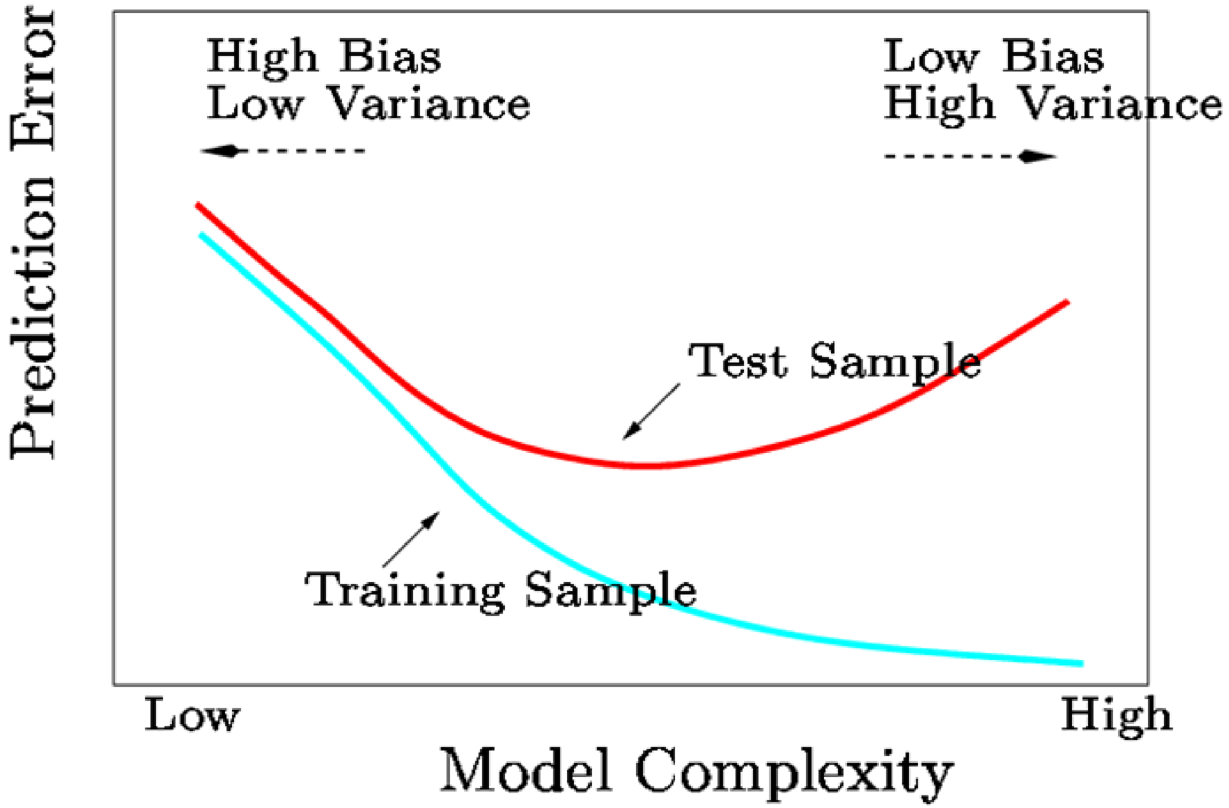
ADV DATA MINING AND ANALYTICS

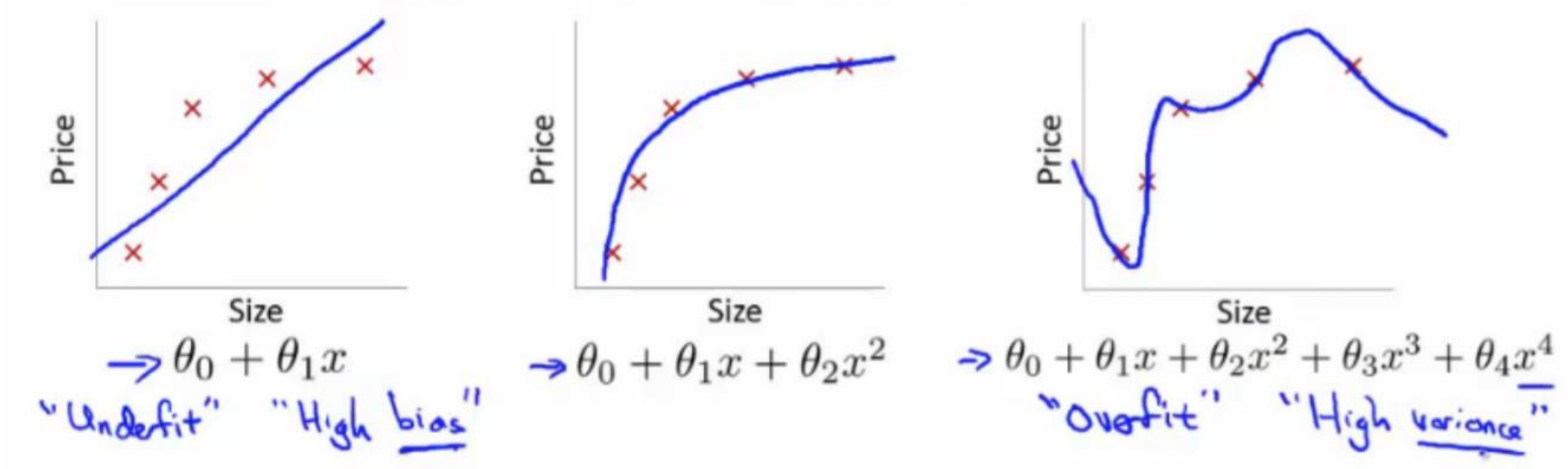
Regularization & GLM

Rouzbeh Razavi, PhD

Our Old Friend !



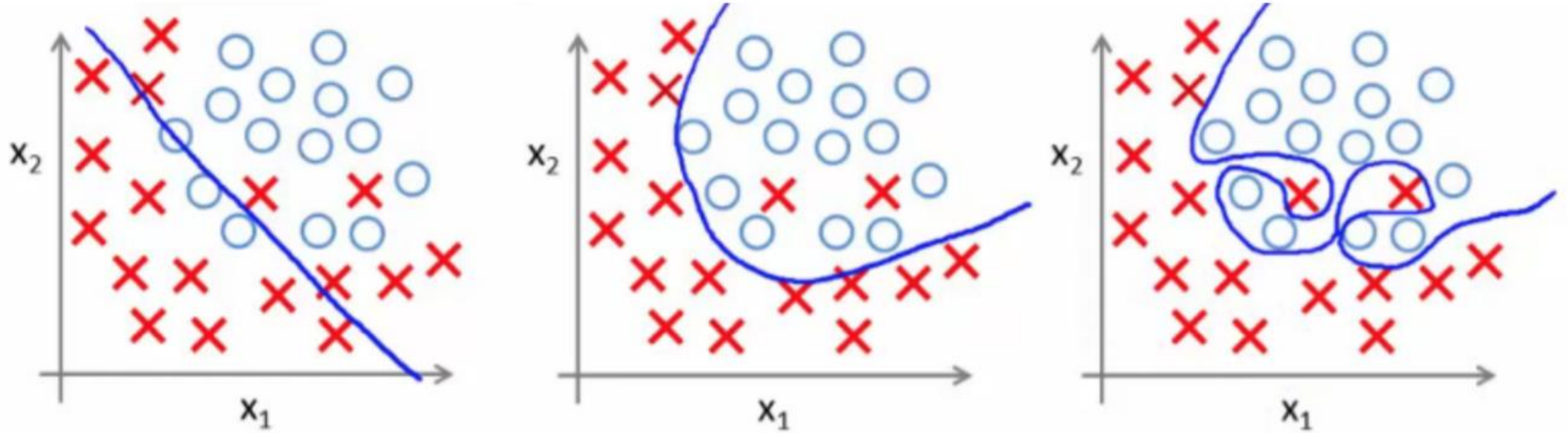
Overfitting: Regression



To recap, if we have too many features then the learned hypothesis may give a cost function of exactly zero

- But this tries too hard to fit the training set
- Fails to provide a *general* solution - **unable to generalize** (apply to new examples)

Overfitting: Classification



$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

(g = sigmoid function)

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$

$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots$

UNDERFITTING
(high bias)

OVERFITTING
(high variance)

Overfitting: Counter Measures

- How do we deal with this?
  - 1) **Reduce number of features**
    - Manually select which features to keep
    - Model selection algorithms are discussed later (good for reducing number of features)
    - But, in reducing the number of features we lose some information
      - Ideally select those features which minimize data loss, but even so, some info is lost
  - 2) **Regularization**
    - Keep all features, but reduce magnitude of parameters θ
    - Works well when we have a lot of features, each of which contributes a bit to predicting y

