Announcements

Struggling?

- Don’t struggle alone
- Come talk to Pat
  - OH
  - 1-on-1 appointment calendar
  - Private message on Piazza with set of times to meet

Announcements

Assignments

- HW5
  - Fri, 2/24, 11:59 pm

Midterm

- Wed, 3/1, in-class
- Details will be coming on Piazza
  - Logistics — scope
  - Learning objectives for Midterm 1 topics
  - Review session
  - Practice exam problems

Announcements

Struggling?

- Don’t struggle alone
- Come talk to Pat
  - OH
  - 1-on-1 appointment calendar
  - Private message on Piazza with set of times to meet

Plan

Today

- Wrap-up neural nets (for now)
- Regularization
  - Make sure they aren't too powerful ☺

Wrap up Neural Nets

Switch to neural nets slides

10-315
Introduction to ML

Regularization

Instructor: Pat Virtue

Poll 1

Which is model do you prefer, assuming both have zero training error?

Model structure (for both models):

$$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7 + \theta_8 x^8$$

Model parameters:

$$\boldsymbol{\theta} = [\theta_0,\ \theta_1,\ \theta_2,\ \theta_3,\ \theta_4,\ \theta_5,\ \theta_6,\ \theta_7,\ \theta_8]^T$$

A. $\boldsymbol{\theta}_A = [-190.0,\ -135.0,\ 310.0,\ 45.0,\ -62.0,\ 90.0,\ -82.0,\ -40.0,\ 29.0]^T$

B. $\boldsymbol{\theta}_B = [\ 25.5,\ -6.4,\ -0.8,\ 0.0,\ 6.6,\ -4.4,\ 0.2,\ -2.9,\ 0.1]^T$

Poll 1

Which is model do you prefer, assuming both have zero training error?

Model structure (for both models):

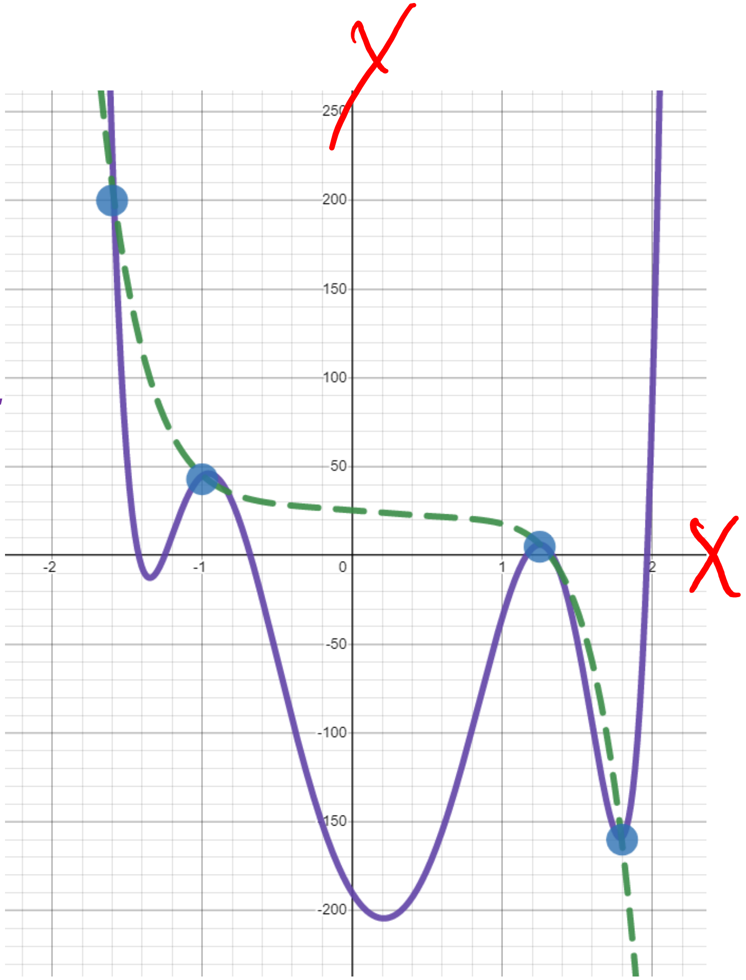
y = $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7 + \theta_8 x^8$

Model parameters:

$\boldsymbol{\theta} = [\theta_0,\ \theta_1,\ \theta_2,\ \theta_3,\ \theta_4,\ \theta_5,\ \theta_6,\ \theta_7,\ \theta_8]^T$

A. $\boldsymbol{\theta}_A = [-190.0,\ -135.0,\ 310.0,\ 45.0,\ -62.0,\ 90.0,\ -82.0,\ -40.0,\ 29.0]^T$

B. $\boldsymbol{\theta}_B = [\ 25.5,\ -6.4,\ -0.8,\ 0.0,\ 6.6,\ -4.4,\ 0.2,\ -2.9,\ 0.1]^T$

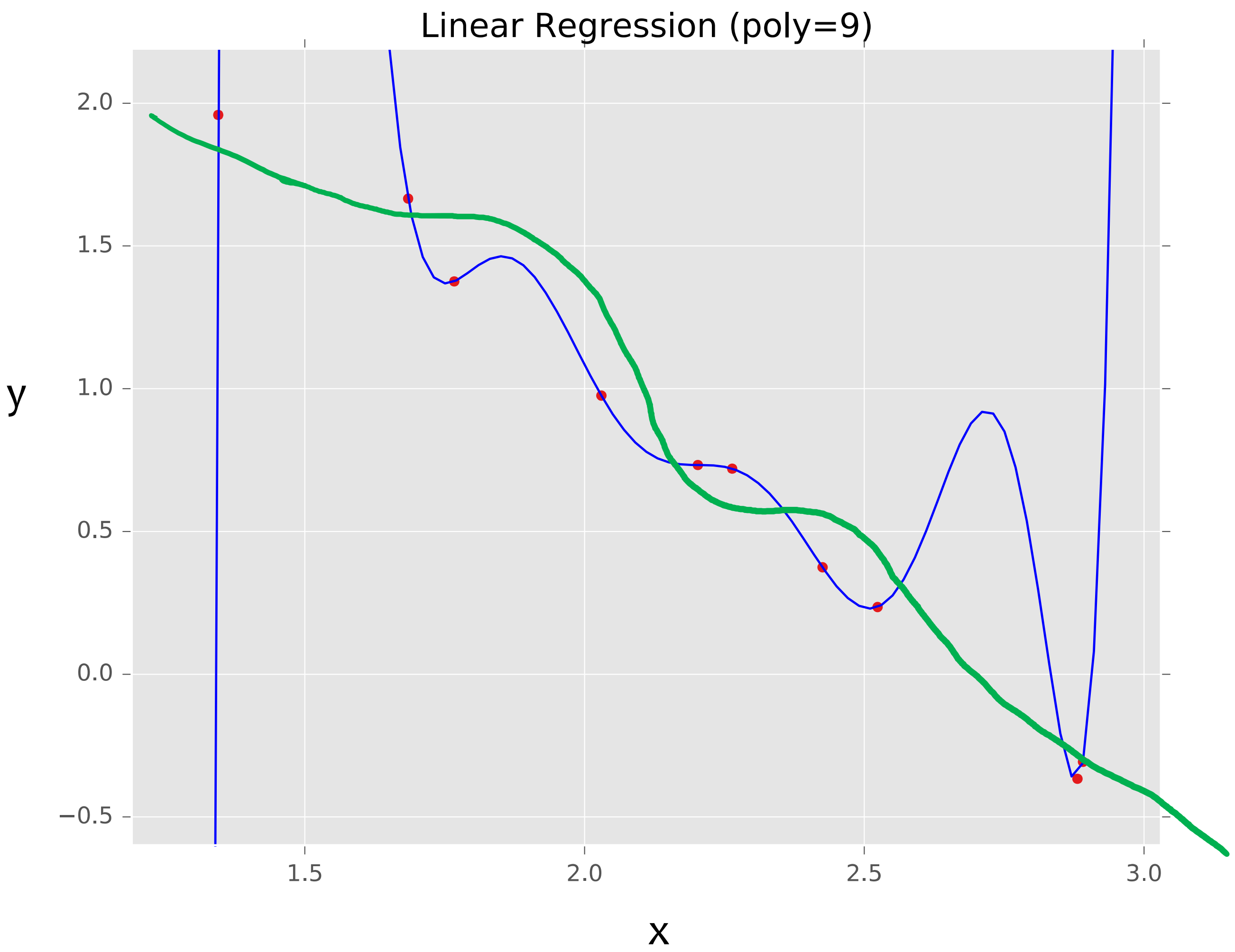


Example: Linear Regression

Goal: Learn y = $\mathbf{w}^T f(\mathbf{x}) + b$ where f(.) is a polynomial basis function

y	x	x²	…	x⁹
2.0	1.2	$(1.2)^2$	…	$(1.2)^9$
1.3	1.7	$(1.7)^2$	…	$(1.7)^9$
0.1	2.7	$(2.7)^2$	…	$(2.7)^9$
1.1	1.9	$(1.9)^2$	…	$(1.9)^9$

true "unknown" target function is linear with negative slope and gaussian noise



Slide credit: CMU MLD Matt Gormley

Symptoms of Overfitting

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
θ_0	0.19	0.82	0.31	0.35
θ_1		-1.27	7.99	232.37
θ_2			-25.43	-5321.83
θ_3			17.37	48568.31
θ_4				-231639.30
θ_5				640042.26
θ_6				-1061800.52
θ_7				1042400.18
θ_8				-557682.99
θ_9				125201.43

Slide credit: CMU MLD William Cohen

Model Preference

Which is model do you prefer, assuming both have zero training error?

Model structure (for both models):

$$h_{\boldsymbol{\theta}}(\mathbf{x}) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5 + \theta_6 x_6 + \theta_7 x_7 + \theta_8 x_8$$

Model parameters:

$$\boldsymbol{\theta} = [\theta_0, \ \theta_1, \ \theta_2, \ \theta_3, \ \theta_4, \ \theta_5, \ \theta_6, \ \theta_7, \ \theta_8]^T$$

A. $\boldsymbol{\theta}_A = [-190.0, \ -135.0, \ 310.0, \ 45.0, \ -62.0, \ 90.0, \ -82.0, \ -40.0, \ 29.0]^T$

B. $\boldsymbol{\theta}_B = [\ 25.5, \ -6.4, \ -0.8, \ 0.0, \ 6.6, \ -4.4, \ 0.2, \ -2.9, \ 0.1]^T$

What if $\mathbf{x}$ was a vector of input feature measurements (rather than polynomial features)?

Motivation: Regularization

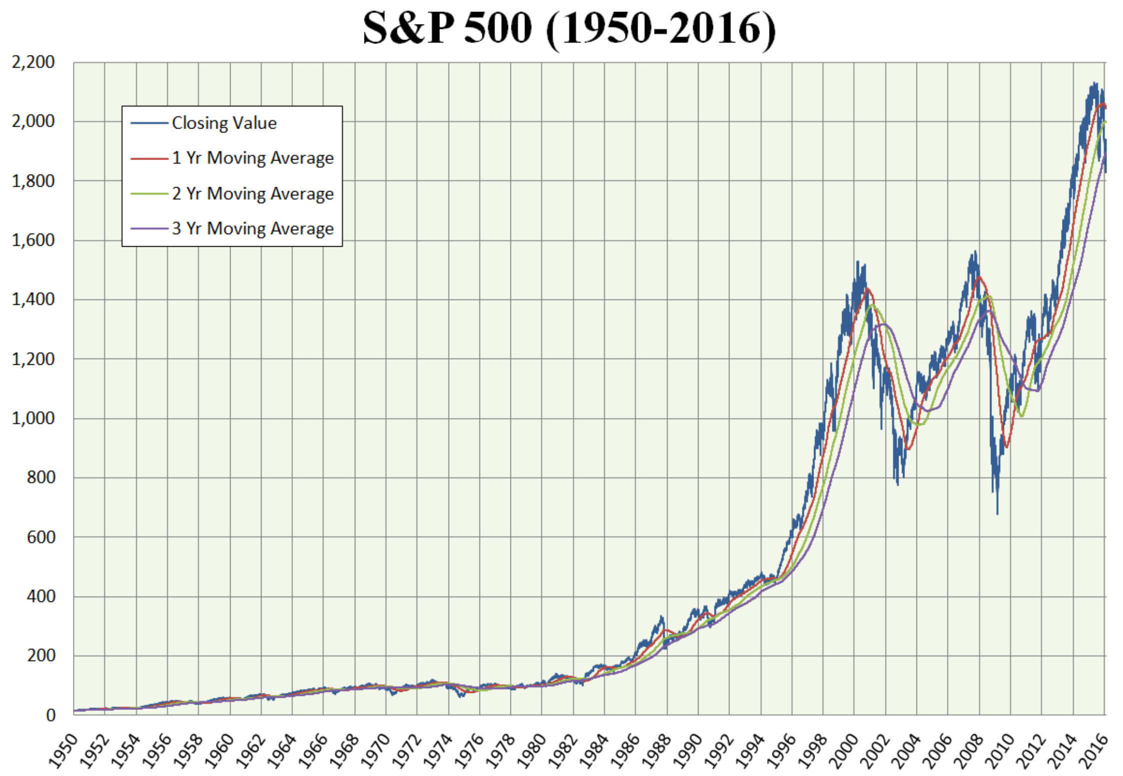
Example: Stock Prices

Suppose we wish to predict Google's stock price at time t+1

What features should we use?
(putting all computational concerns aside)

- Stock prices of all other stocks at times t, t-1, t-2, …, t - k
- Mentions of Google with positive / negative sentiment words in all newspapers and social media outlets

Do we believe that **all** of these features are going to be useful?



Slide credit: CMU MLD Matt Gormley

Overfitting

Definition: The problem of **overfitting** is when the model captures the noise in the training data instead of the underlying structure

Overfitting can occur in all the models we've seen so far:

- Decision Trees (e.g. when tree is too deep)
- K-NN (e.g. when k is small)
- Linear Regression (e.g. with nonlinear features or extraneous features)
- Logistic Regression (e.g. with nonlinear features or extraneous features)
- Neural networks

Slide credit: CMU MLD Matt Gormley

Motivation: Regularization

Occam's Razor: prefer the simplest hypothesis

What does it mean for a hypothesis (or model) to be **simple**?

1. small number of features (**model selection**)
2. small number of "important" features (**shrinkage**)

3. small magnitude of features

Regularization

Key idea:

Define regularizer $r(\boldsymbol{\theta})$ that we will add to our minimization objective to keep the model simple

$r(\boldsymbol{\theta})$ should be:

- Small for a simple model
- Large for a complex model

L2 norm: square-root of sum of squares

$$r(\theta) = ||\theta||_2 = \left(\sum_{i=1}^{M} (\theta_i)^2\right)^{1/2}$$

L1 norm: sum of absolute values

L0 norm: count of non-zero values

Regularization

	$\|\boldsymbol{\theta}\|_2$	$\|\boldsymbol{\theta}\|_1$	$\|\boldsymbol{\theta}\|_0$
A. $\boldsymbol{\theta}_A = [6,\ 3, -4, -2]^T$	8.06	15	4
B. $\boldsymbol{\theta}_B = [0,\ 3, -4,\ \ 0]^T$	5	7	2

Poll 2

Which model do you prefer?

