10-315
Introduction to ML

Regularization

Instructor: Pat Virtue

Plan

Today

- Regularization
  - (Make sure they aren't too powerful ☺)
  - Regularization with L2 norm
  - Regularization optimization
  - Regularization with L1 norm

lin reg

Regularization with L2 norm

Example: Linear regression with polynomial features

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):

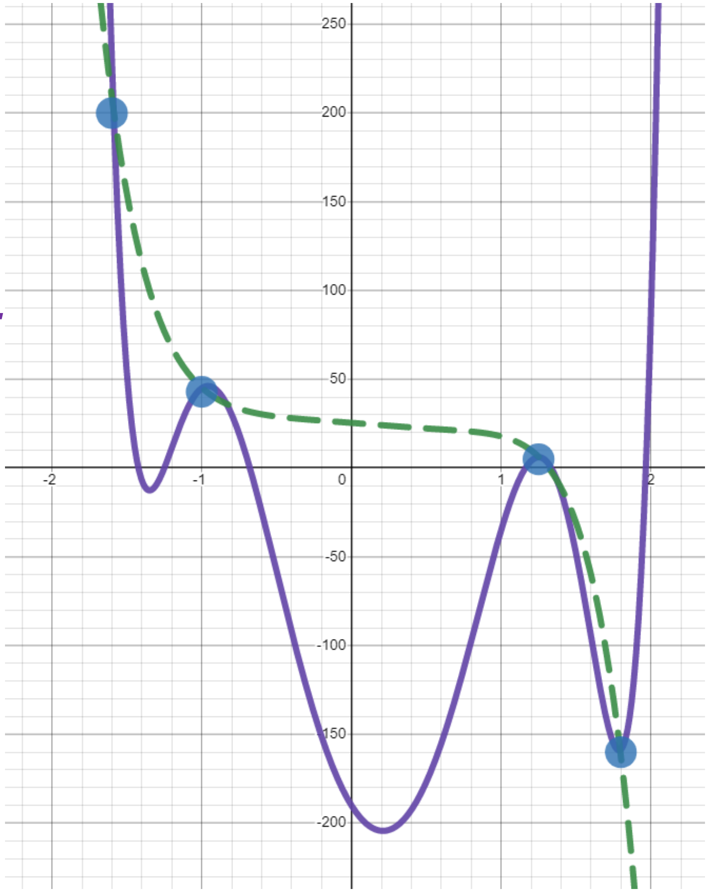
$$h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5 + \theta_6 x^6 + \theta_7 x^7 + \theta_8 x^8$$

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



Overfitting

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

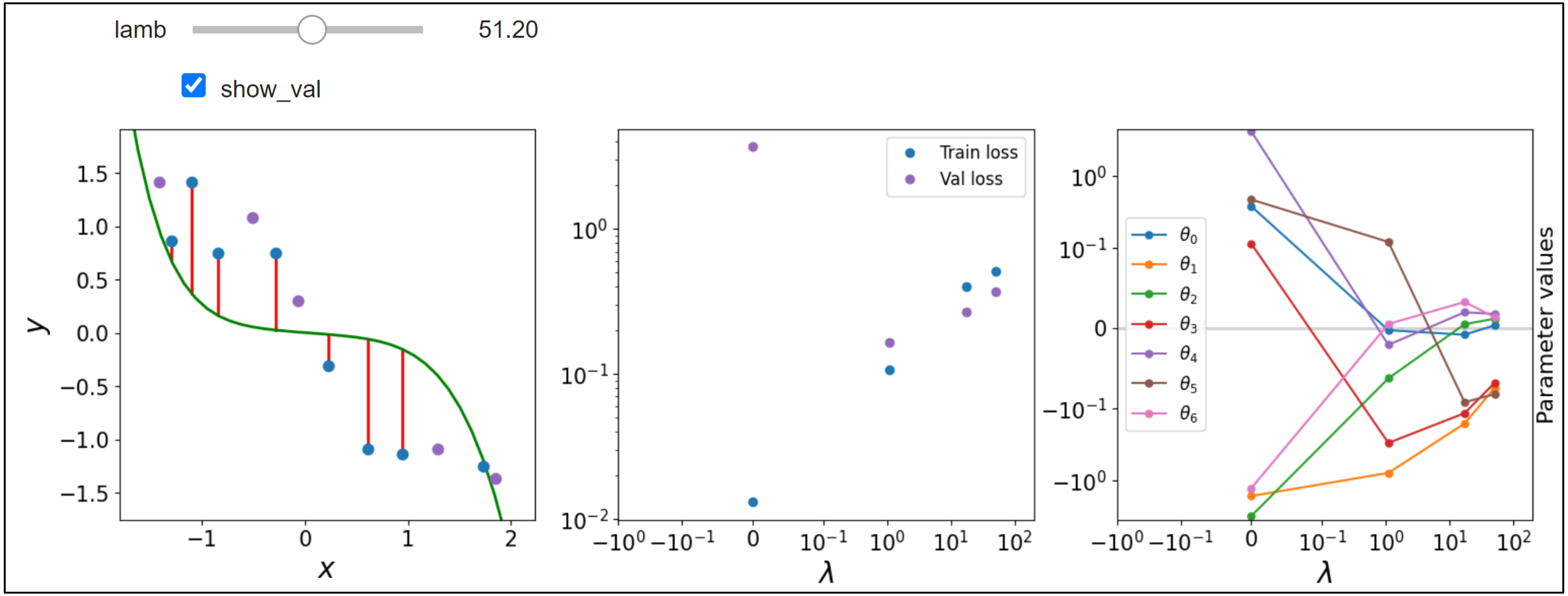
Overfitting can occur in all the models we've seen so far:

- 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

Best of both worlds

How can we keep the expressive power of a complex model while still avoiding overfitting?