Overfitting: Counter Measures

Cost function optimization for regularization

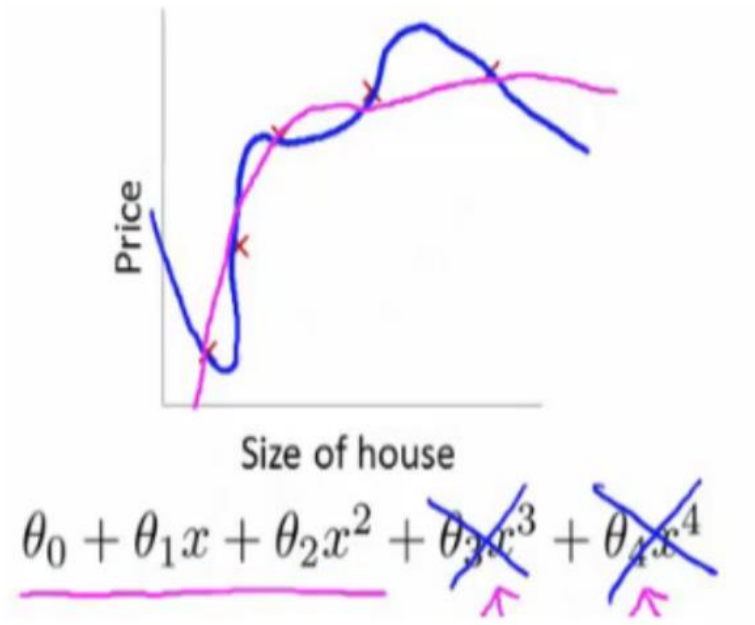
- Penalize and make some of the θ parameters really small
  - e.g. here θ_3 and θ_4

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + 1000\,\theta_3^2 + 1000\,\theta_4^2$$

- The addition in blue is a modification of our cost function to help penalize θ_3 and θ_4

Overfitting: Counter Measures

- So here we end up with θ_3 and θ_4 being close to zero (because the constants are massive)
- So we're basically left with a quadratic function



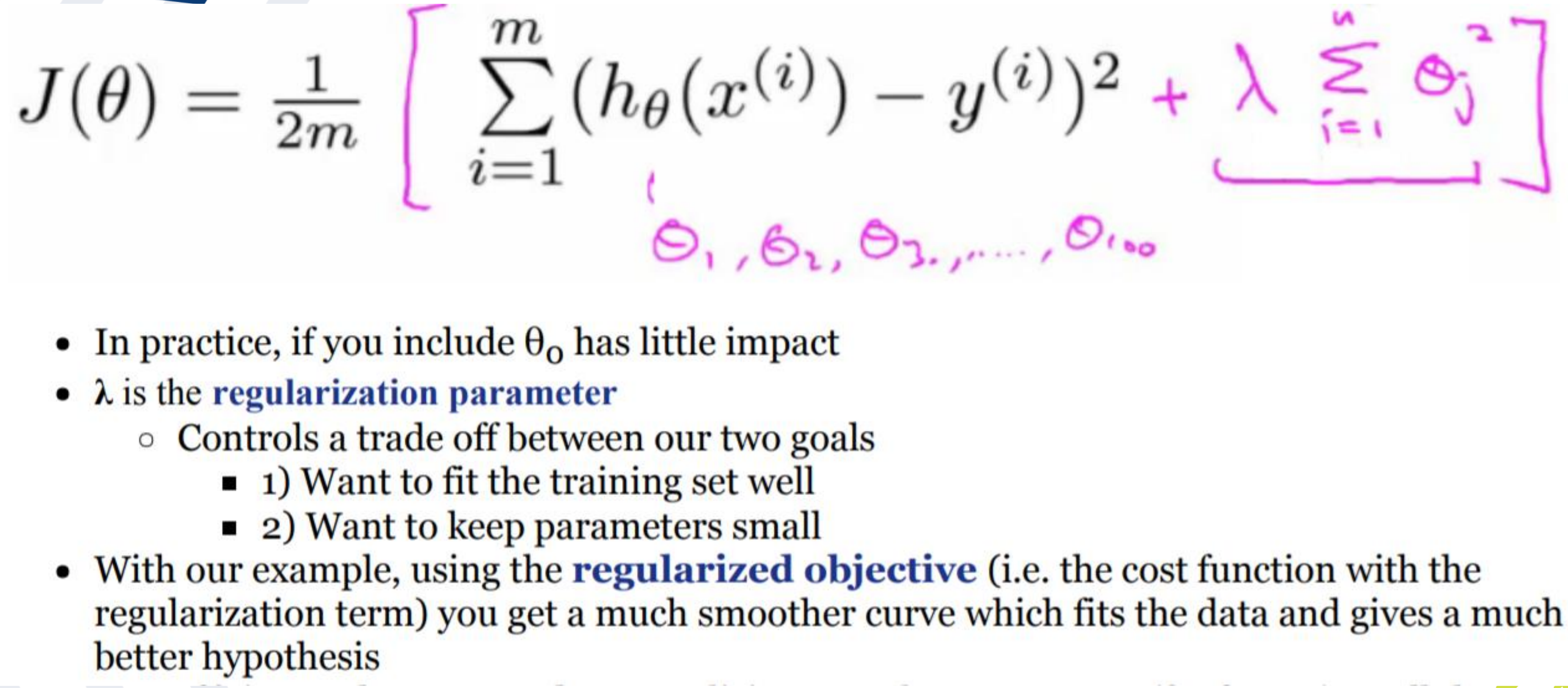
$$\underline{\theta_0 + \theta_1 x + \theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

Regularization

- In this example, we penalized two of the parameter values
  - More generally, regularization is as follows

- Regularization
  - Small values for parameters corresponds to a simpler hypothesis (you effectively get rid of some of the terms)
  - A simpler hypothesis is less prone to overfitting
- Another example
  - Have 100 features $x_1, x_2, \ldots, x_{100}$
  - Unlike the polynomial example, we don't know what are the high order terms
    - How do we pick the ones to pick to shrink?
  - With regularization, take cost function and modify it to shrink all the parameters
    - Add a term at the end
      - This regularization term shrinks every parameter
      - By convention you don't penalize θ_0 - minimization is from θ_1 onwards

Regularization

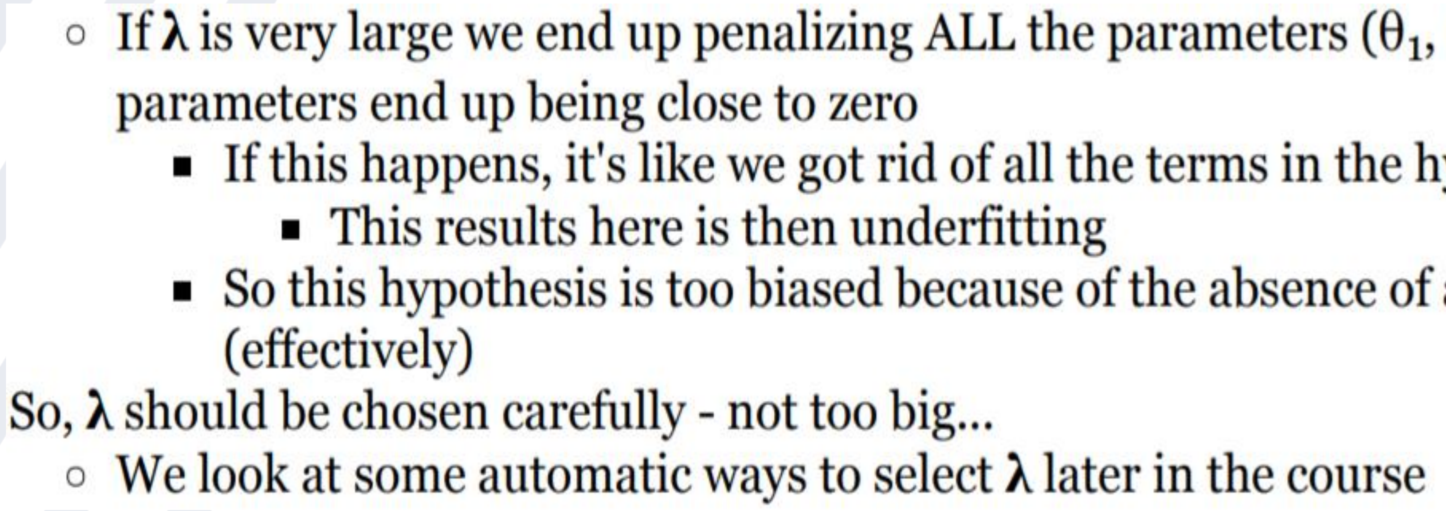


$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \underbrace{\lambda \sum_{i=1}^{n} \theta_j^2}\right]$$

$$\theta_1, \theta_2, \theta_3, \ldots, \theta_{100}$$

- In practice, if you include θ_0 has little impact
- λ is the **regularization parameter**
  - Controls a trade off between our two goals
    - 1) Want to fit the training set well
    - 2) Want to keep parameters small
- With our example, using the **regularized objective** (i.e. the cost function with the regularization term) you get a much smoother curve which fits the data and gives a much better hypothesis