A. $\boldsymbol{\theta}_A = [-190.0,\ -135.0,\ 310.0,\ 45.0]^T$ Training error: 0.0

B. $\boldsymbol{\theta}_B = [\ \ 0.0,\ \ 0.0,\ \ 0.0,\ \ 0.0]^T$ Training error: 34.2

$r(\theta)$ e.g. $\|\theta\|_2$

model complexity

$J(\theta) = \hat{R}(h)$

training error

Regularization

$(1-\alpha) J(\theta) + \alpha r(\theta)$

$||\theta||_2^2 \quad \theta^2$

Given objective function: $J(\theta)$

Goal is to find: $$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

Key idea: Define regularizer $r(\boldsymbol{\theta})$ s.t. we tradeoff between fitting the data and keeping the model simple

Choose form of $r(\boldsymbol{\theta})$:

- Example: q-norm (usually p-norm)

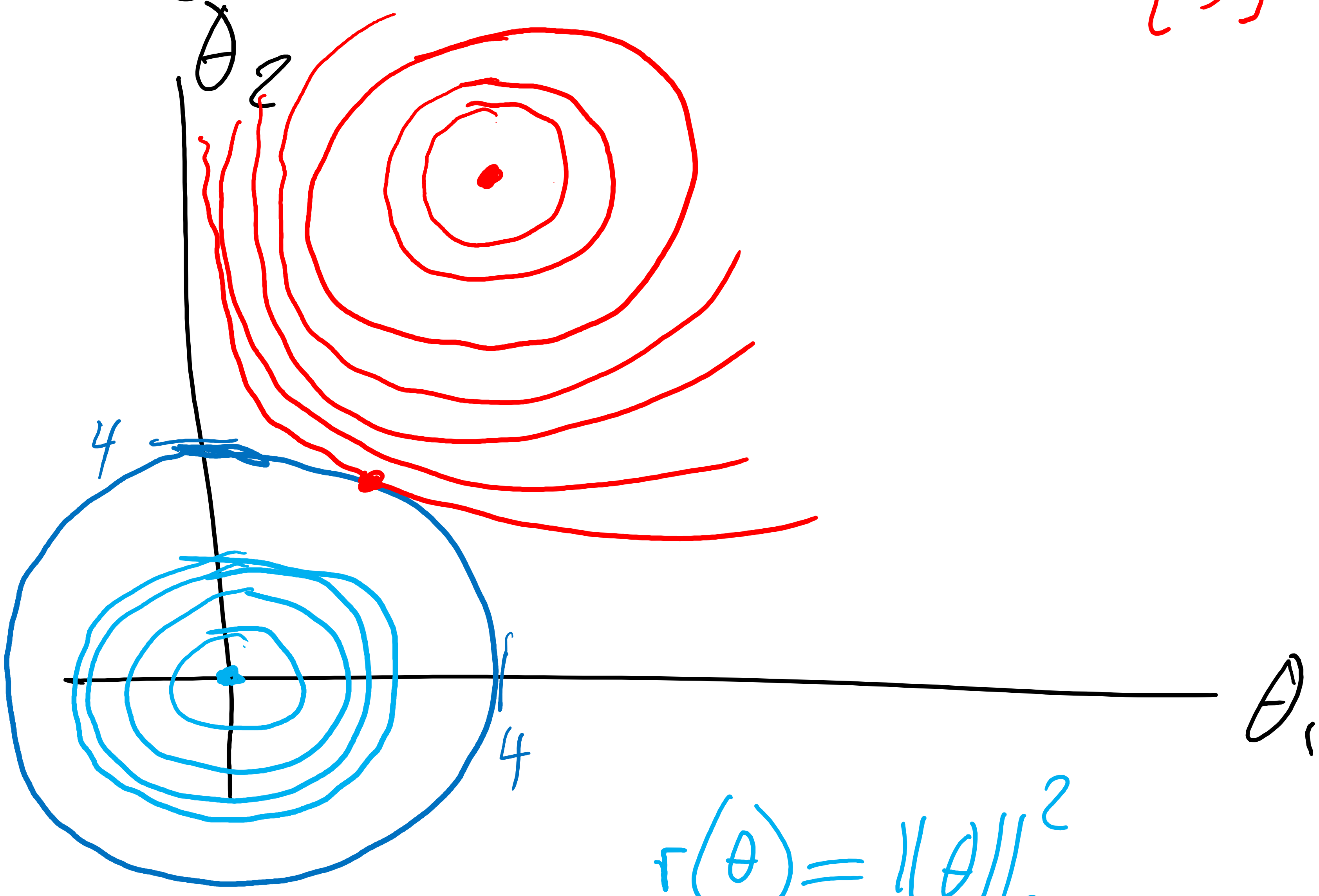
$$r(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_q = \left[\sum_{m=1}^{M} ||\theta_m||^q\right]^{\left(\frac{1}{q}\right)}$$

q	$r(\boldsymbol{\theta})$	**yields parameters that are...**	**name**	**optimization notes**
0	$\|\|\boldsymbol{\theta}\|\|_0 = \sum \mathbb{1}(\theta_m \neq 0)$	zero values	L0 reg.	no good computational solutions
1	$\|\|\boldsymbol{\theta}\|\|_1 = \sum \|\theta_m\|$	zero values	L1 reg.	subdifferentiable
2	$(\|\|\boldsymbol{\theta}\|\|_2)^2 = \sum \theta_m^2$	small values	L2 reg.	differentiable

Slide credit: CMU MLD Matt Gormley

Regularization

$J(\theta)$ $\theta = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$



$r(\theta) = \|\theta\|_2^2$ $r(\theta) \le 4$

Poll 3

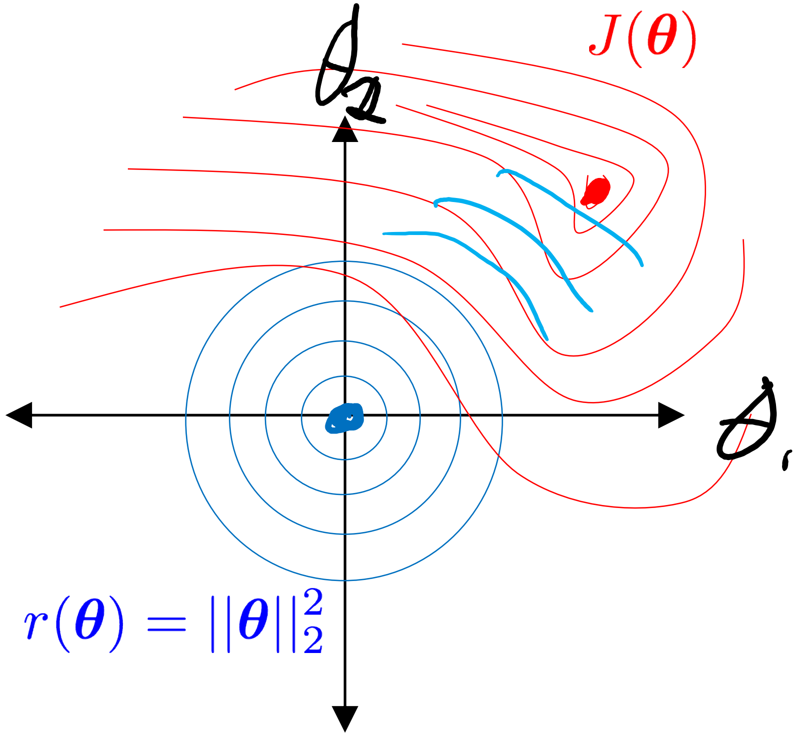
Question:

Suppose we are minimizing J'(θ) where

$$J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

As λ increases, the minimum of J'(θ) will…

A. …move towards the midpoint between J'(θ) and r(θ)
B. …move towards the minimum of J(θ)
C. …move towards the minimum of r(θ)
D. …move towards a theta vector of positive infinities
E. …move towards a theta vector of negative infinities
F. …stay the same

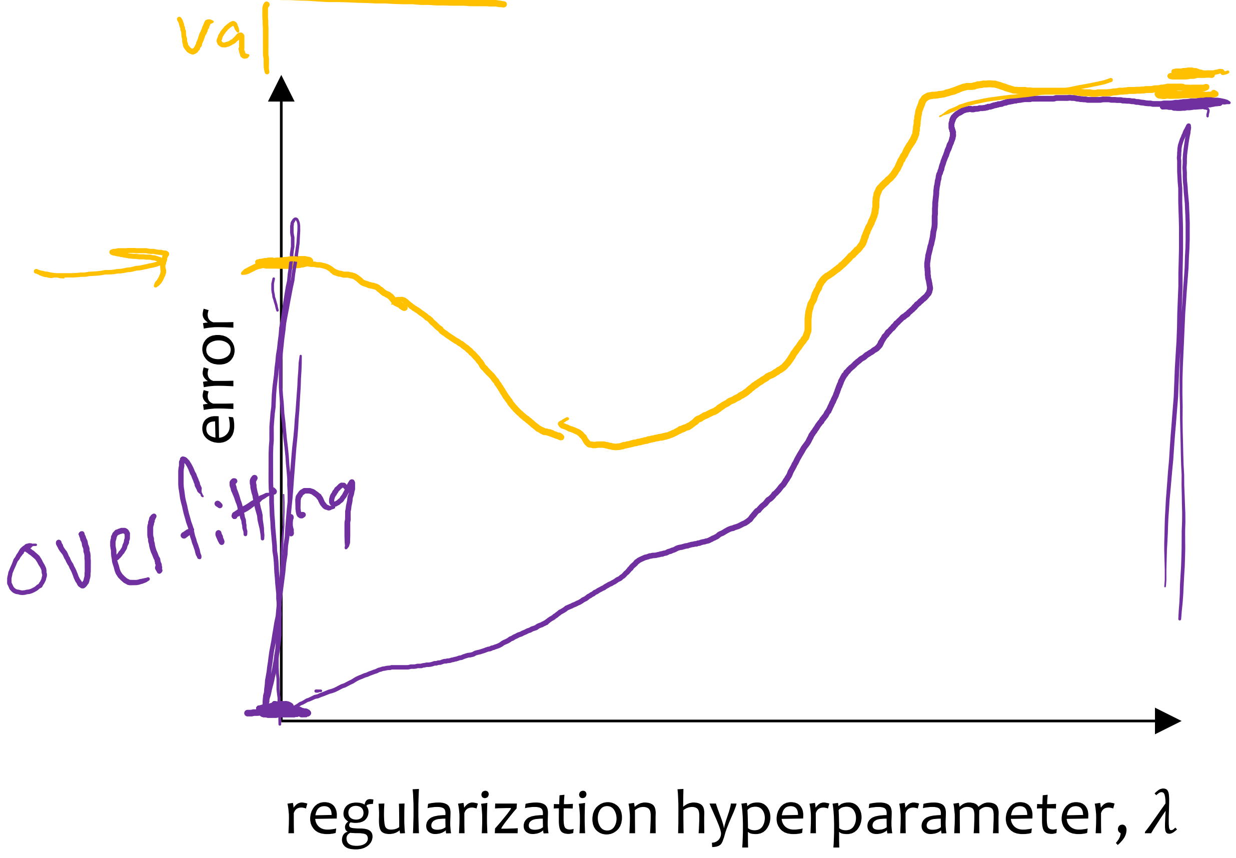


Slide credit: CMU MLD Matt Gormley

Regularization Exercise

In-class Exercise

1. Plot train error vs. regularization hyperparameter (cartoon)
2. Plot test error vs . regularization hyperparameter (cartoon)



$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

underfitting

Slide credit: CMU MLD Matt Gormley

Poll 4

Question:

Suppose we are minimizing J'(θ) where

$$J'(\boldsymbol{\theta}) = J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

As we increase λ from zero, the **validation** error will…

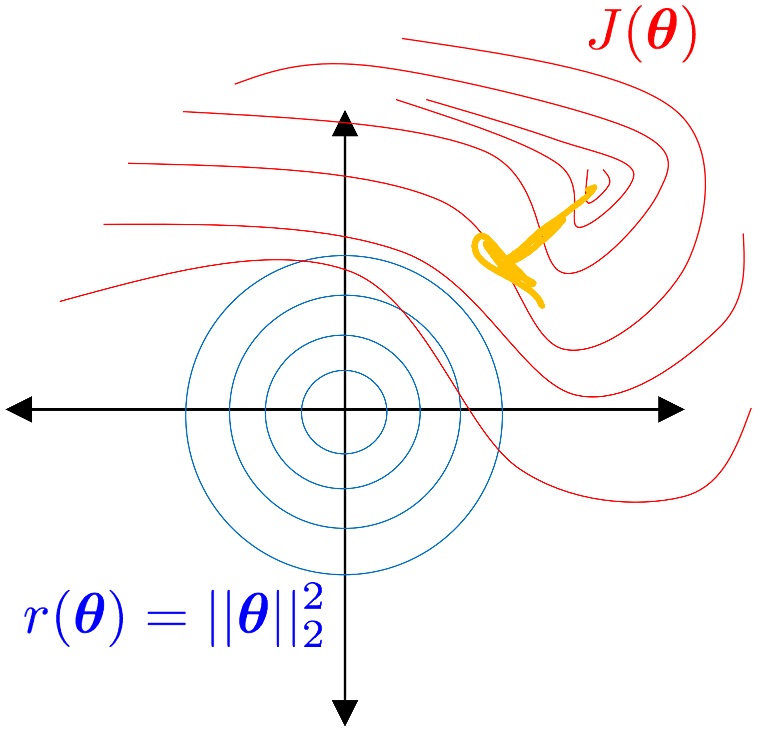
A. …increase

B. …decrease

C. …first increase, then decrease

D. …first decrease, then increase

E. …stay the same



Slide credit: CMU MLD Matt Gormley

Regularization

Don't Regularize the Bias (Intercept) Parameter

- In our models so far, the bias / intercept parameter is usually denoted by θ_0 -- that is, the parameter for which we fixed $x_0 = 1$
- Regularizers always avoid penalizing this bias / intercept parameter
- Why? Because otherwise the learning algorithms wouldn't be invariant to a shift in the y-values

Whitening Data

- It's common to *whiten* each feature by subtracting its mean and dividing by its variance
- For regularization, this helps all the features be penalized in the same units
(e.g. convert both centimeters and kilometers to z-scores)

Slide credit: CMU MLD Matt Gormley

Regularization

Given objective function: $J(\theta)$

Goal is to find:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\text{argmin}}\, J(\boldsymbol{\theta}) + \lambda r(\boldsymbol{\theta})$$

Key idea: Define regularizer $r(\boldsymbol{\theta})$ s.t. we tradeoff between fitting the data and keeping the model simple

Choose form of $r(\boldsymbol{\theta})$:

- Example: q-norm (usually p-norm)

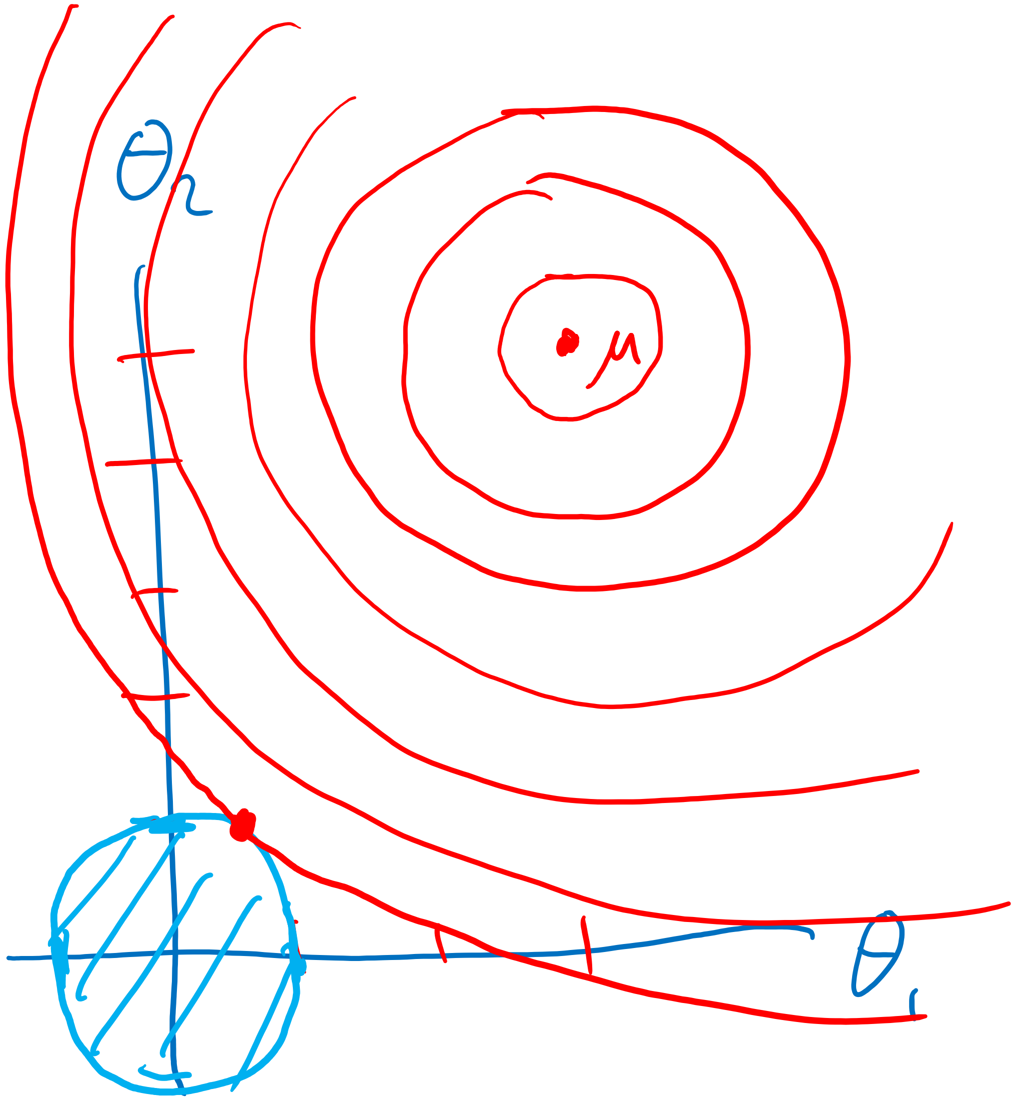
$$r(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_q = \left[\sum_{m=1}^{M} ||\theta_m||^q\right]^{(\frac{1}{q})}$$

q	$r(\boldsymbol{\theta})$	**yields parameters that are...**	**name**	**optimization notes**
0	$\|\|\boldsymbol{\theta}\|\|_0 = \sum \mathbb{1}(\theta_m \neq 0)$	zero values	L0 reg.	no good computational solutions
1	$\|\|\boldsymbol{\theta}\|\|_1 = \sum \|\theta_m\|$	zero values	L1 reg.	subdifferentiable
2	$(\|\|\boldsymbol{\theta}\|\|_2)^2 = \sum \theta_m^2$	small values	L2 reg.	differentiable