Regularization

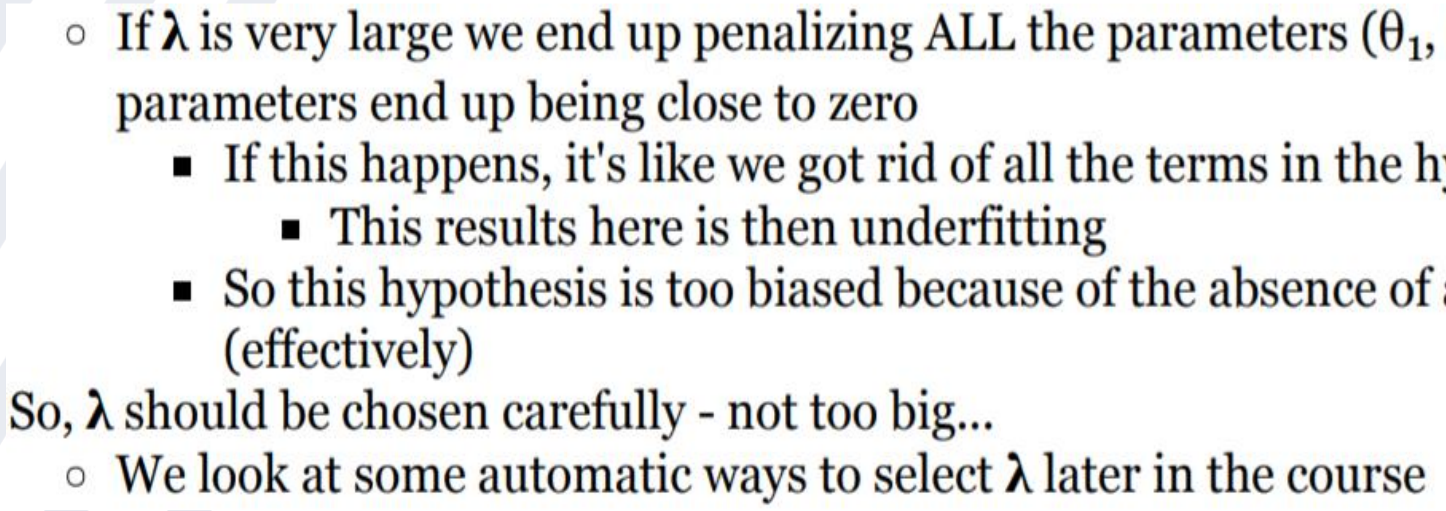


- If λ is very large we end up penalizing ALL the parameters (θ_1, θ_2 etc.) so all the parameters end up being close to zero
  - If this happens, it's like we got rid of all the terms in the hypothesis
    - This results here is then underfitting
  - So this hypothesis is too biased because of the absence of any parameters (effectively)

So, λ should be chosen carefully - not too big...

- We look at some automatic ways to select λ later in the course

Regularization



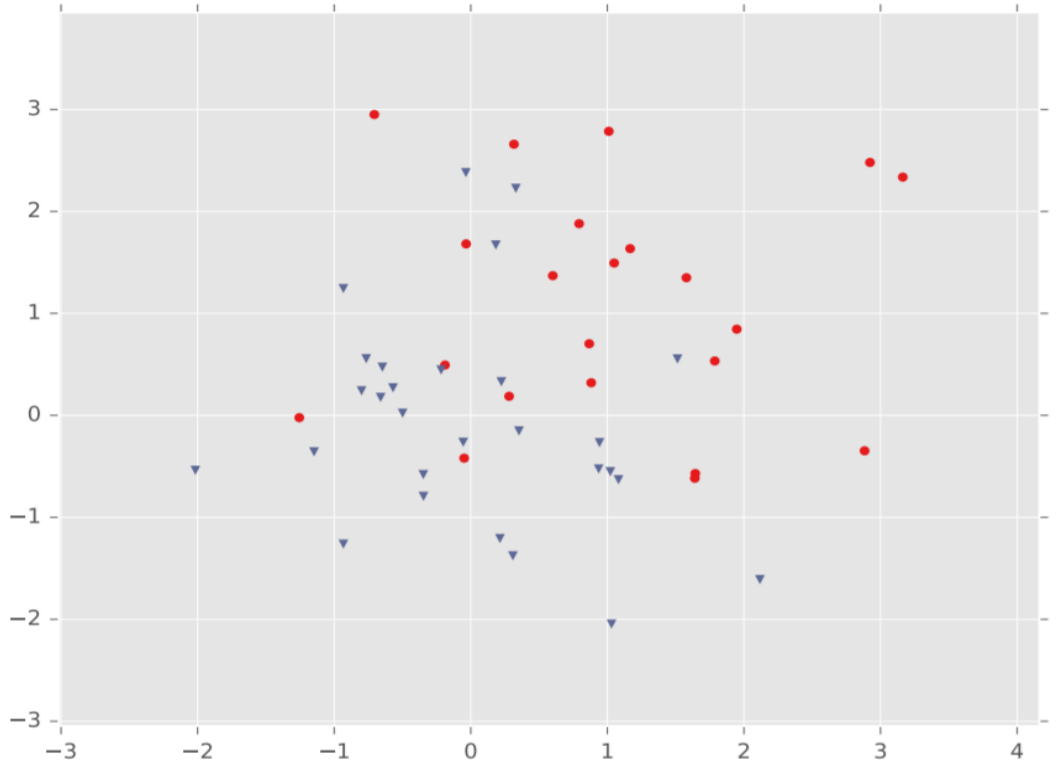
- If λ is very large we end up penalizing ALL the parameters (θ_1, θ_2 etc.) so all the parameters end up being close to zero
  - If this happens, it's like we got rid of all the terms in the hypothesis
    - This results here is then underfitting
  - So this hypothesis is too biased because of the absence of any parameters (effectively)

So, λ should be chosen carefully - not too big...

- We look at some automatic ways to select λ later in the course

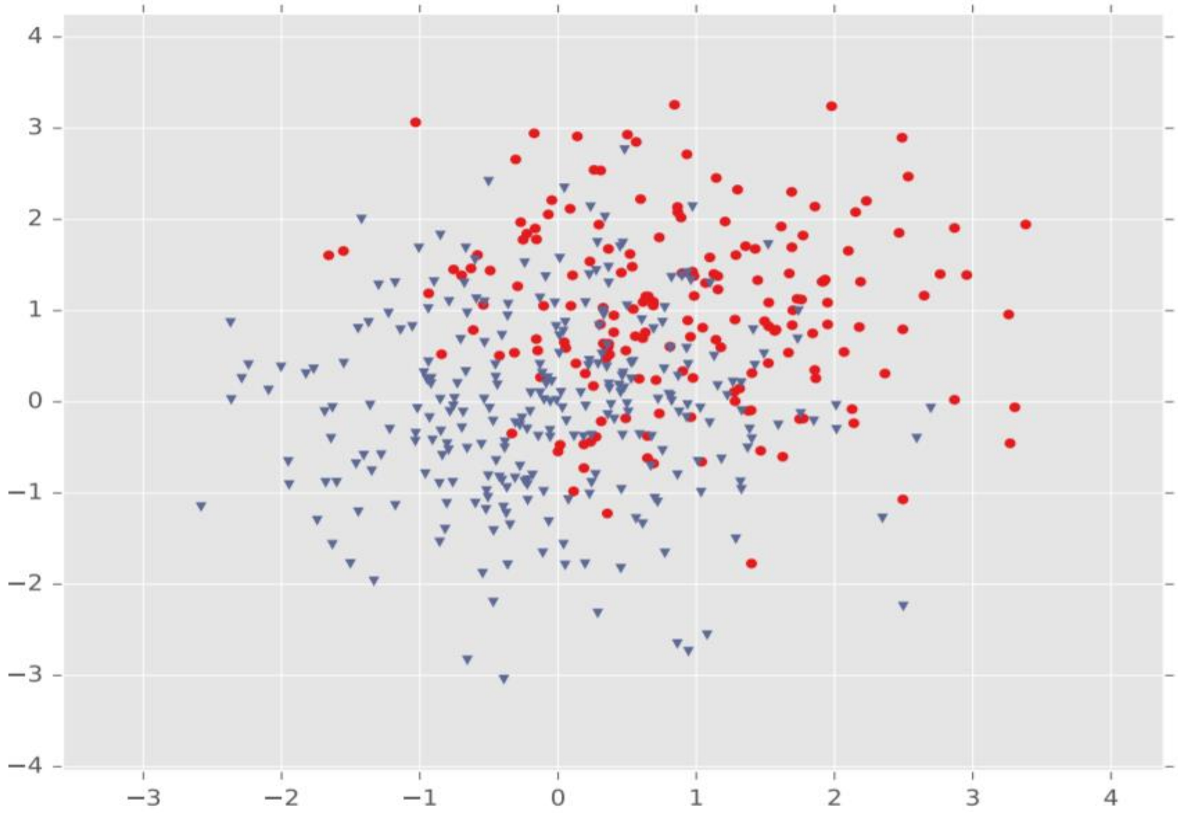
Example: Logistic Regression

Training Data

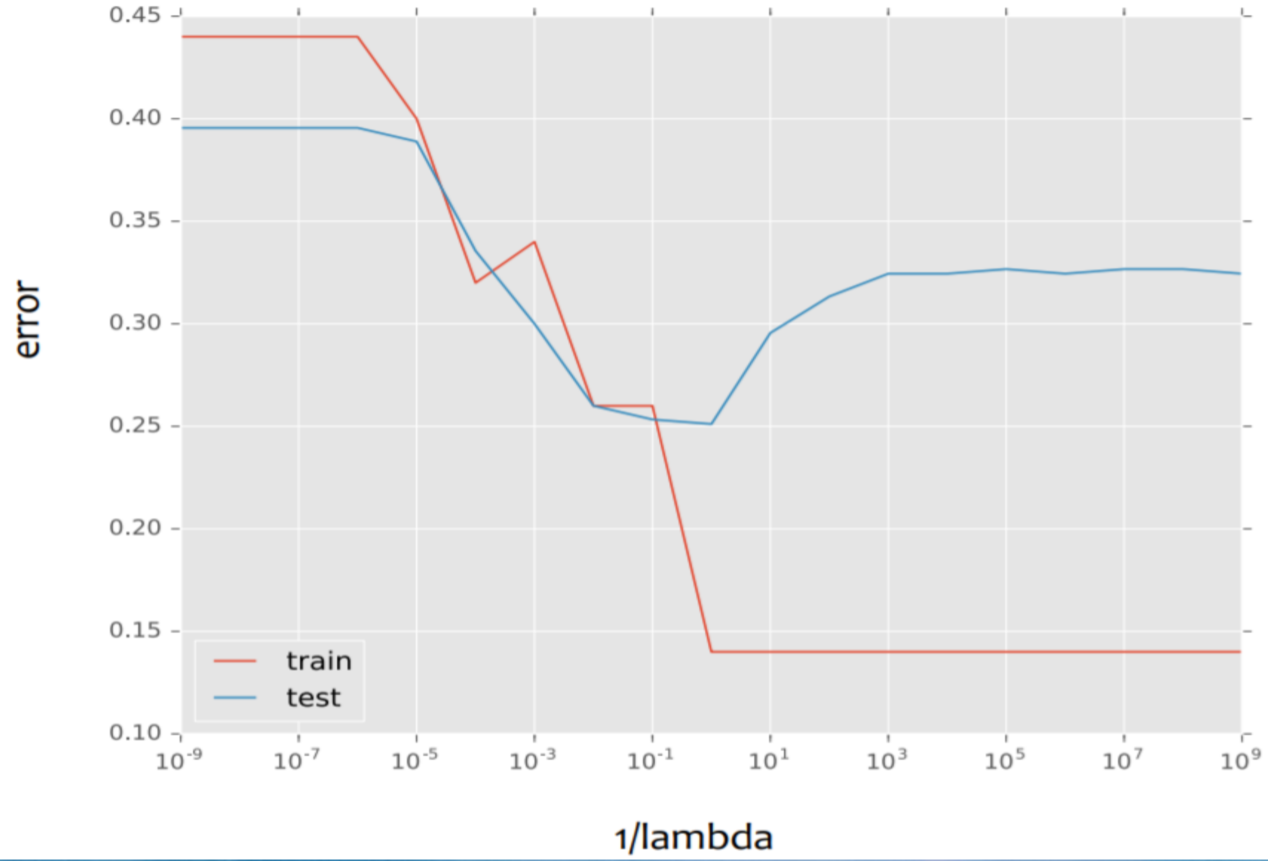


Example: Logistic Regression

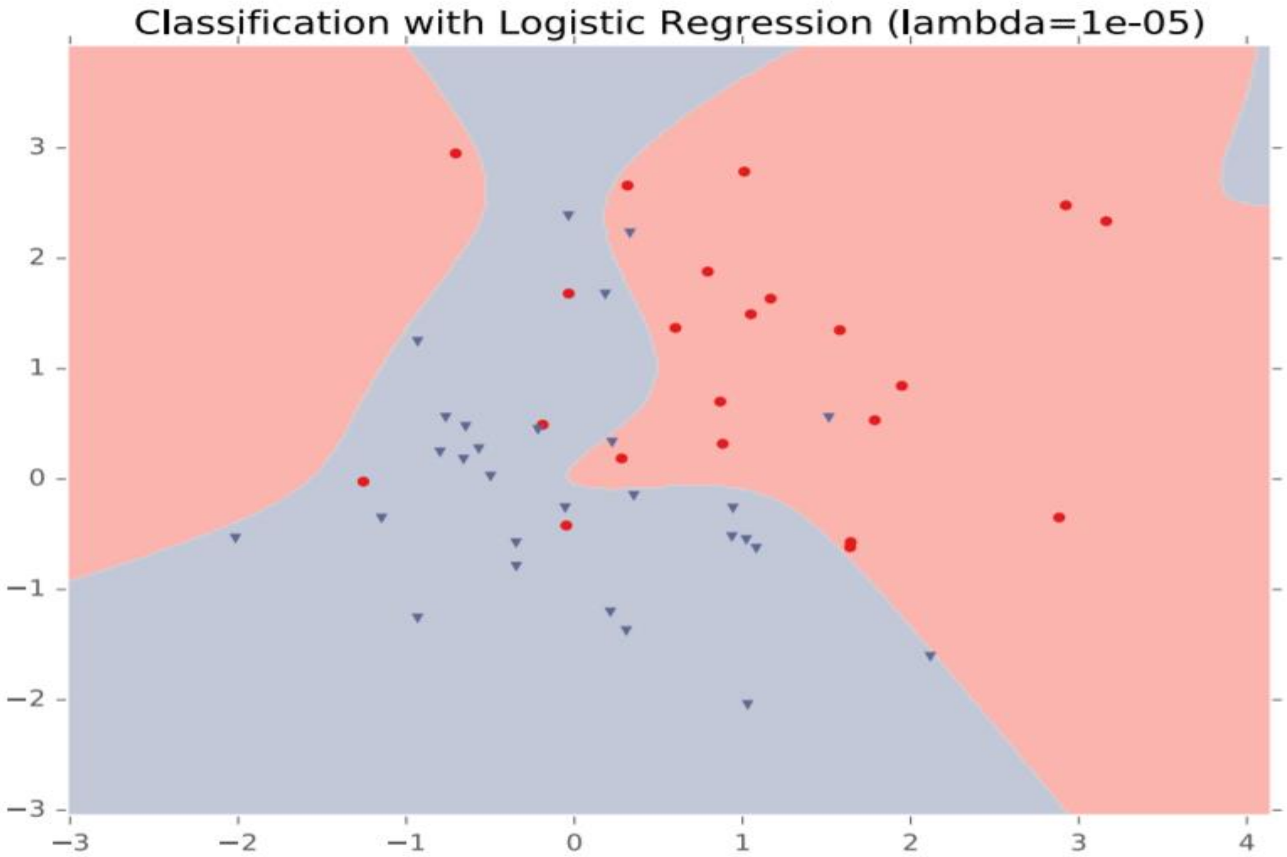
Test Data



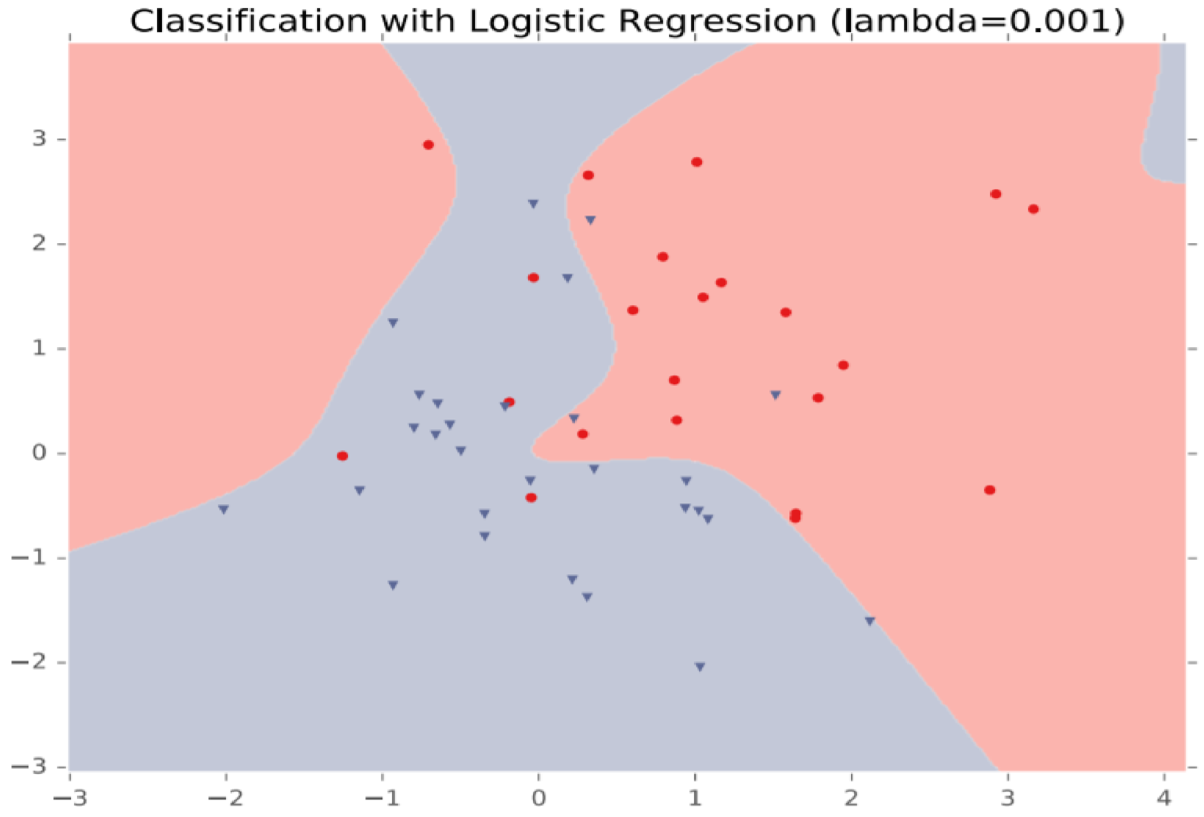
Example: Logistic Regression



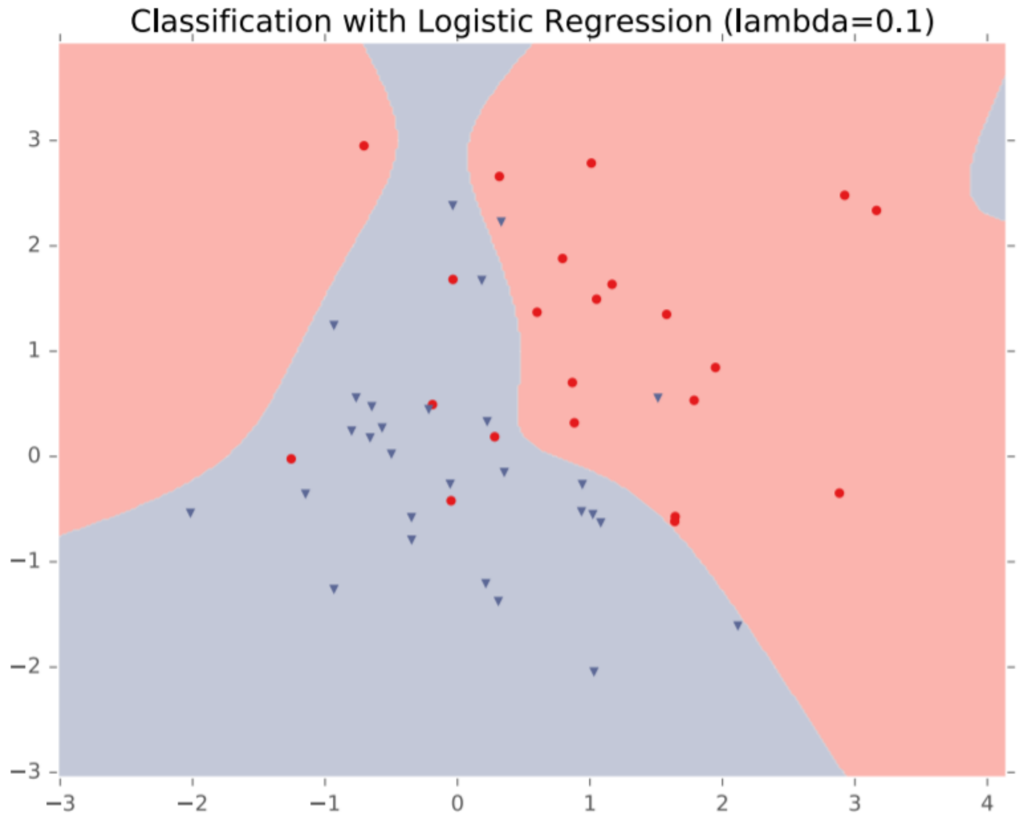
Example: Logistic Regression



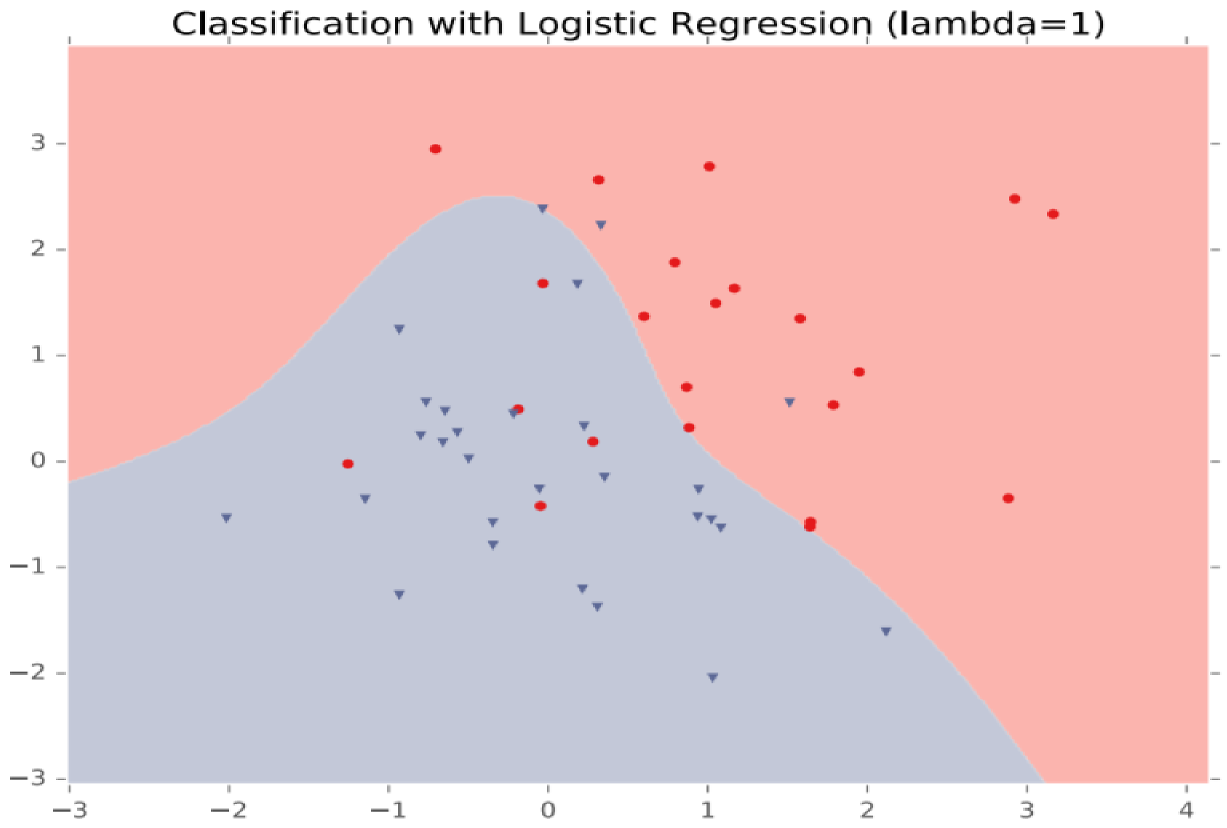