not convex

Slide credit: CMU MLD Matt Gormley

Regularization

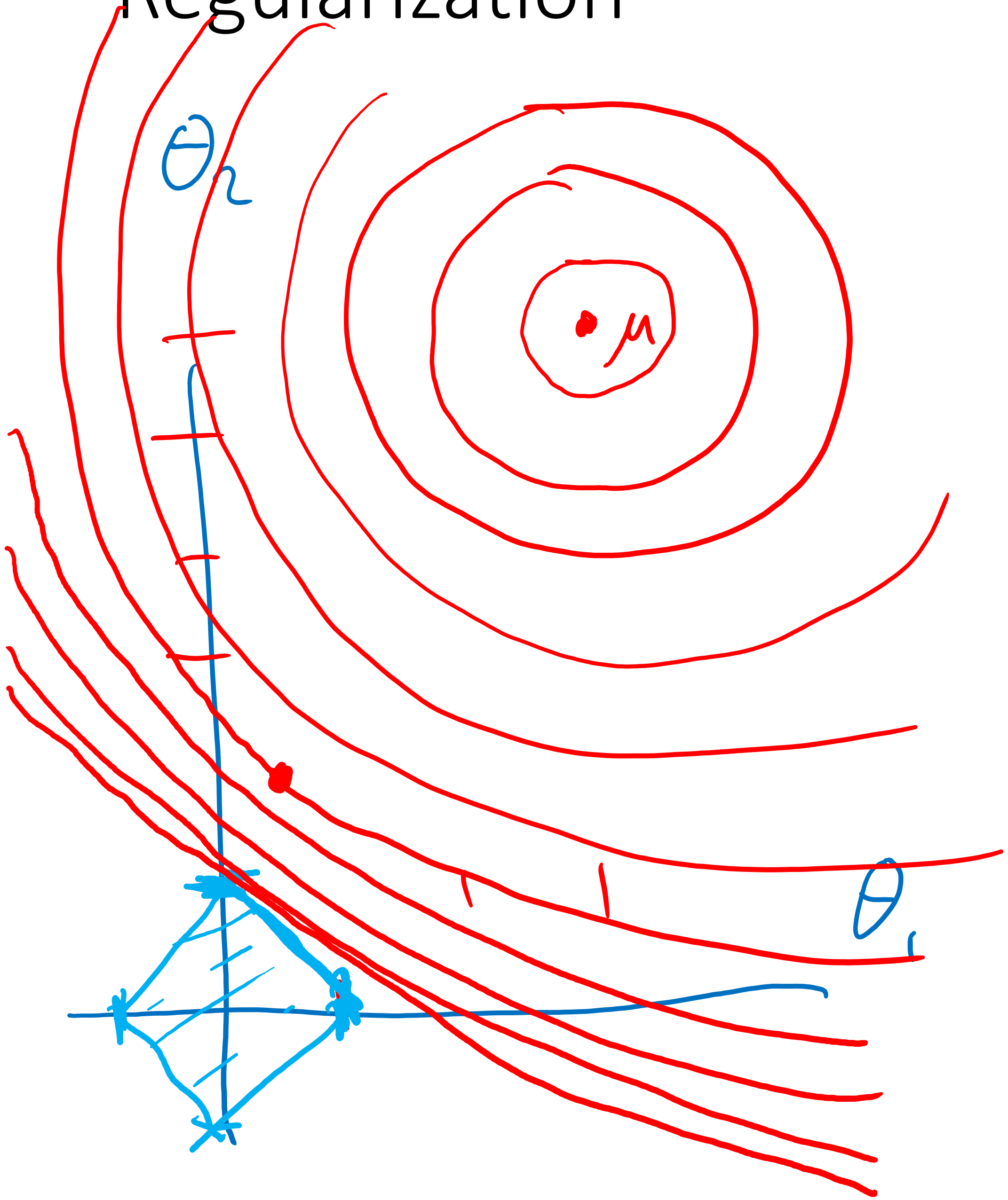


$$J(\theta_1, \theta_2) = \|\vec{\theta} - \vec{\mu}\| \qquad \mu = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

$$\min_{\theta} \; J(\theta_1, \theta_2)$$

$$\text{s.t.} \; \|\theta\|_2^2 \leq 1$$

Regularization



$$J(\theta_1, \theta_2) = \|\vec{\theta} - \vec{\mu}\| \qquad \mu = \begin{bmatrix} 3 \\ 5 \end{bmatrix}$$

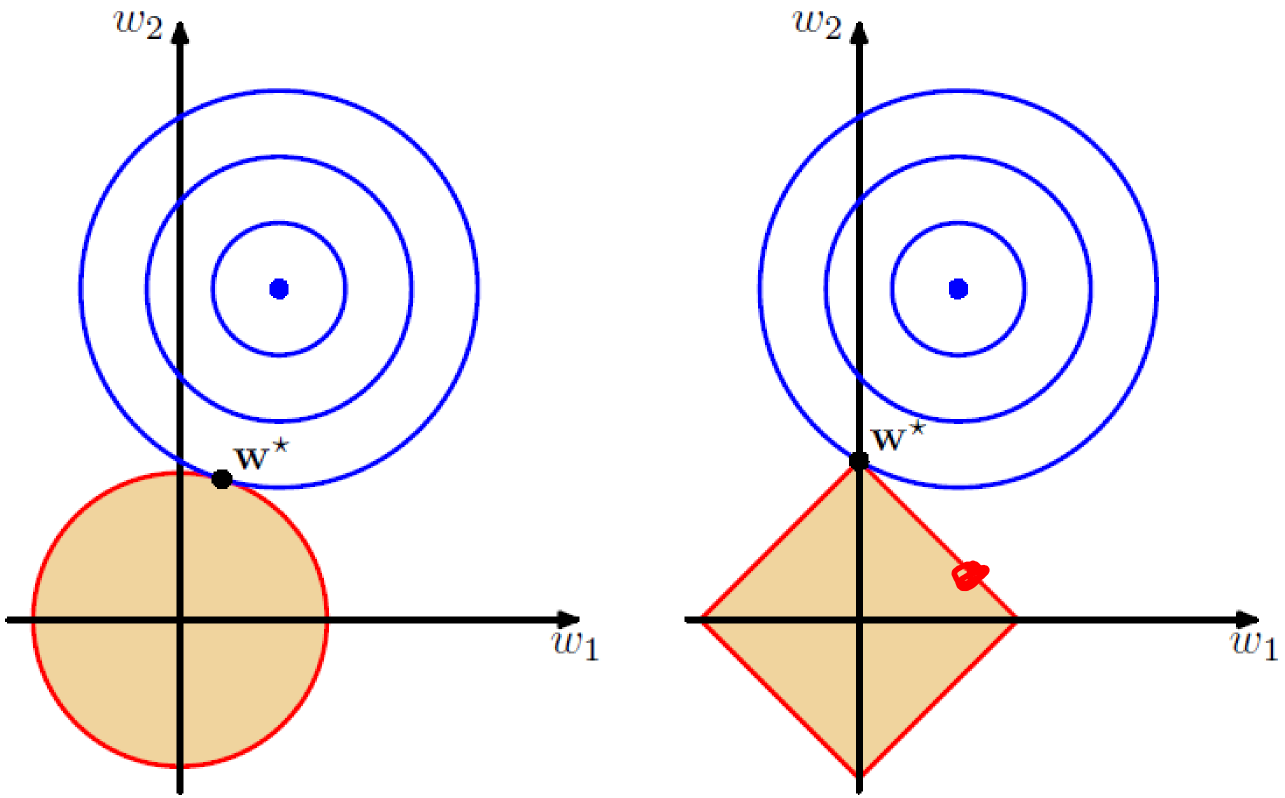
$$\min_{\theta} \ J(\theta_1, \theta_2)$$

$$\text{s.t.} \ \|\theta\|_1 \leq 1$$

$$|\theta_1| + |\theta_2|$$

L2 vs L1 Regularization

Combine original objective with penalty on parameters



Figures: Bishop, Ch 3.1.4

L2 vs L1: Housing Price Example

Predict housing price from several features

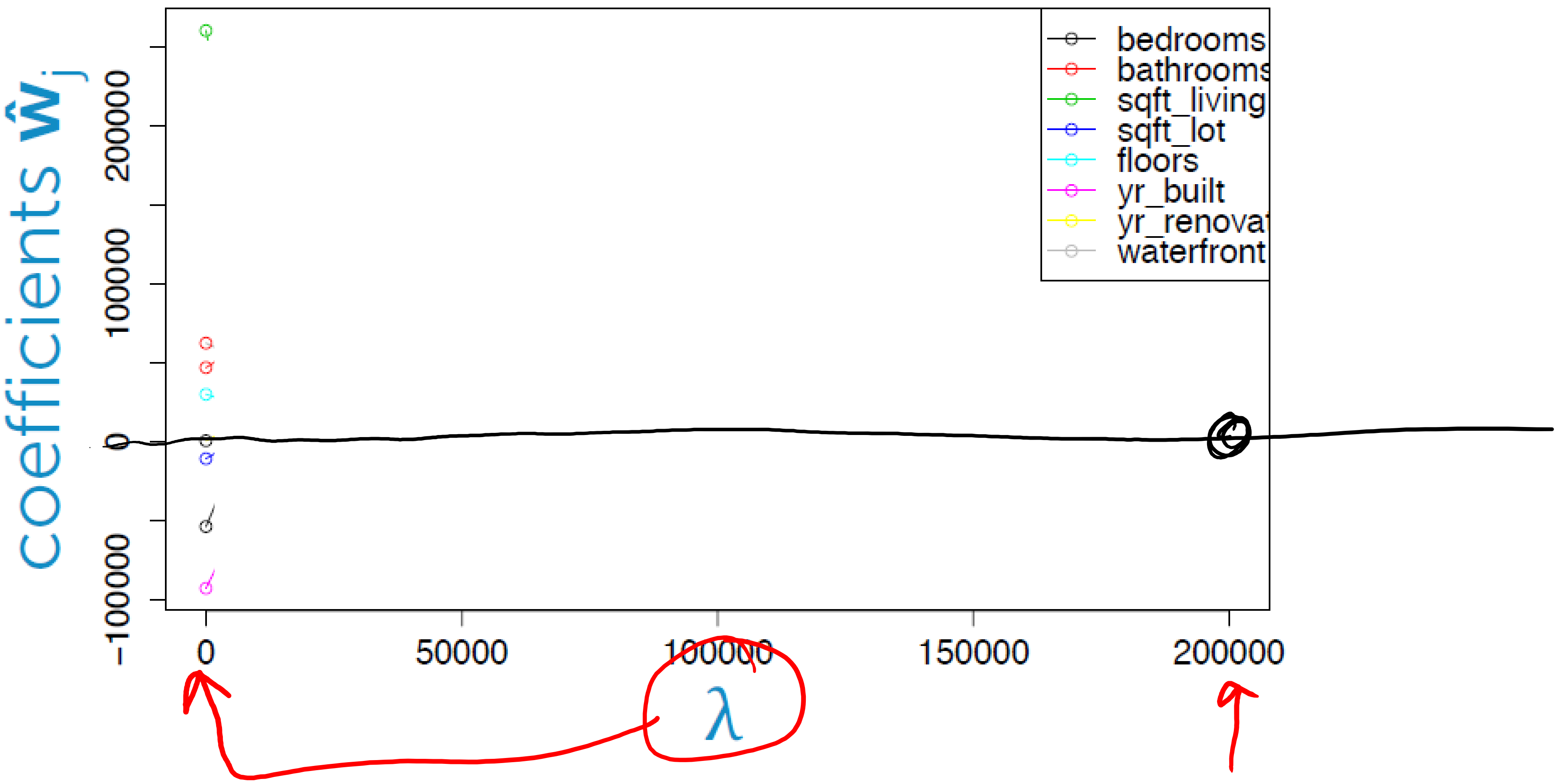


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

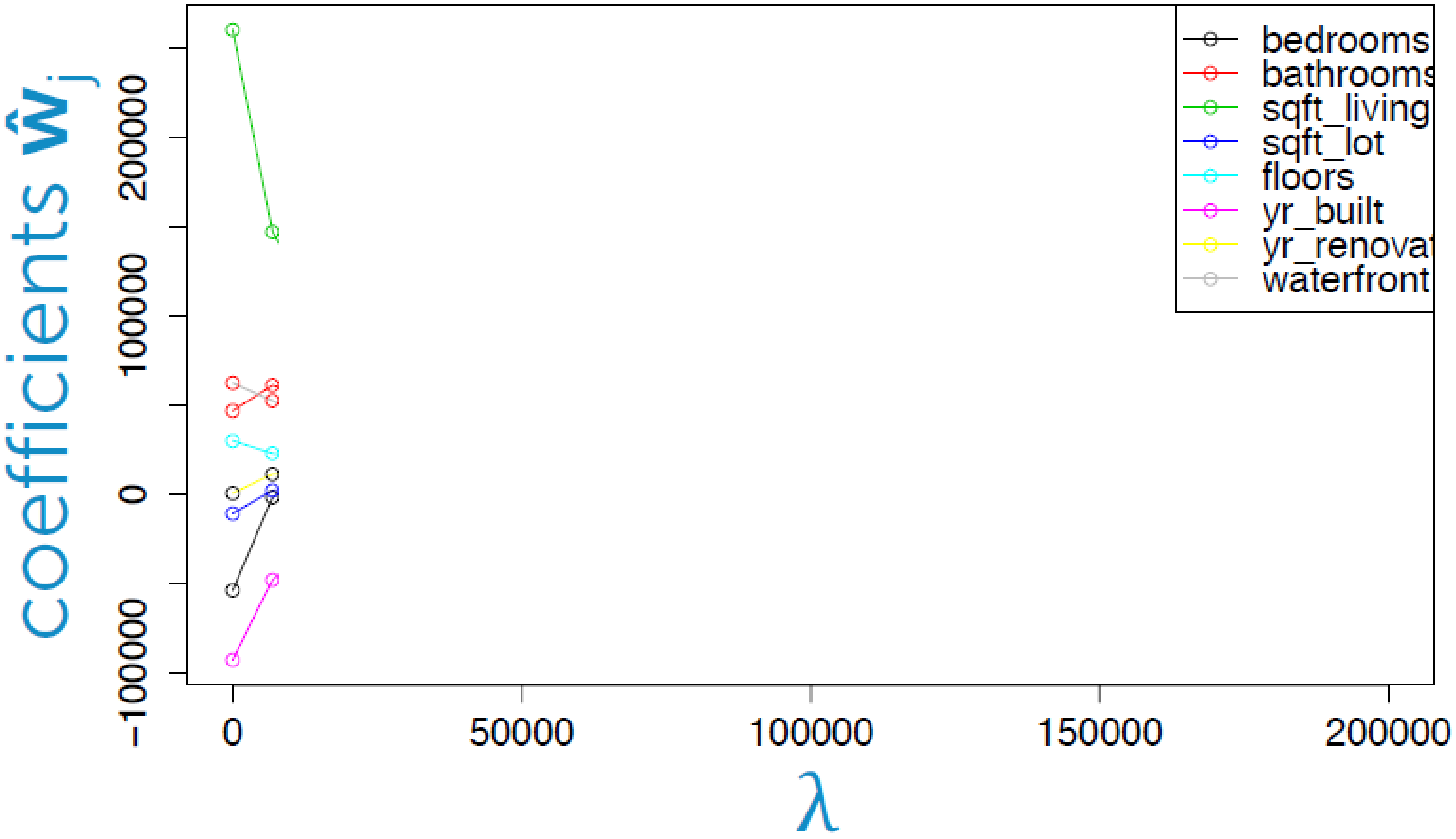


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

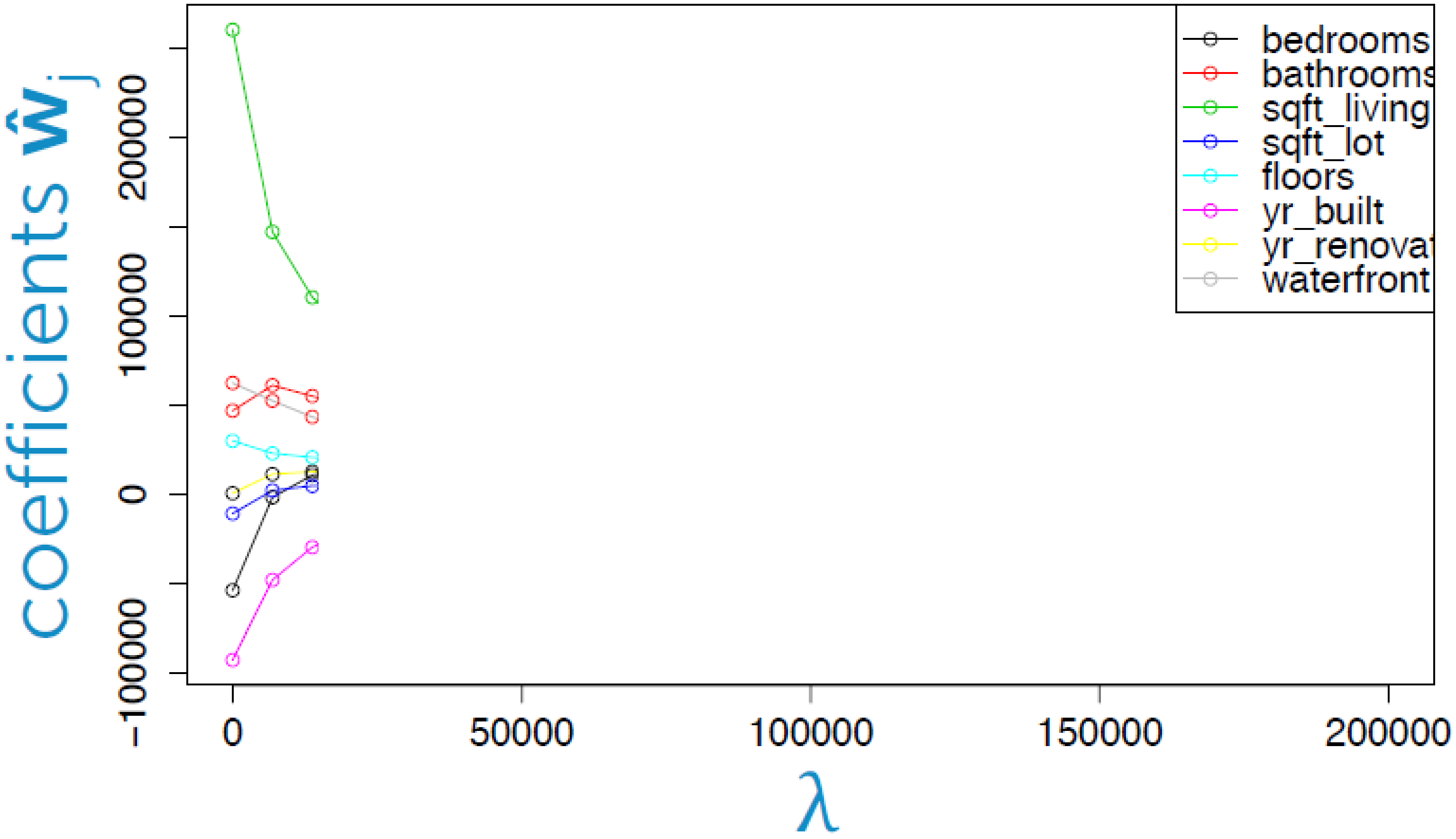


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

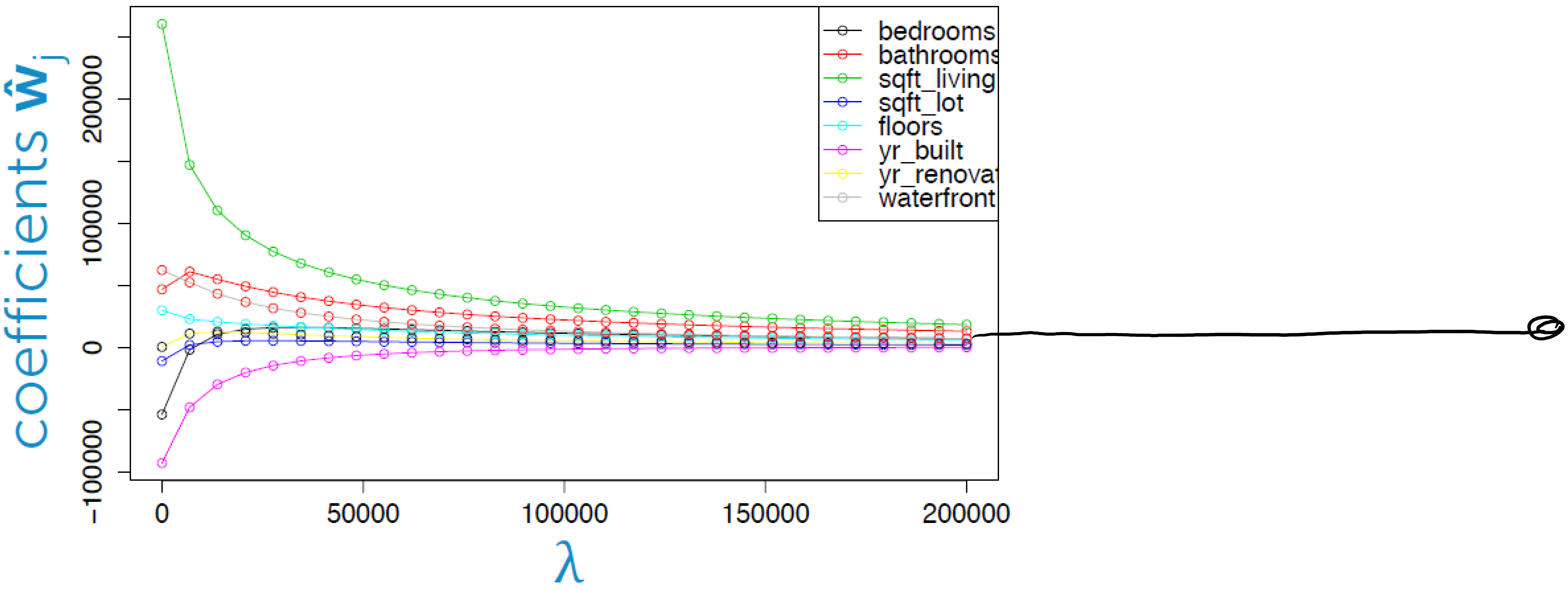


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

$J(\theta) + \lambda \|\vec{w}\|_1$

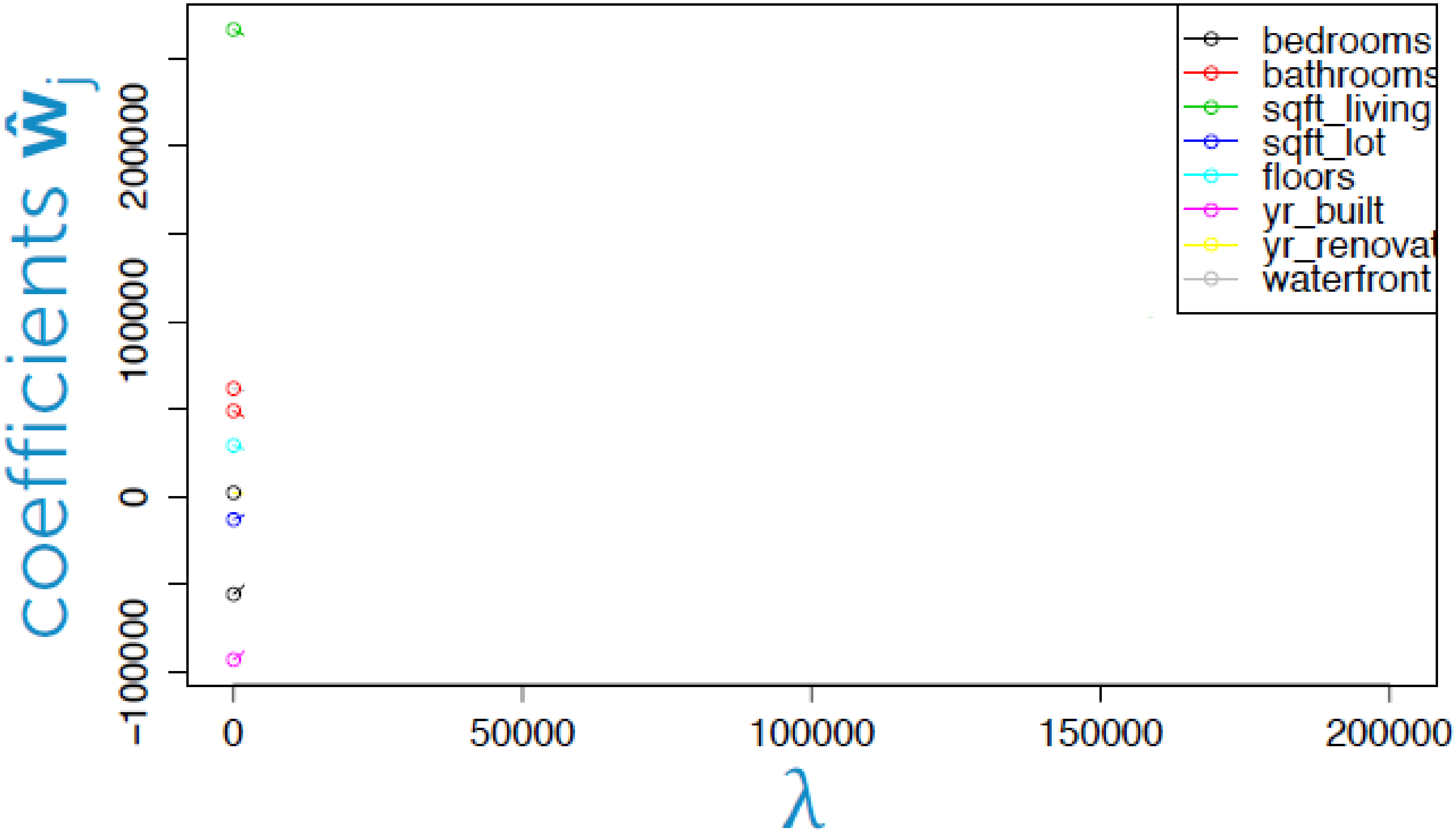


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

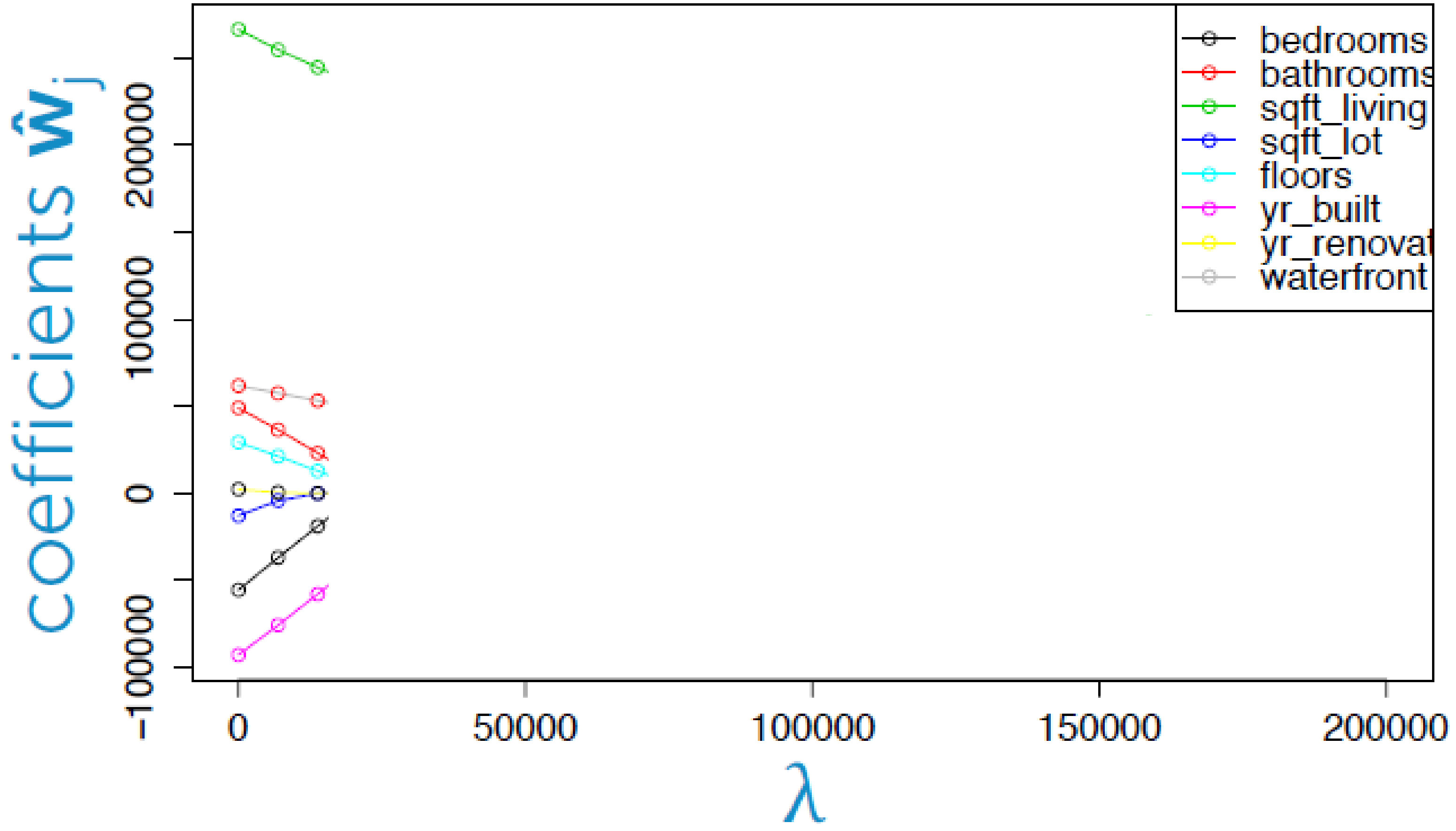


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

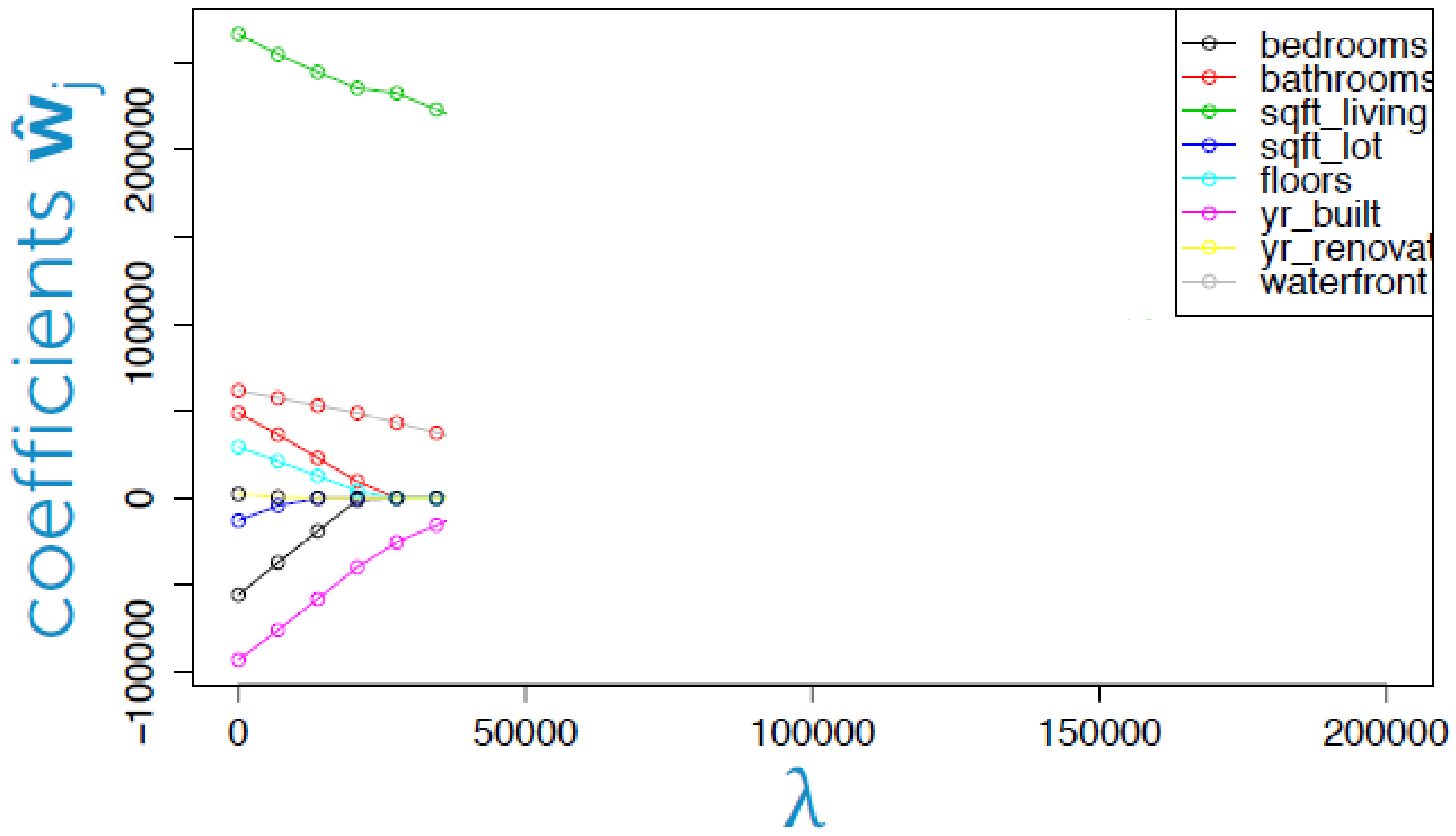


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

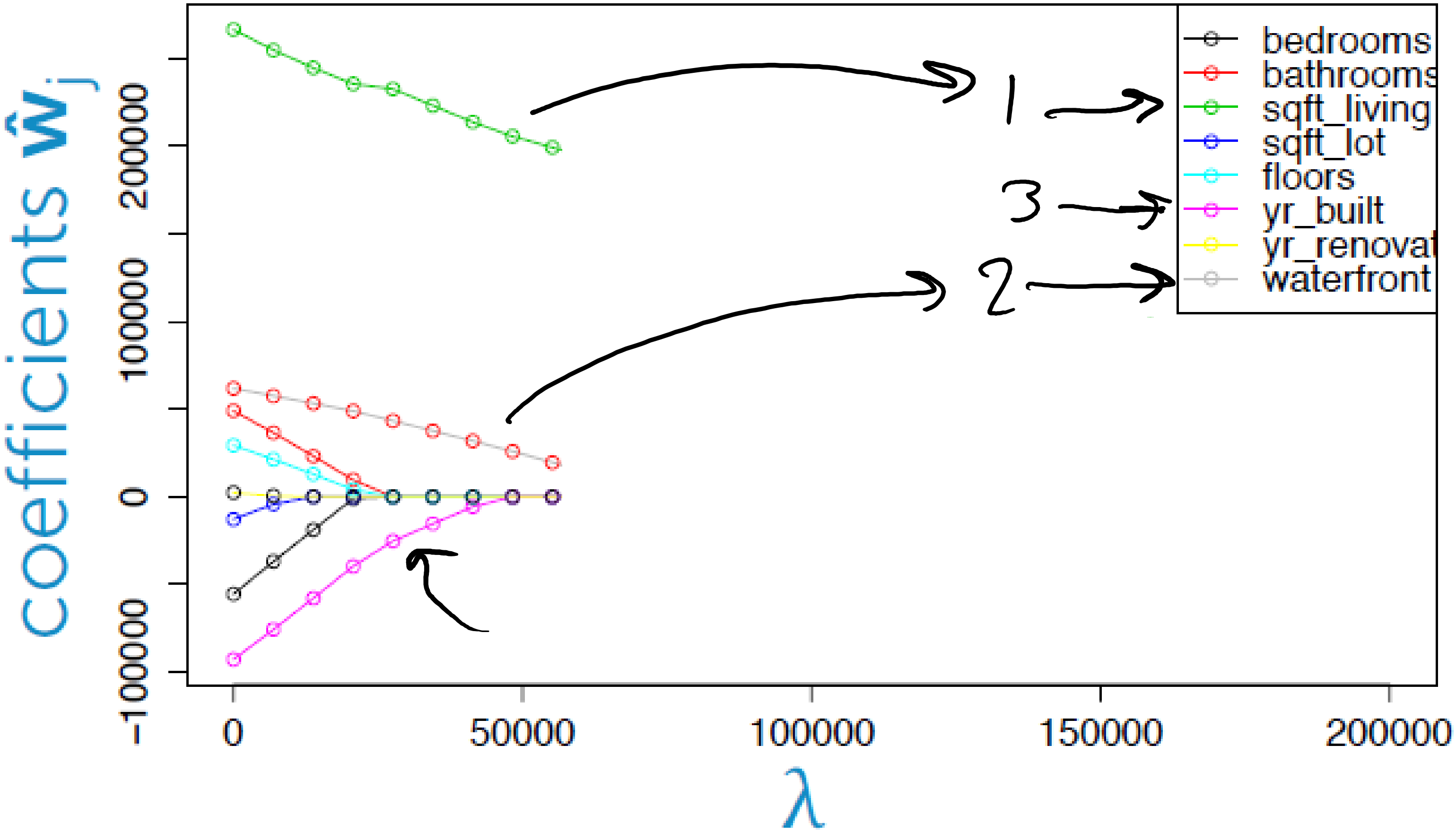


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

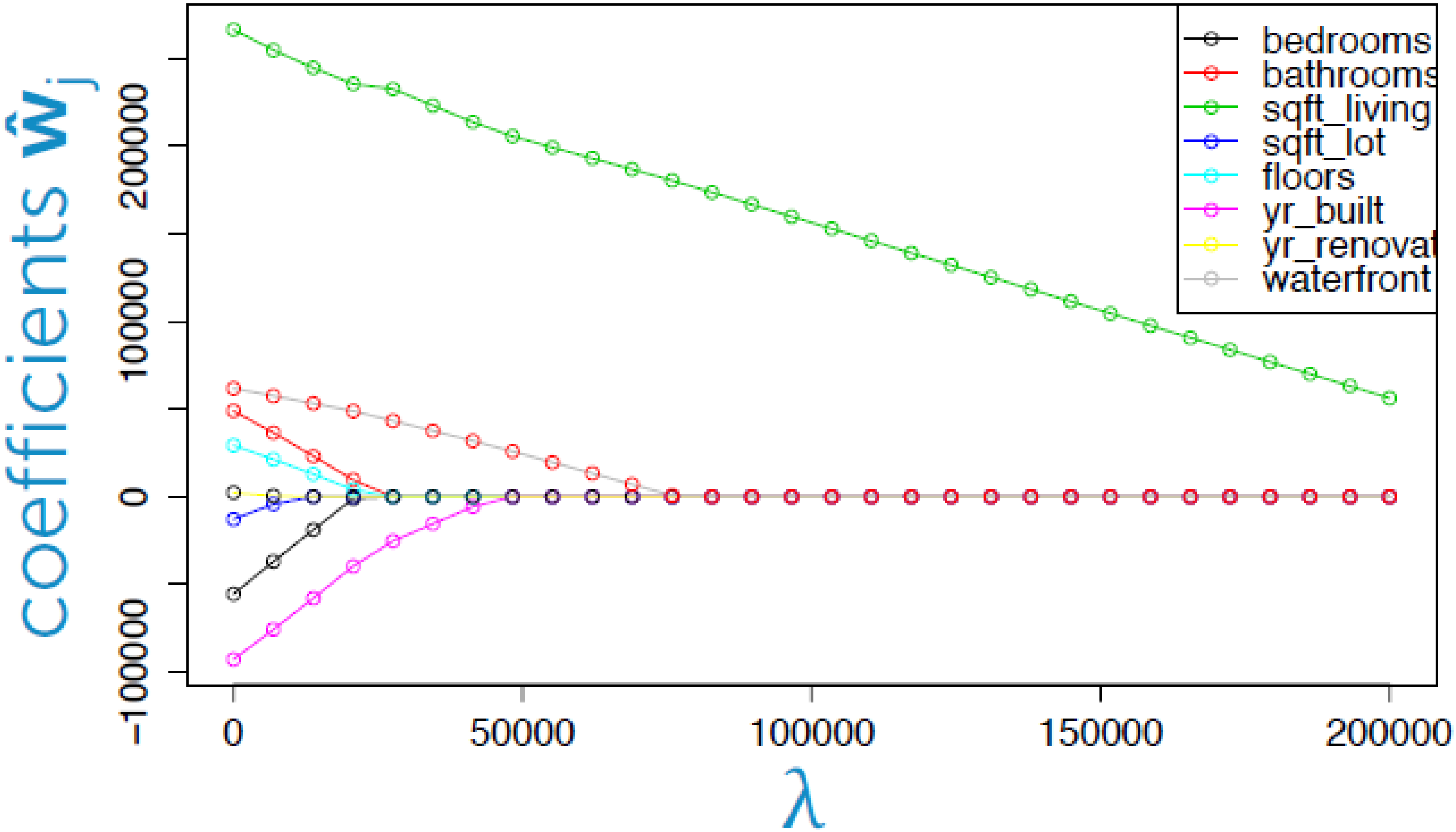


Figure: Emily Fox, University of Washington

Regularization as MAP