Example: Logistic Regression



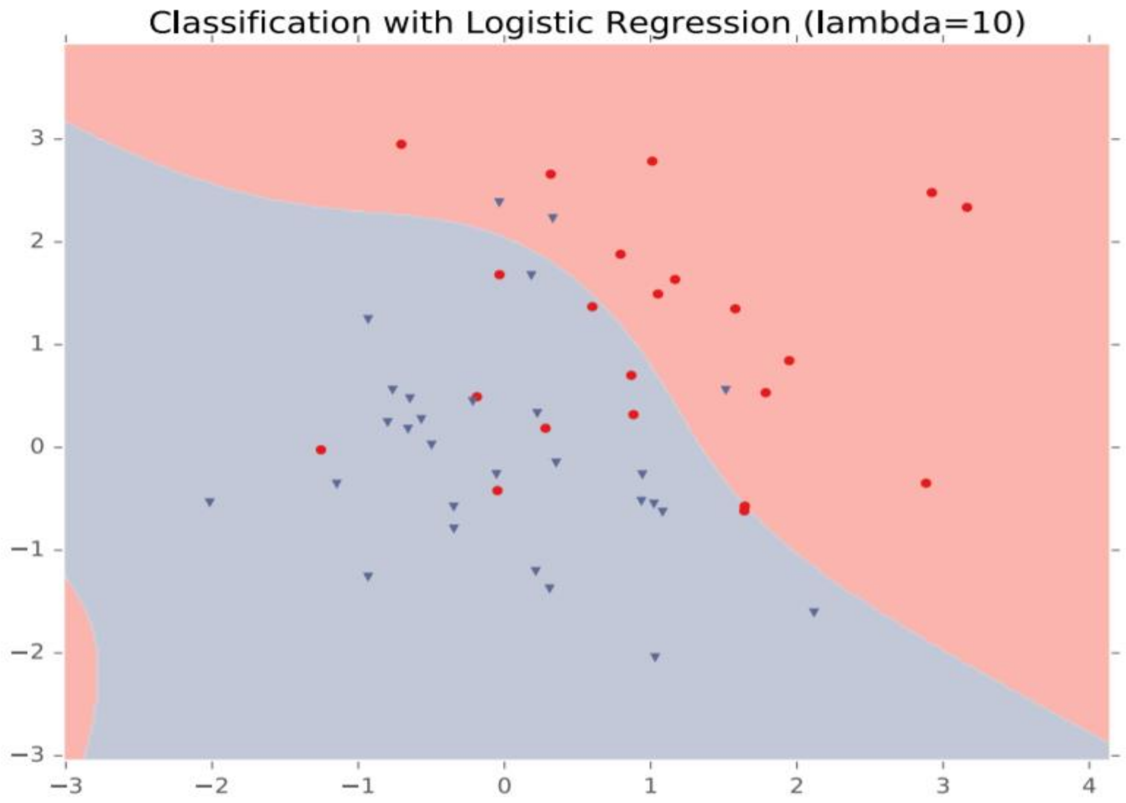
Example: Logistic Regression



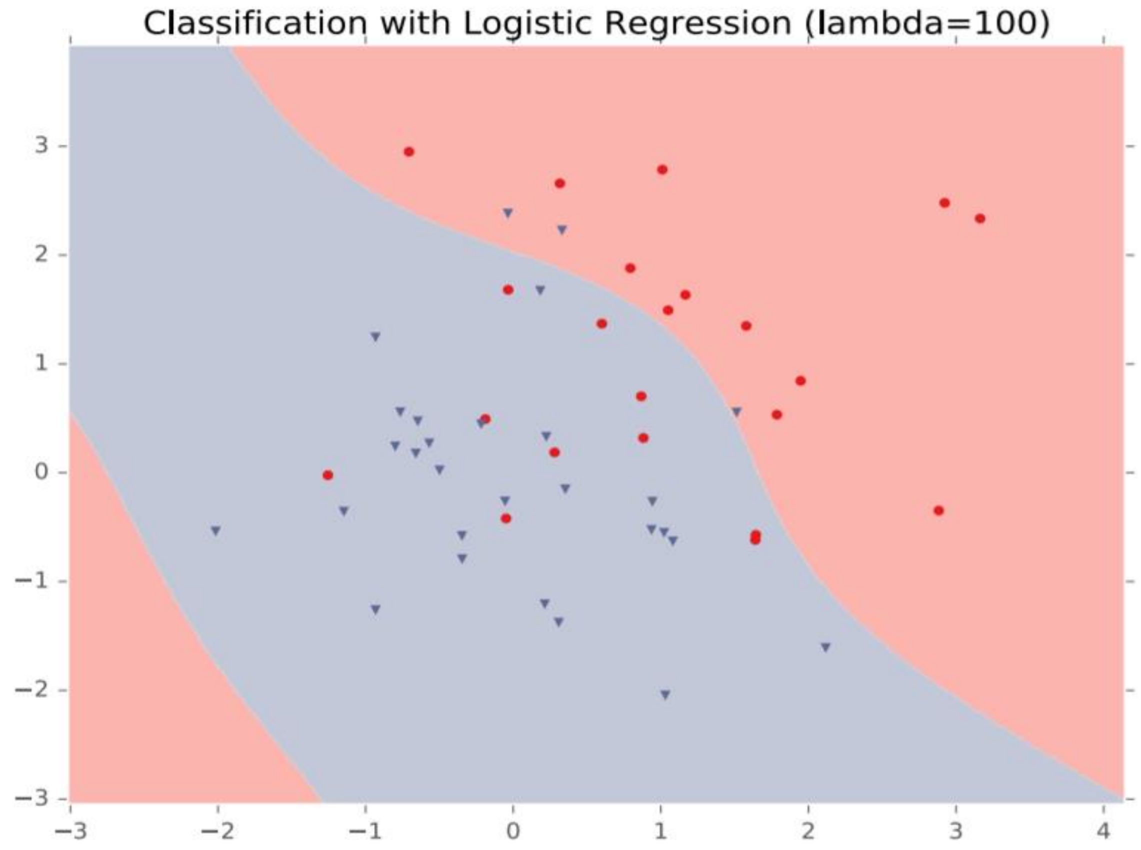
Example: Logistic Regression



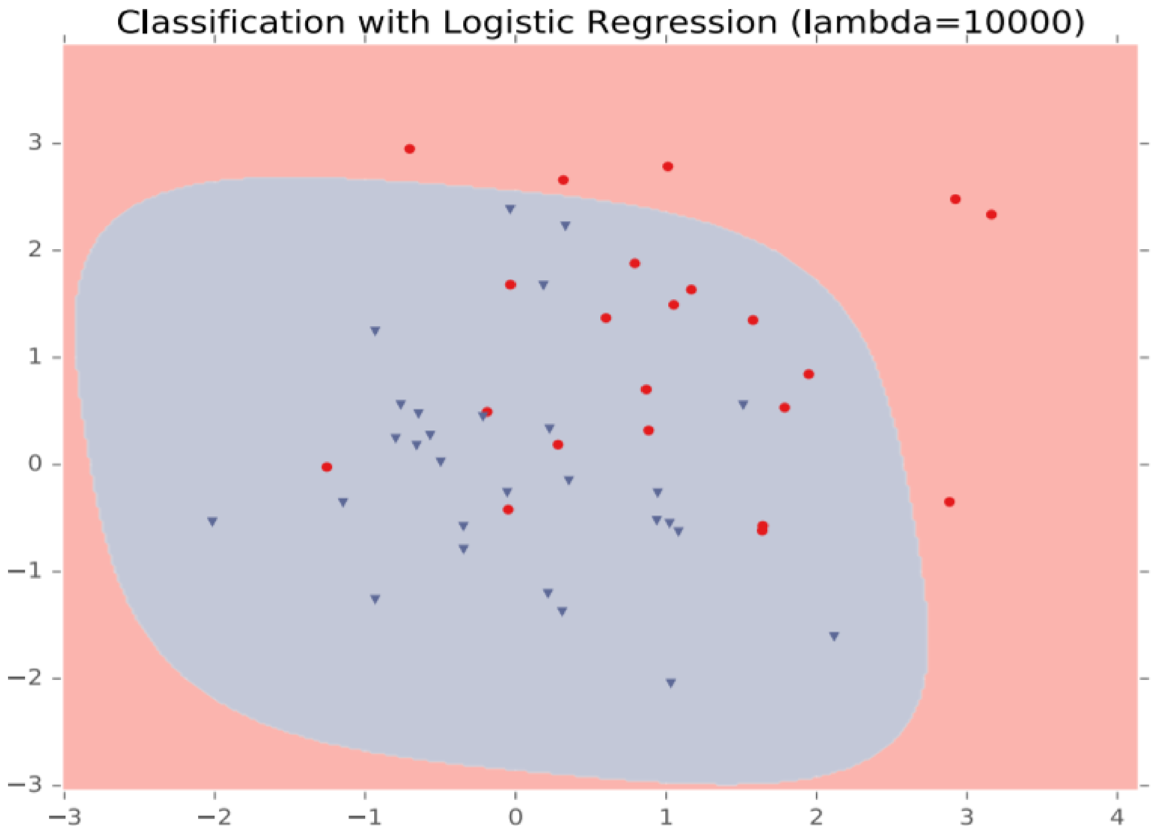
Example: Logistic Regression



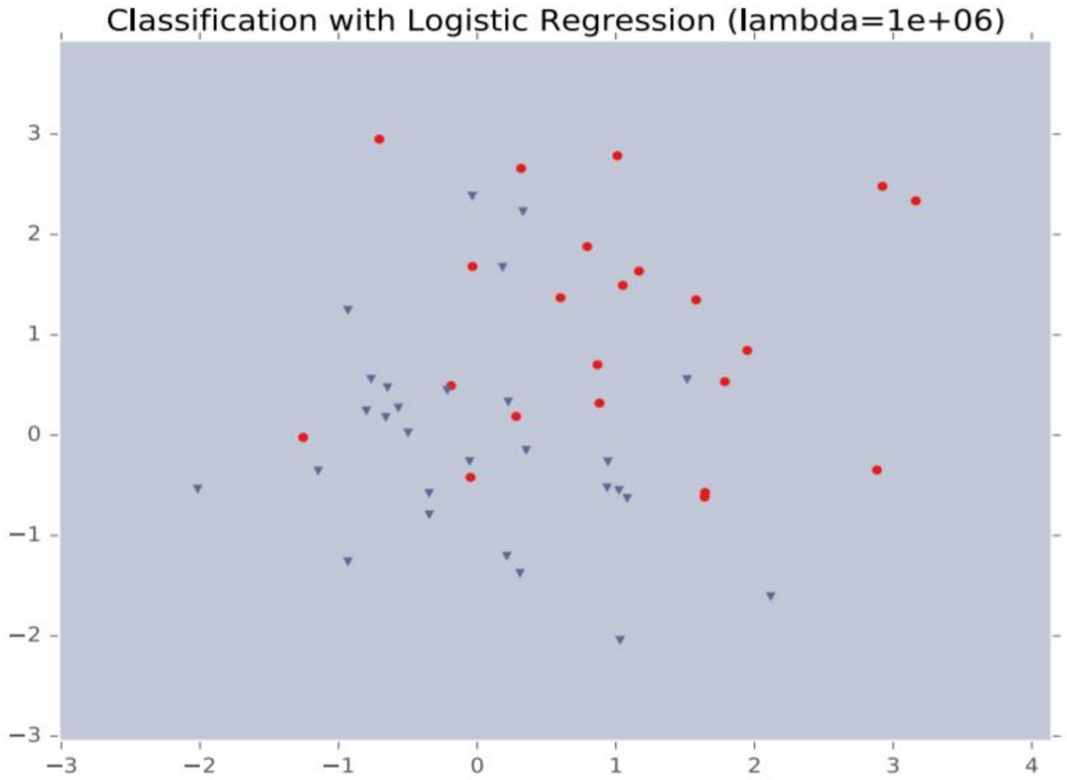
Example: Logistic Regression



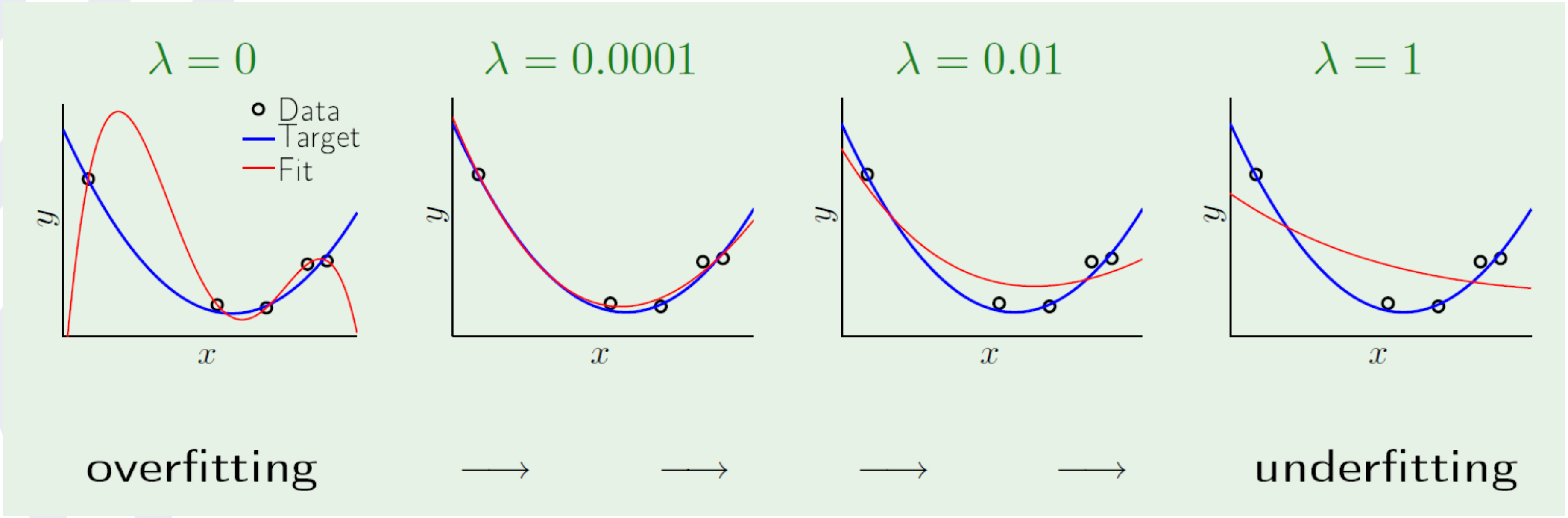
Example: Logistic Regression



Example: Logistic Regression