L1 and L2 regularization can be interpreted as **maximum a-posteriori (MAP) estimation** of the parameters

To be discussed later in the course…

Optimization

Takeaways

1. **Nonlinear basis functions** allow **linear models** (e.g. Linear Regression, Logistic Regression) to capture **nonlinear** aspects of the original input
2. Nonlinear features **require no changes to the model** (i.e. just preprocessing)
3. **Regularization** helps to avoid **overfitting**
4. **(Regularization** and **MAP estimation** are equivalent for appropriately chosen priors)

Slide credit: CMU MLD Matt Gormley

Additional Slides

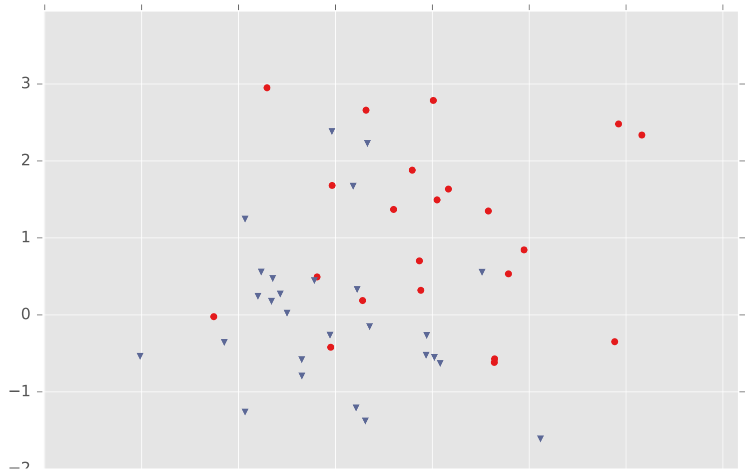
Logistic Regression with Nonlinear Features

Jupyter notebook demo

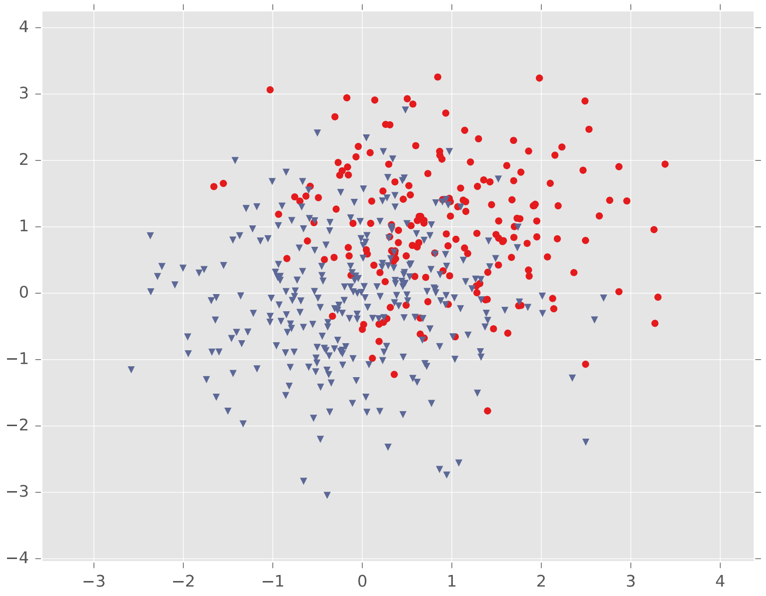
quad logist

Example: Logistic Regression

Training Data



Test Data



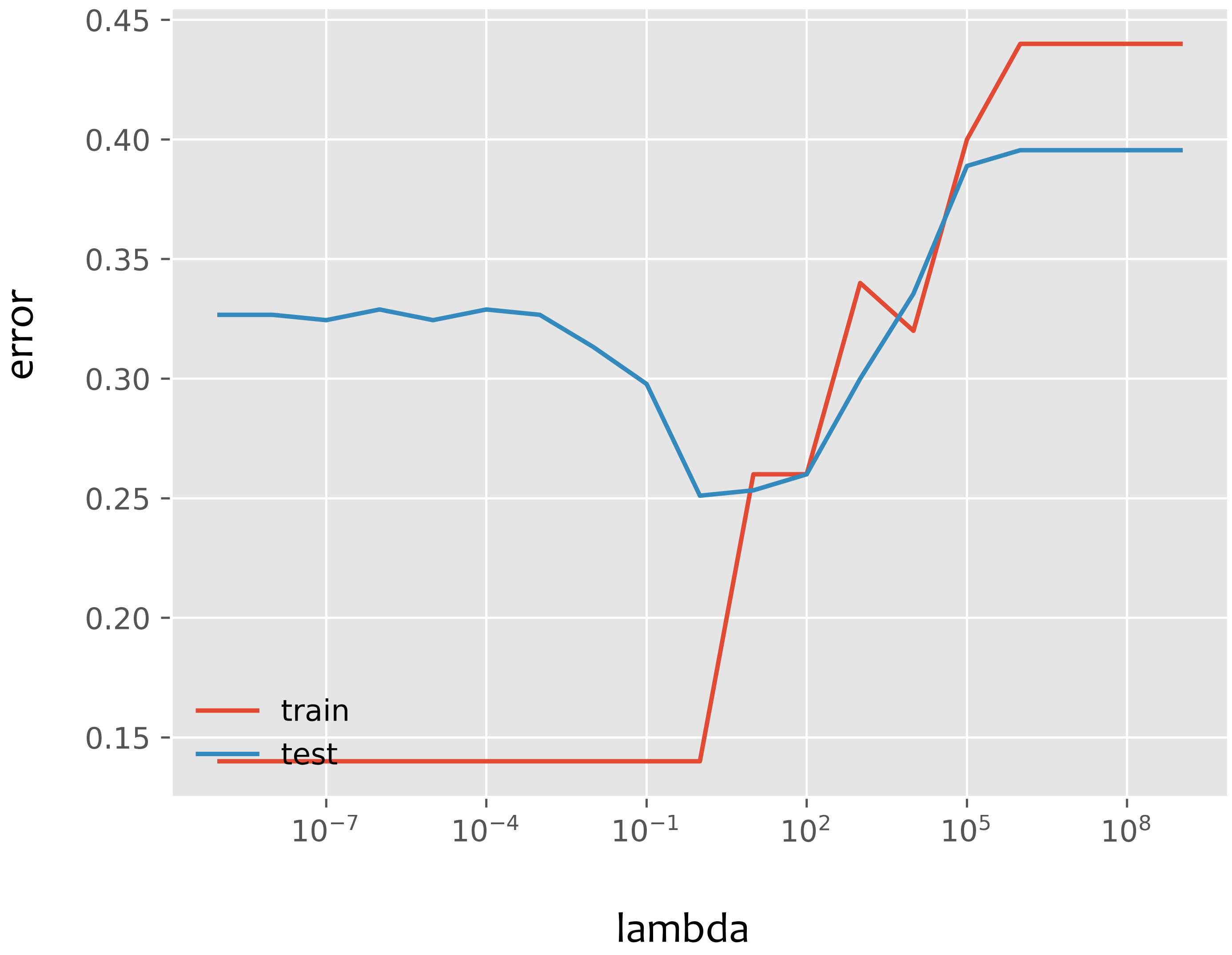
For this example, we construct **nonlinear features** (i.e. feature engineering)

Specifically, we add **polynomials up to order 9** of the two original features x_1 and x_2

Thus our classifier is **linear** in the **high-dimensional feature space**, but the decision boundary is **nonlinear** when visualized in **low-dimensions** (i.e. the original two dimensions)

Slide credit: CMU MLD Matt Gormley

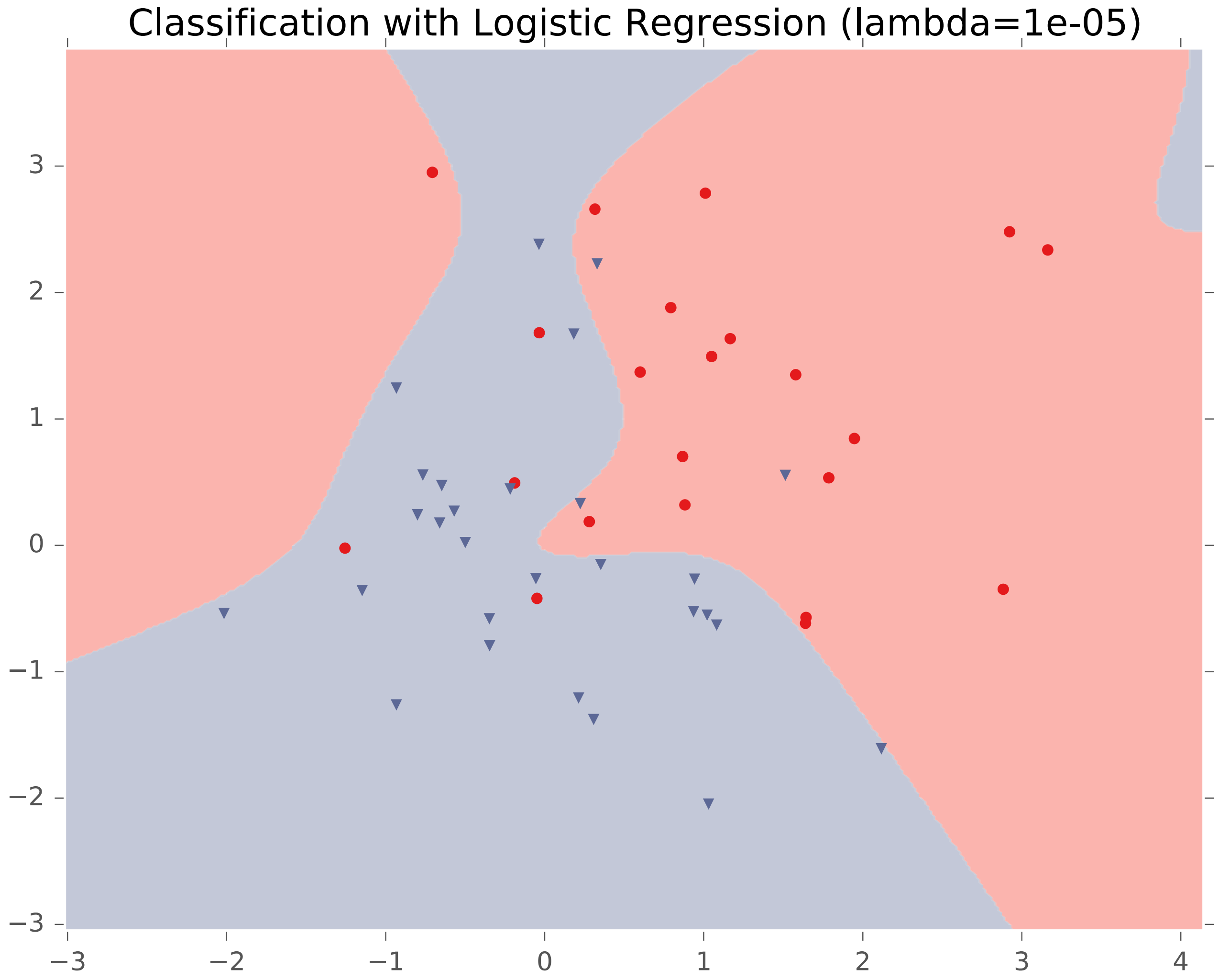
Example: Logistic Regression



Slide credit: CMU MLD Matt Gormley

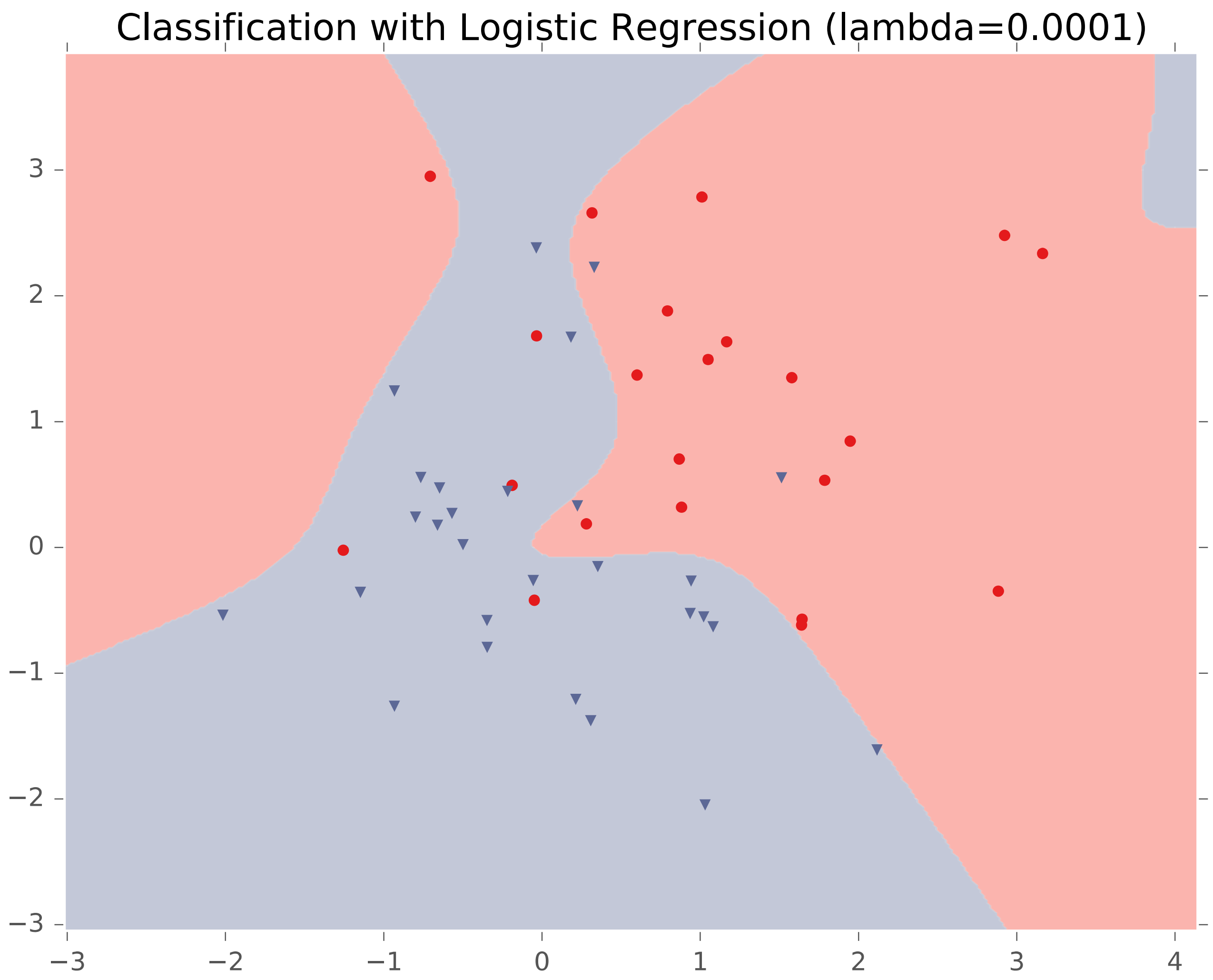
Example: Logistic Regression

✓ 1e-5

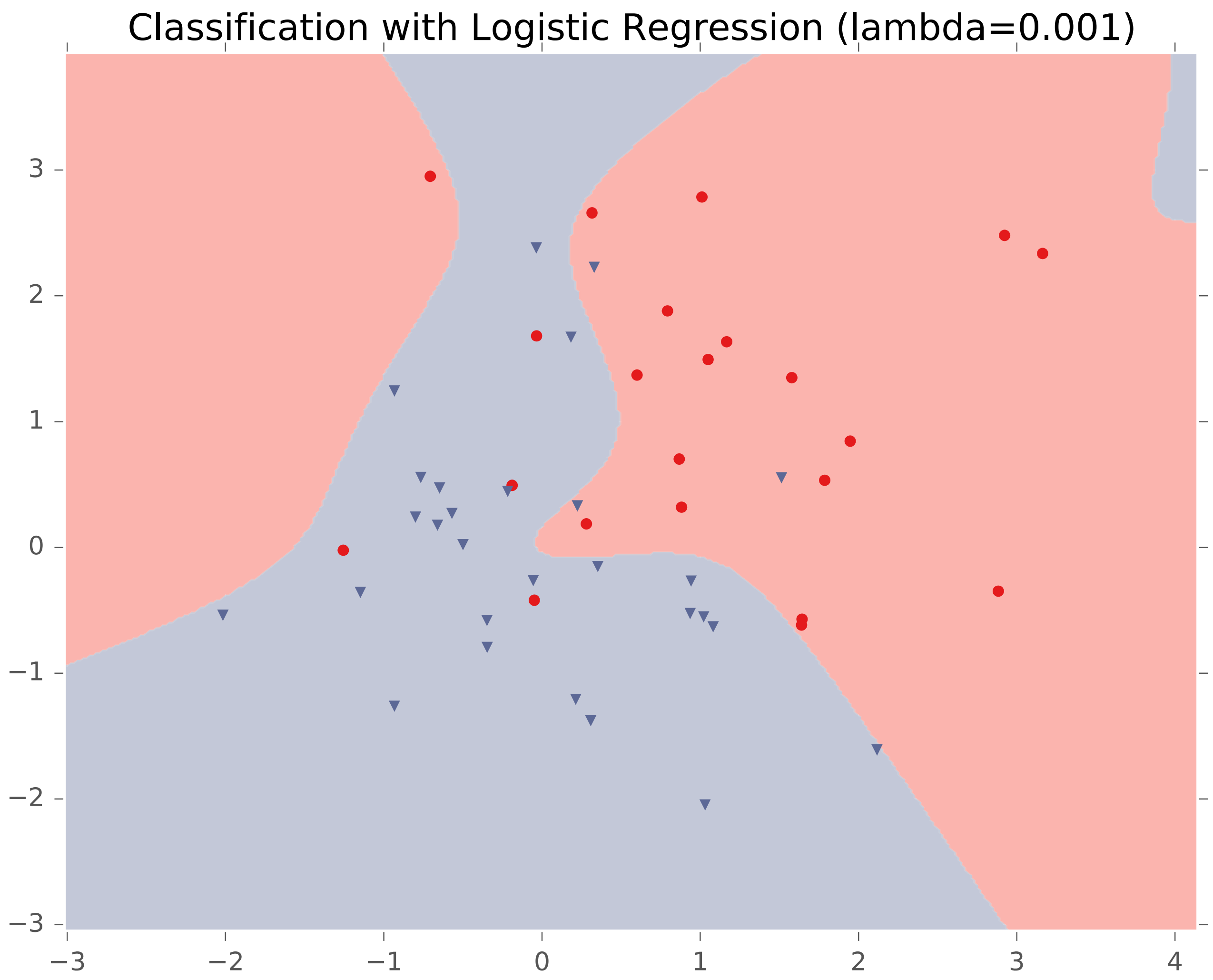


Slide credit: CMU MLD Matt Gormley

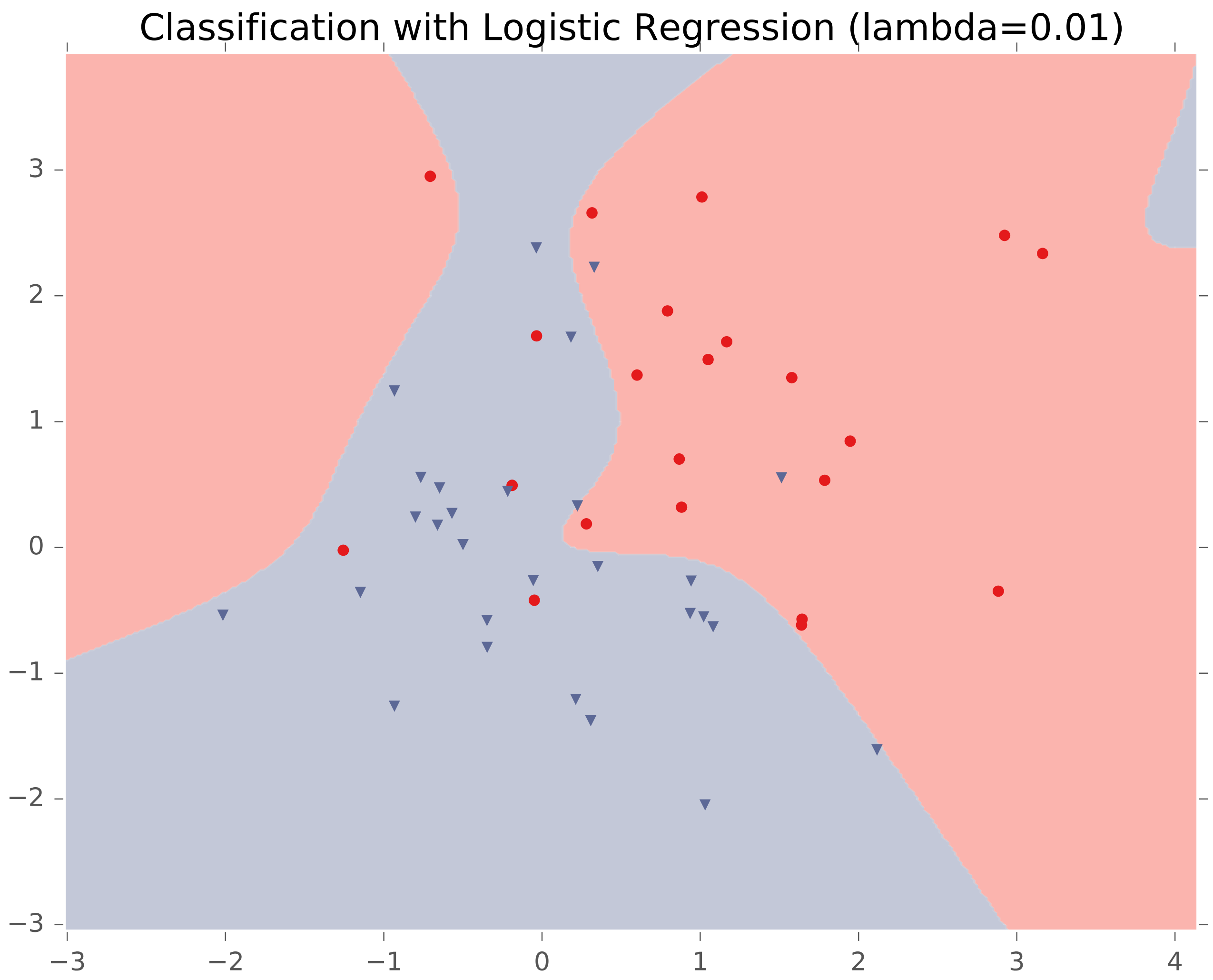
Example: Logistic Regression



Example: Logistic Regression

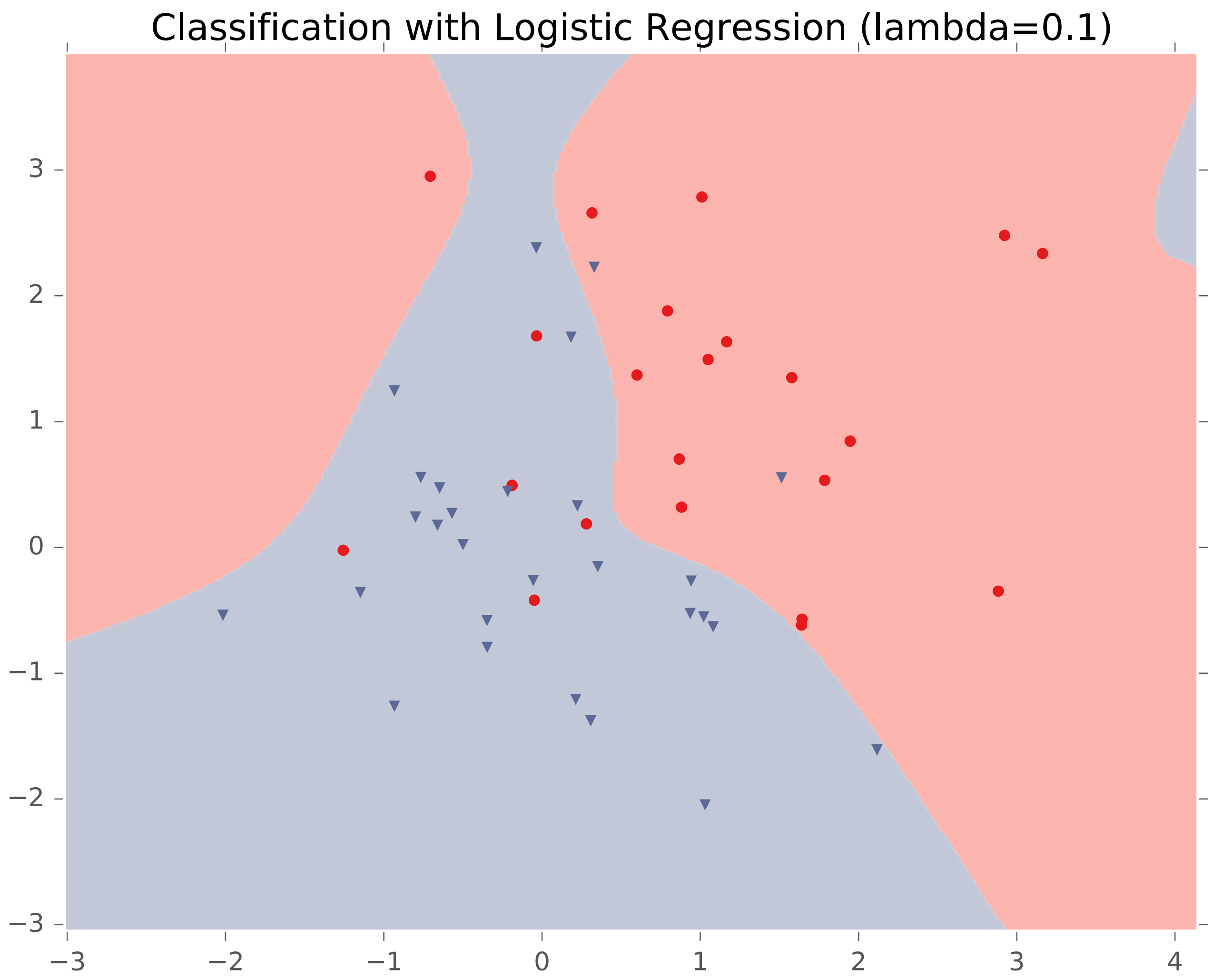


Example: Logistic Regression