Another Example



Loss Functions to Minimize

Square of Errors as seen before

Ridge Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\overbrace{(h_\theta(x^{(i)}) - y^{(i)})^2} + \underbrace{\lambda \sum_{j=1}^{n} \theta_j^2}\right]$$

$\theta_1, \theta_2, \theta_3, \ldots, \theta_{100}$

Lasso Regression

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \underbrace{\lambda \sum_{j=1}^{n} |\theta_j|}\right]$$

$\theta_1, \theta_2, \theta_3, \ldots, \theta_{100}$

Lasso vs Ridge

Note the name "lasso" is actually an acronym for: Least Absolute Selection and Shrinkage Operator

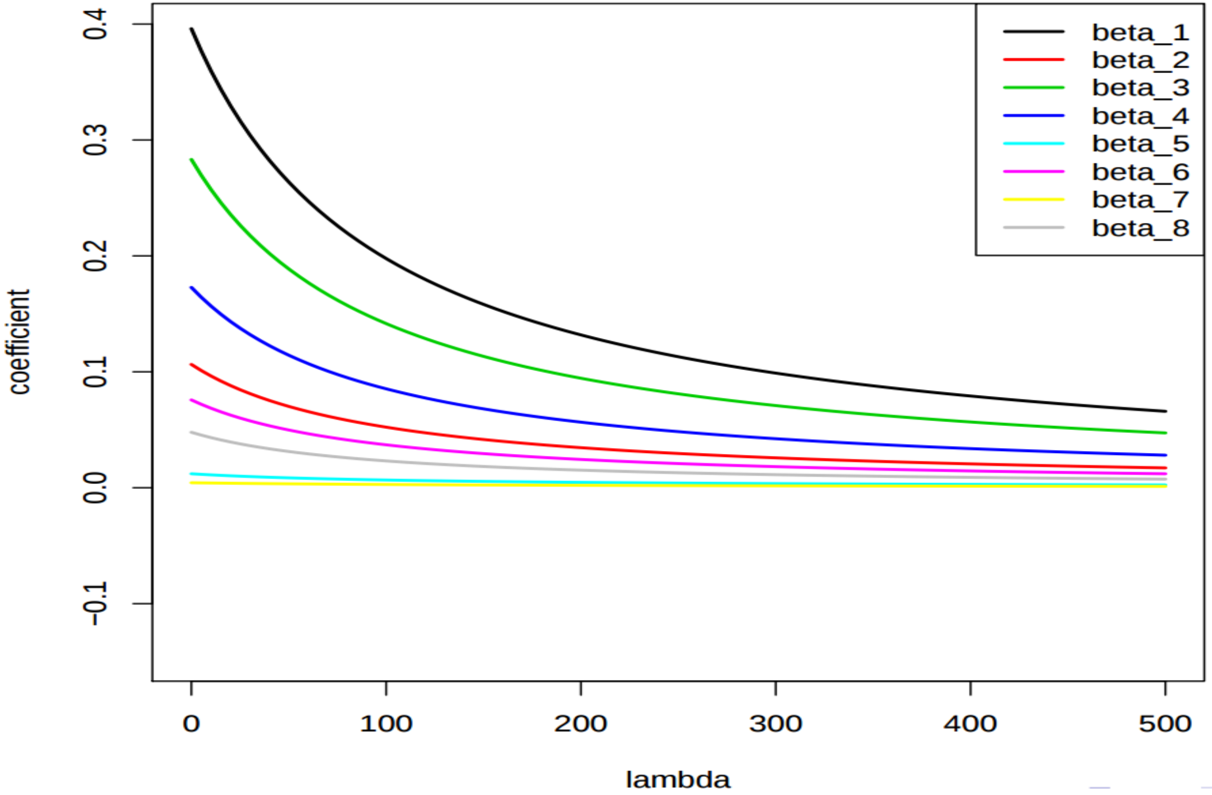
The only difference between the lasso problem and ridge regression is that the latter uses a (squared) ℓ_2 penalty $\|\theta\|_2^2$, while the former uses an ℓ_1 penalty $\|\theta\|_1$. But even though these problems look similar, their solutions behave very differently