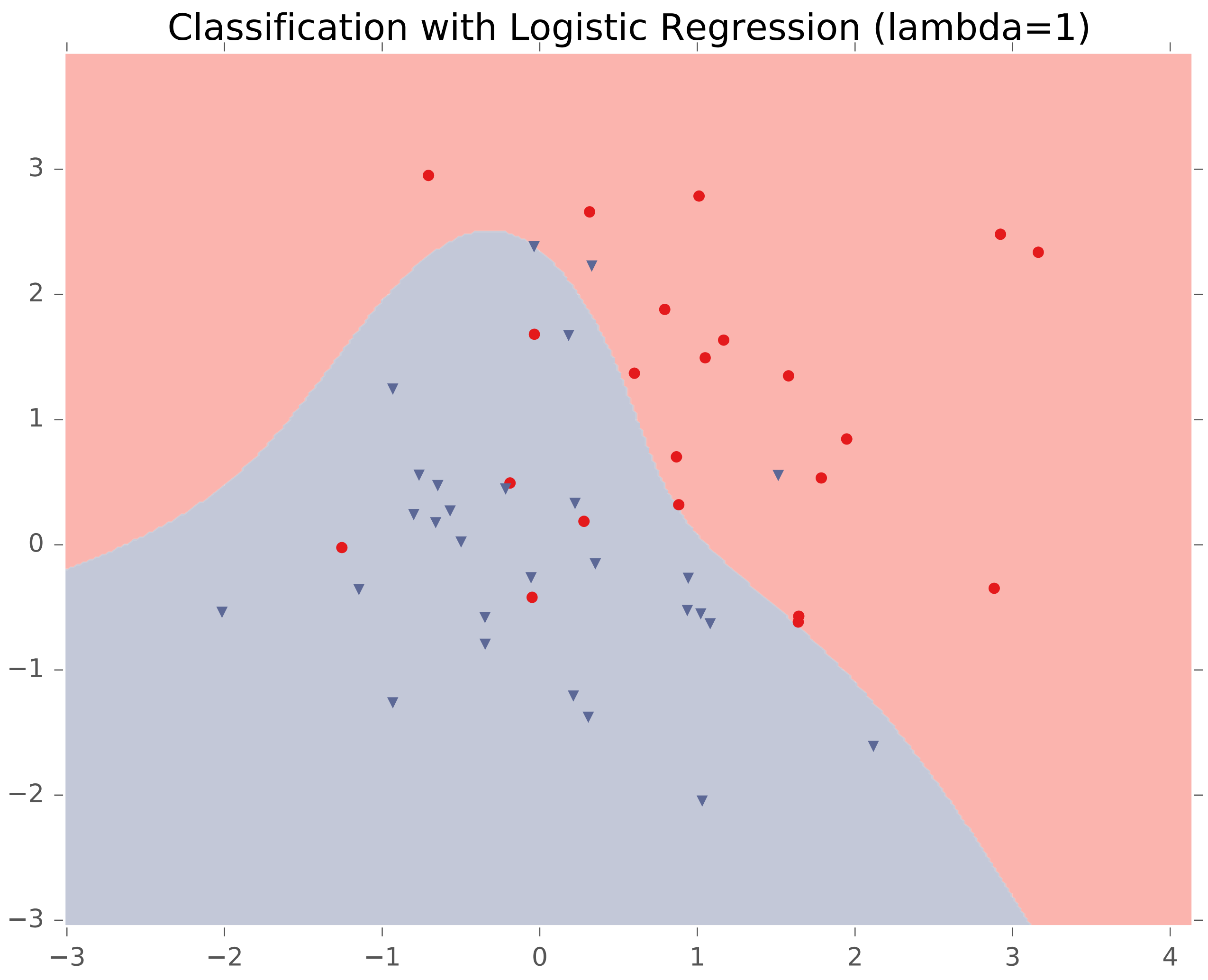


Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

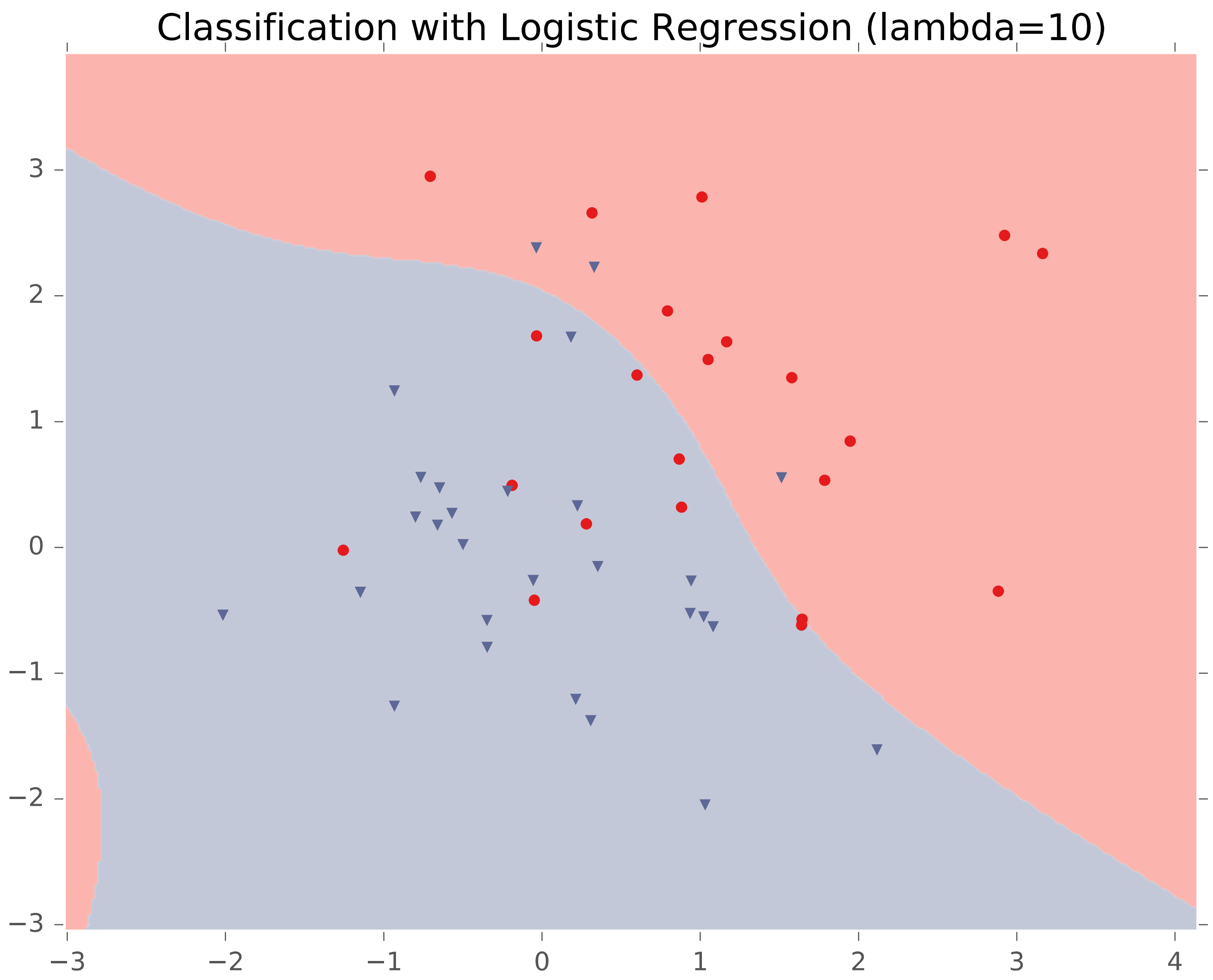


Example: Logistic Regression



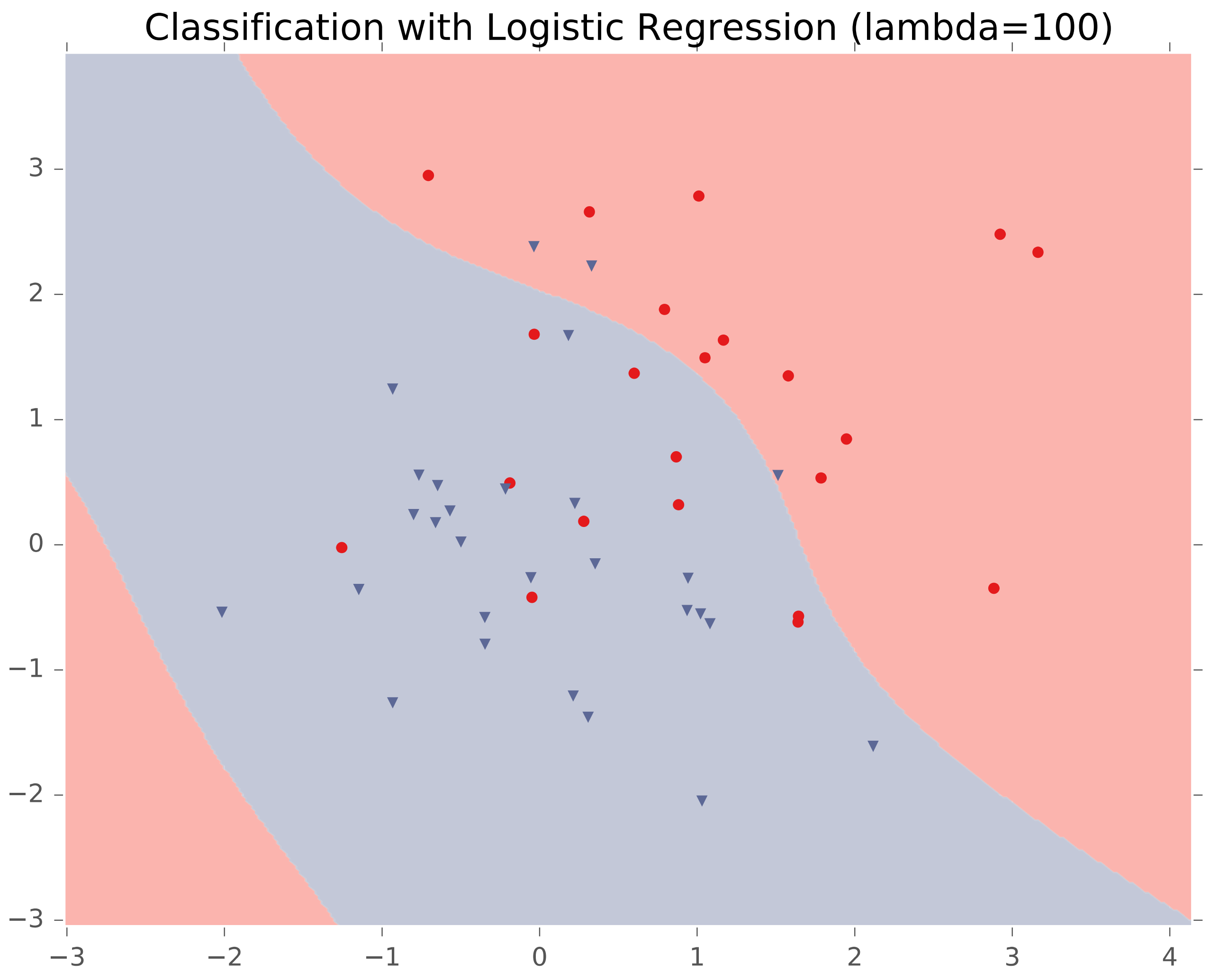
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



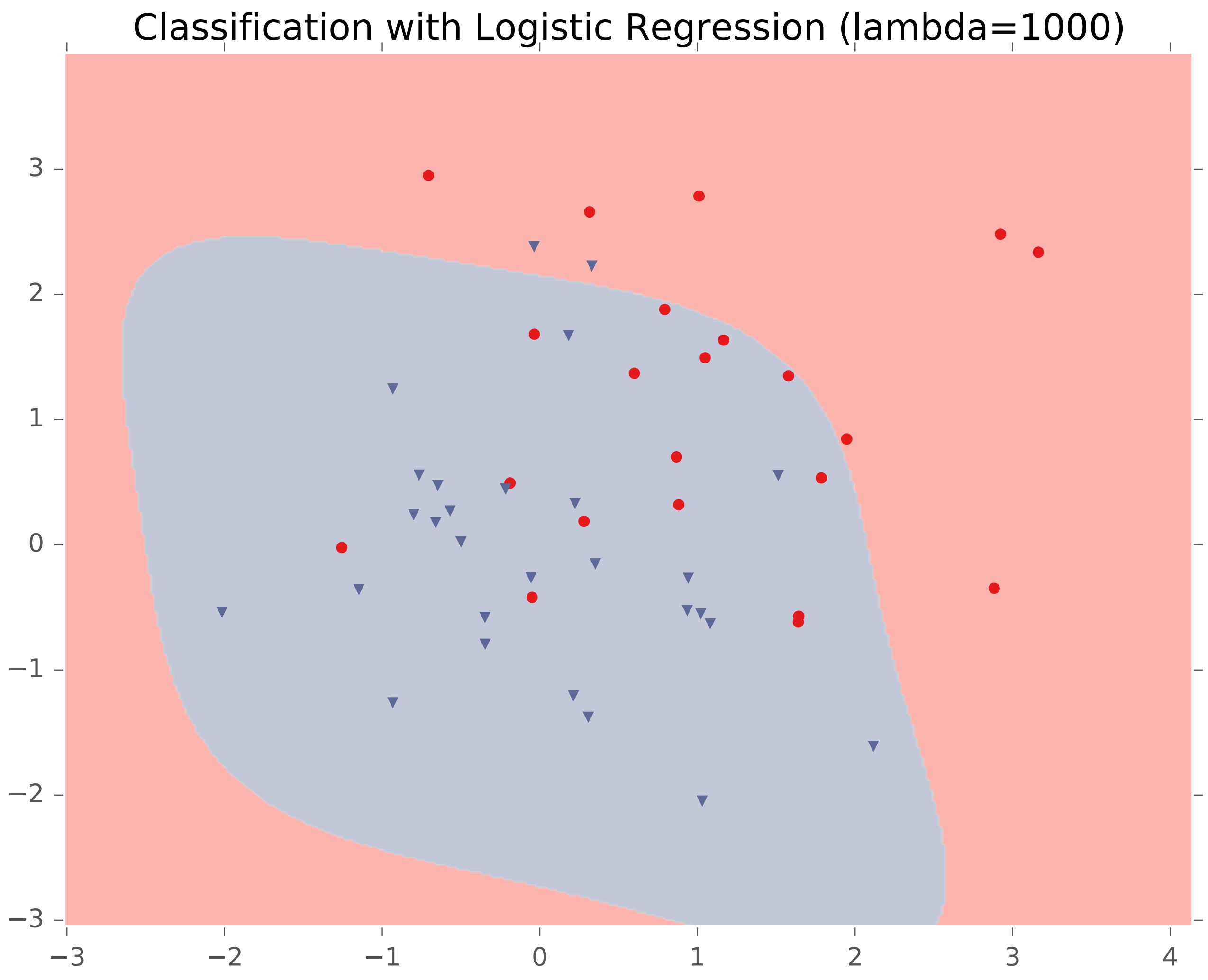
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



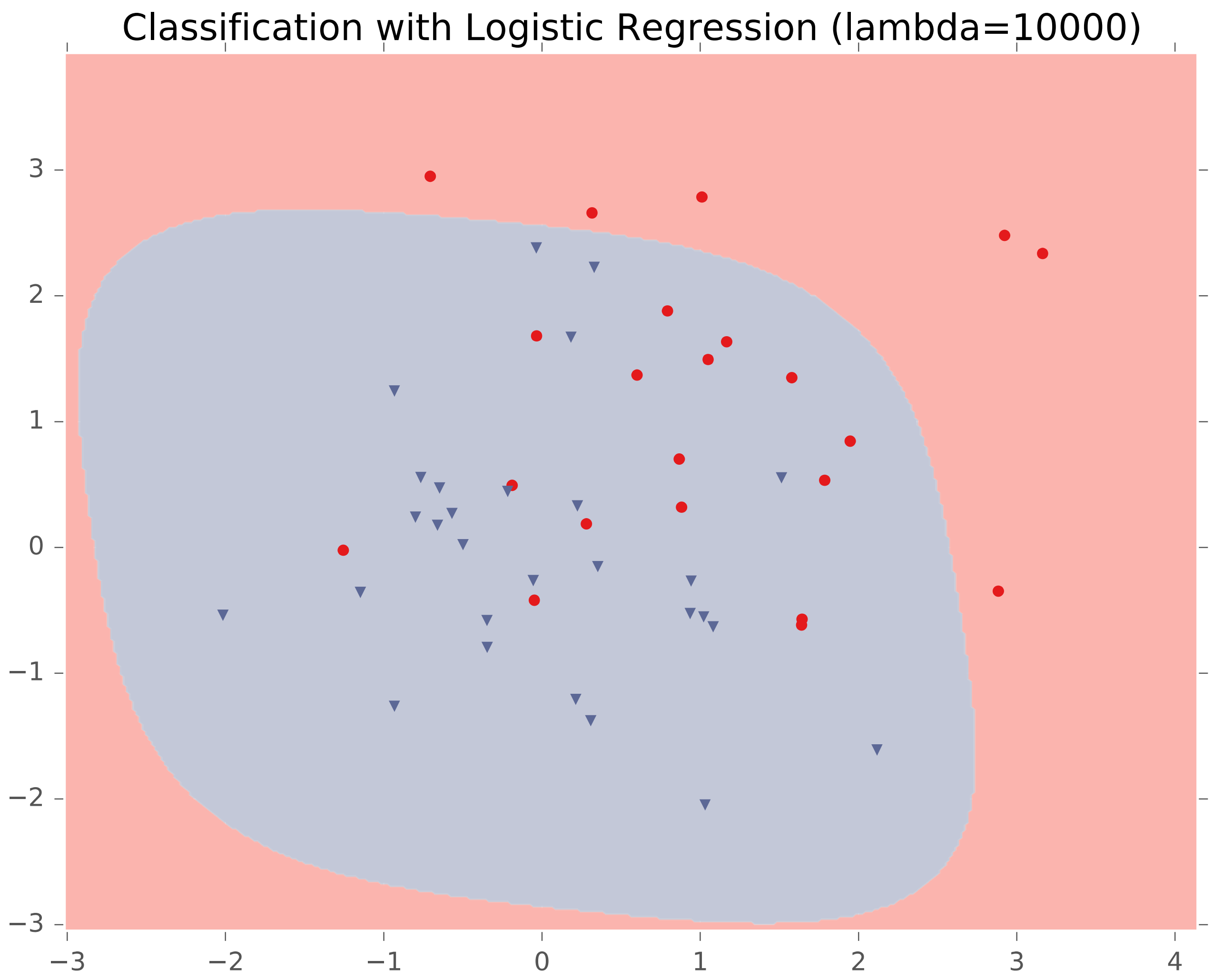
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



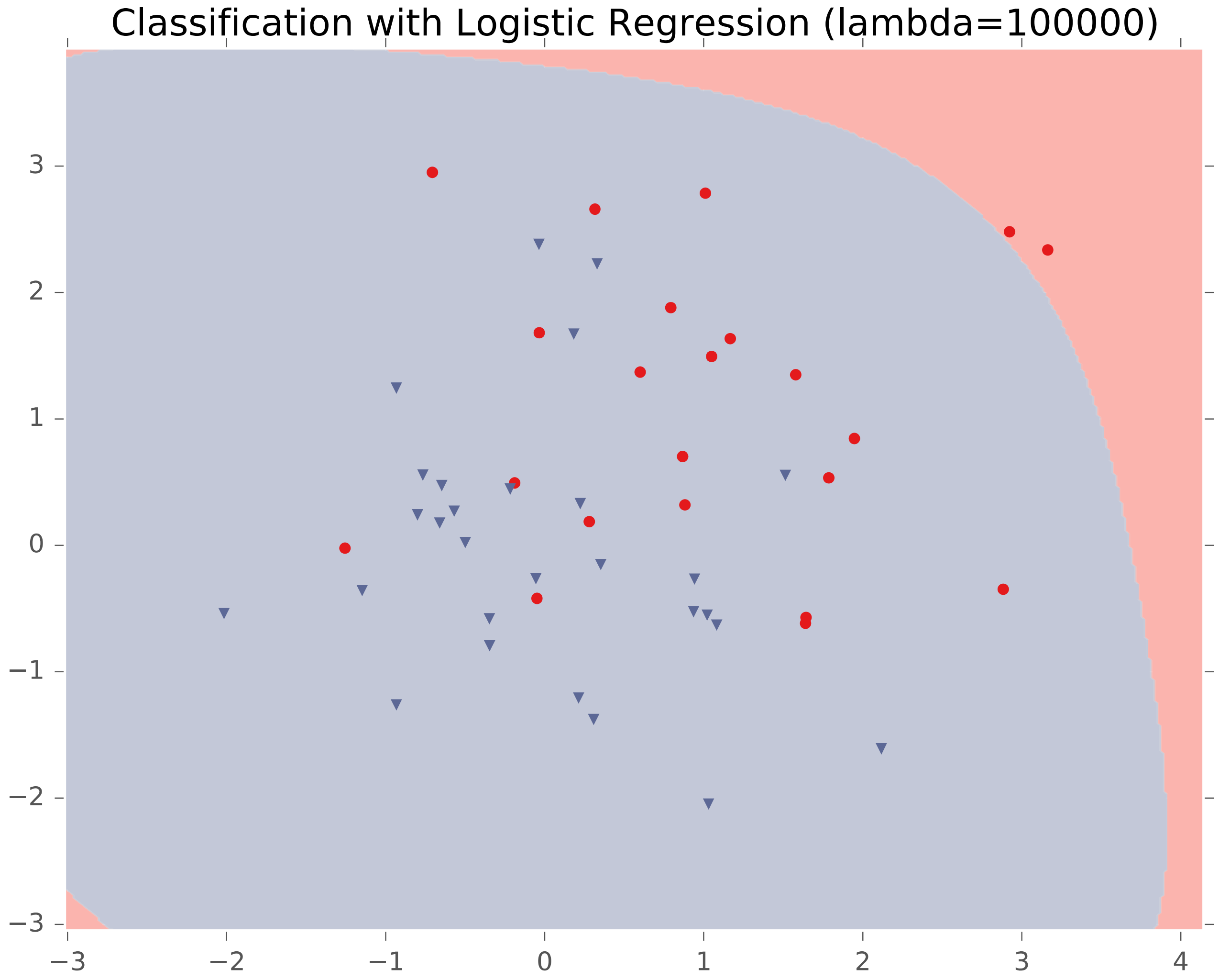
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



Slide credit: CMU MLD Matt Gormley

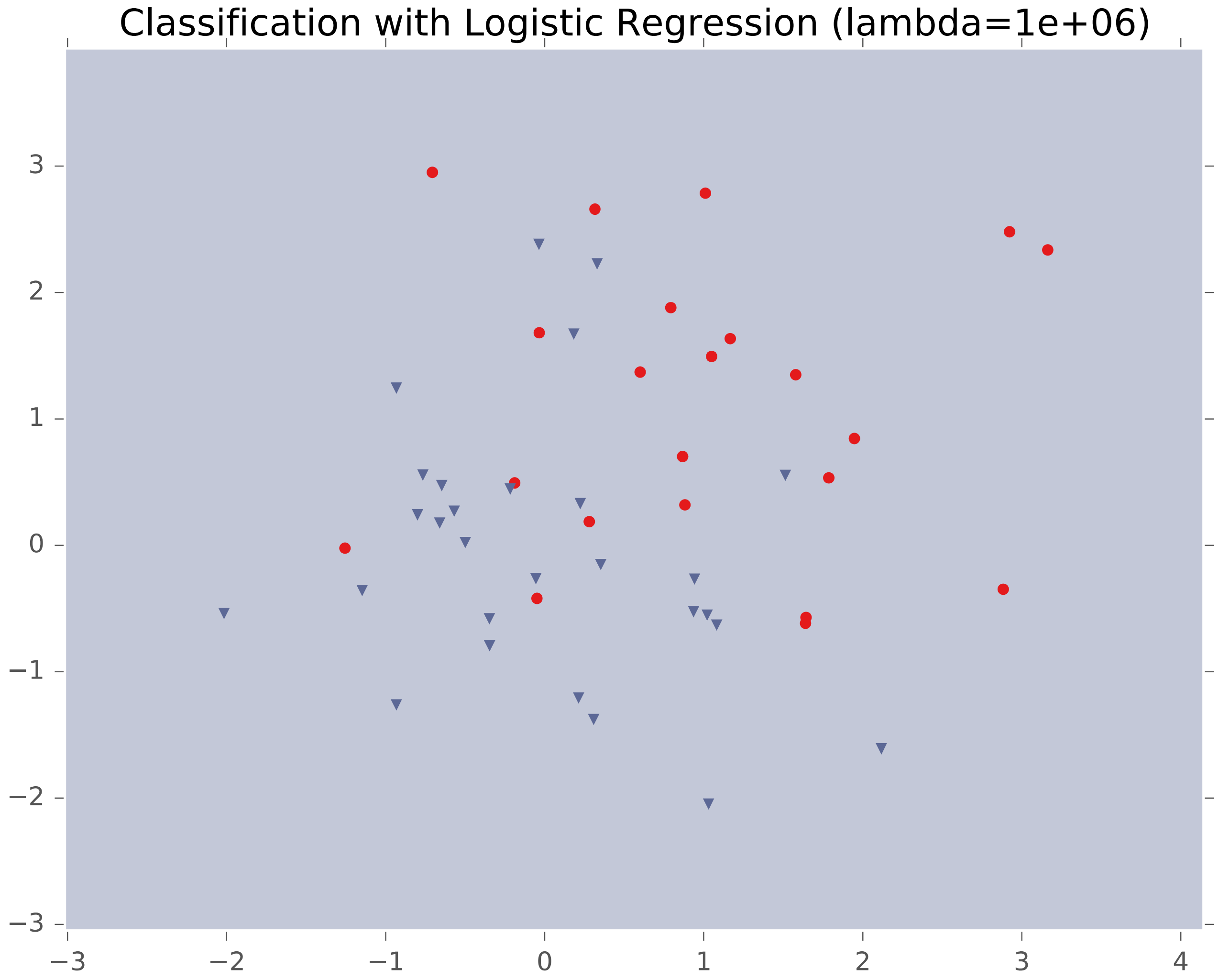
Example: Logistic Regression



Slide credit: CMU MLD Matt Gormley

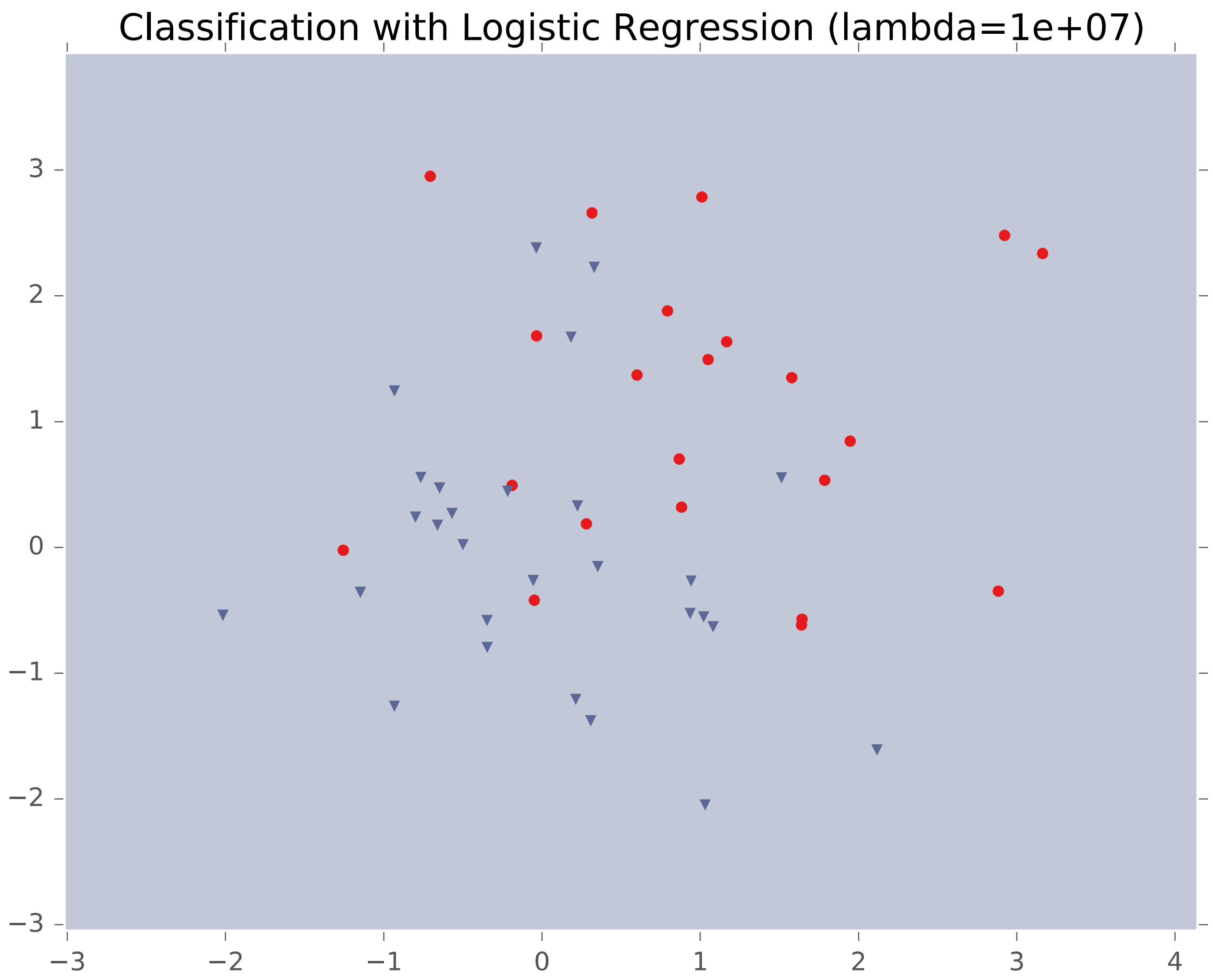
Example: Logistic Regression

1e6



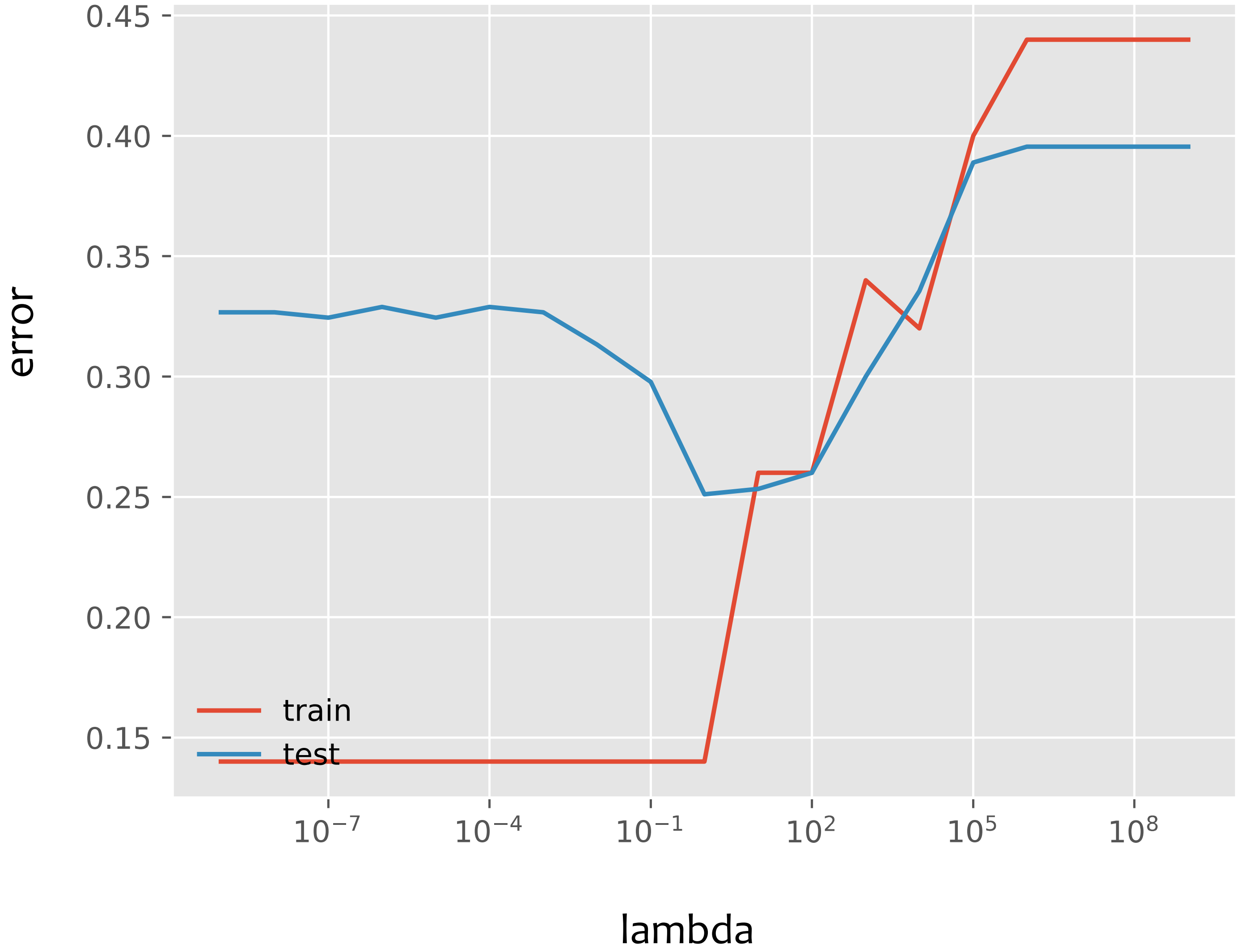
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



Slide credit: CMU MLD Matt Gormley