Lasso regression

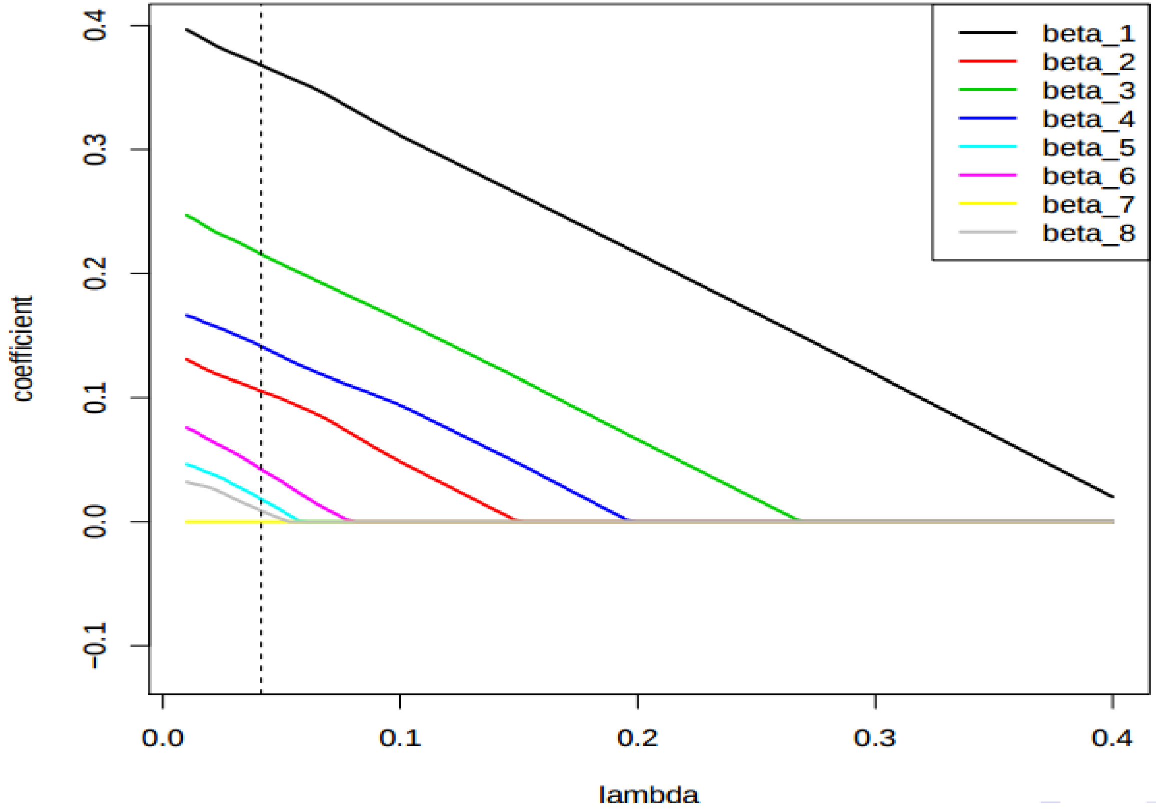
For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients. But the nature of the ℓ_1 penalty causes some coefficients to be shrunken to zero exactly

This is what makes the lasso substantially different from ridge regression: it is able to perform variable selection in the linear model. As λ increases, more coefficients are set to zero (less variables are selected), and among the nonzero coefficients, more shrinkage is employed

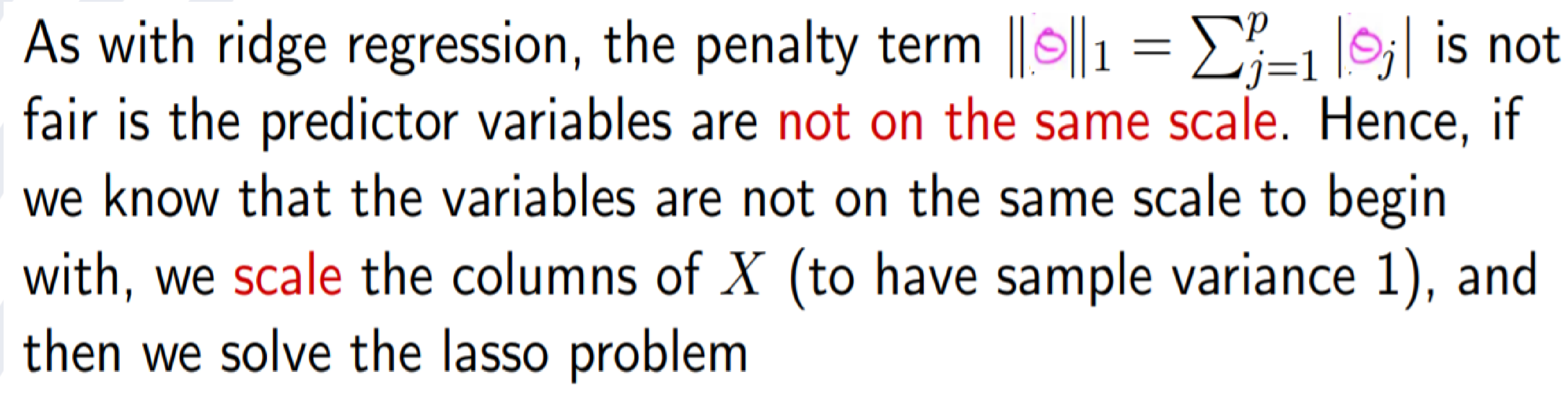
Ridge regression



Lasso regression



Lasso regression



As with ridge regression, the penalty term $\|\theta\|_1 = \sum_{j=1}^{p} |\theta_j|$ is not fair is the predictor variables are not on the same scale. Hence, if we know that the variables are not on the same scale to begin with, we scale the columns of X (to have sample variance 1), and then we solve the lasso problem

Constrained form

It can be helpful to think of our two problems constrained form:

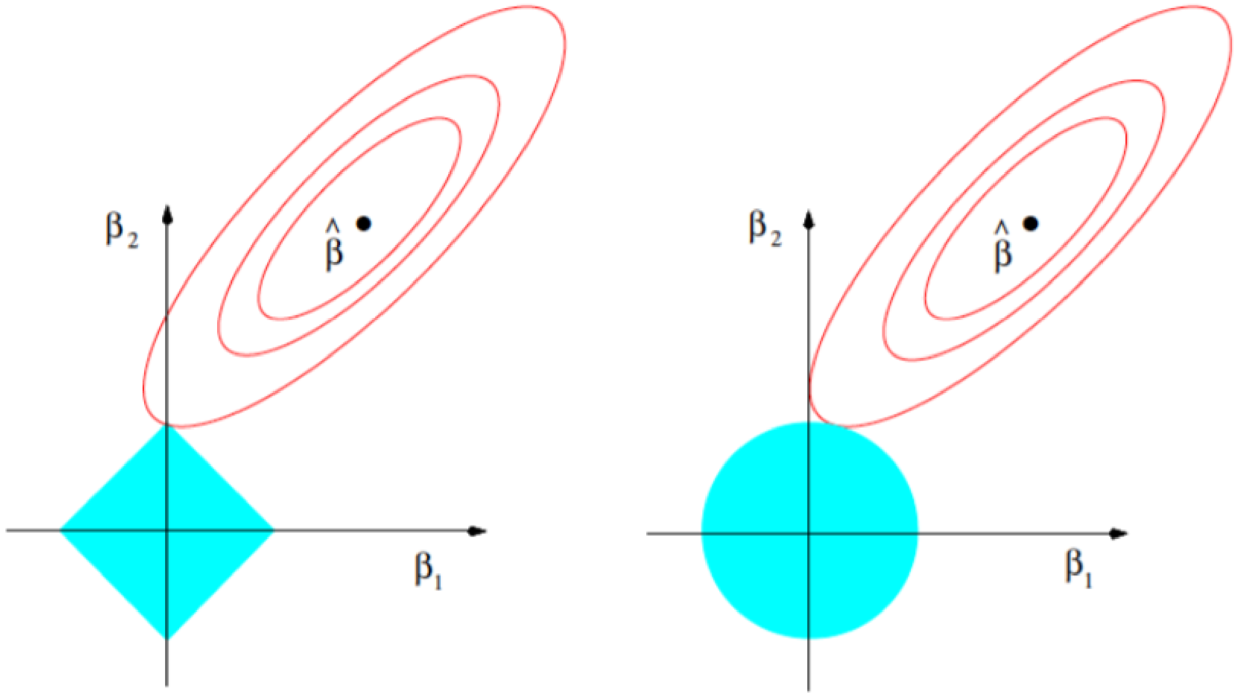
$$\hat{\theta}^{\text{ridge}} = \underset{\theta \in \mathbb{R}^p}{\text{argmin}} \; \|y - X\theta\|_2^2 \quad \text{subject to } \|\theta\|_2^2 \leq t$$

$$\hat{\theta}^{\text{lasso}} = \underset{\theta \in \mathbb{R}^p}{\text{argmin}} \; \|y - X\theta\|_2^2 \quad \text{subject to } \|\theta\|_1 \leq t$$

Now t is the tuning parameter (before it was λ). For any λ and corresponding solution in the previous formulation (sometimes called penalized form), there is a value of t such that the above constrained form has this same solution

In comparison, the usual linear regression estimate solves the unconstrained least squares problem; these estimates constrain the coefficient vector to lie in some geometric shape centered around the origin. This generally reduces the variance because it keeps the estimate close to zero. But which shape we choose really matters!

Why does the lasso give zero coefficients?



Time for Implementation ! Glmnet Package

https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html

Rob Tibshirani

Trevor Hastie

ANY QUESTIONS ?