Notebook demo: [regression_regularization.ipynb]()

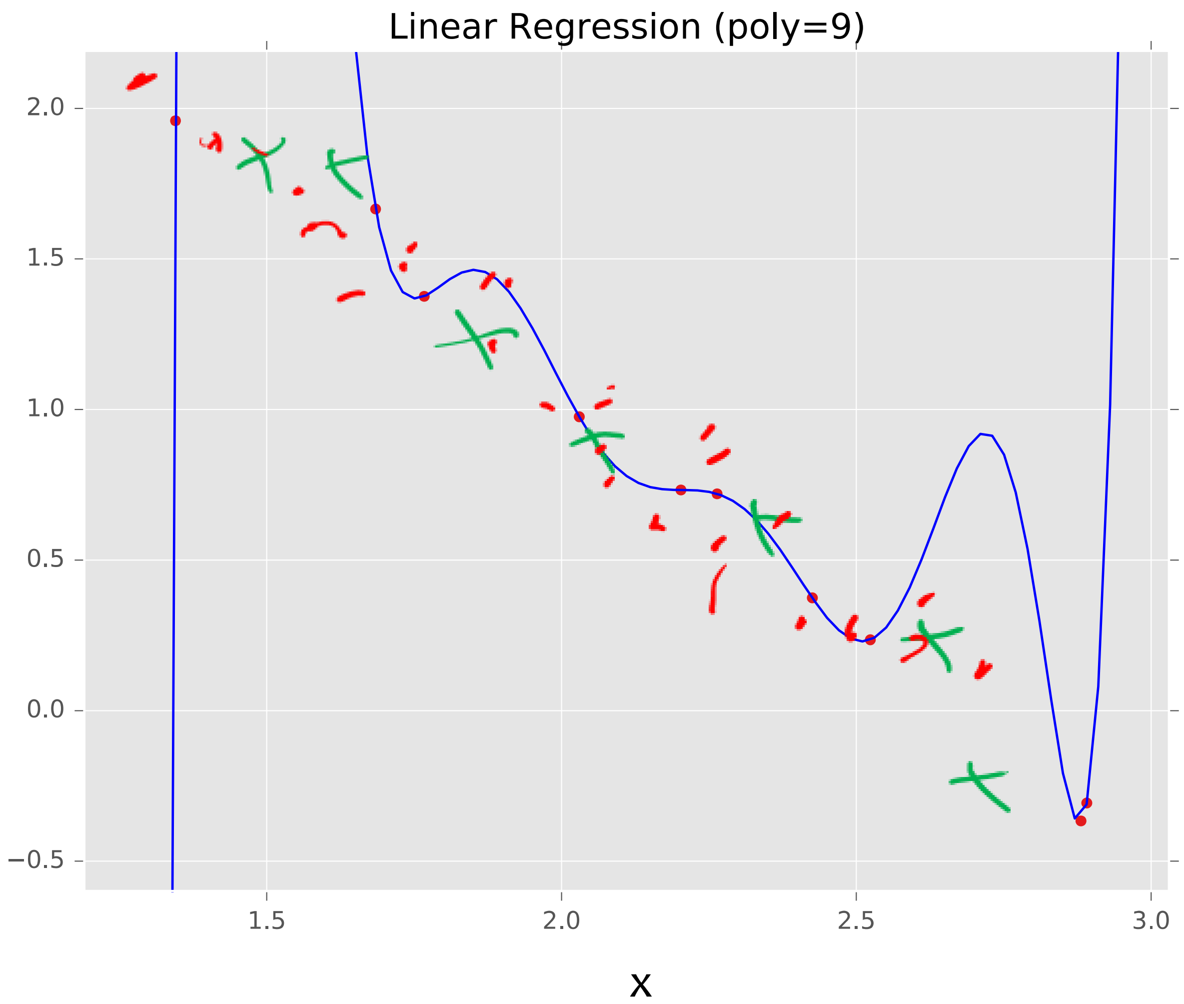


Example: Linear Regression

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

y	x	x^2	…	x^9
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

d

			$M=0$	$M=1$	$M=3$	$M=9$
	b	θ_0	0.19	0.82	0.31	0.35
x_1	w_1	θ_1		-1.27	7.99	232.37
x_1^2	w_2	θ_2			-25.43	-5321.83
x_1^3	w_3	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

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 (feature reduction)
3. small values for associated parameters

Regularization

$$\underline{J(\theta)} + \underline{r(\theta)}$$

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

$$r(\theta) = \|\theta\|_2^2 = \sum_{j=1}^{M} \theta_j^2$$

L1 norm: sum of absolute values

L0 norm: count of non-zero values

Regularization

$r(\theta) = \|\boldsymbol{\theta}\|_2^2$

A. $\boldsymbol{\theta}_A = [6,\ 3, -4, -2]^T$ $\quad 36 + 9 + 16 + 4 = 65$

B. $\boldsymbol{\theta}_B = [0,\ 3, -4,\ 0]^T$ $\quad 0 + 9 + 16 + 0 = 25$

Poll 2

Which model do you prefer?

A. $\boldsymbol{\theta}_A = [-190.0,\ -135.0,\ 310.0,\ 45.0]^T$ Training error: 0.0 $= J(\theta) = 0$

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

$r(\theta_B) = 0$

$h(\vec{x}) = 0 \cdot 1 + 0 \cdot x_1 + 0\, x_2 + 0\, x_3$

$= 0$

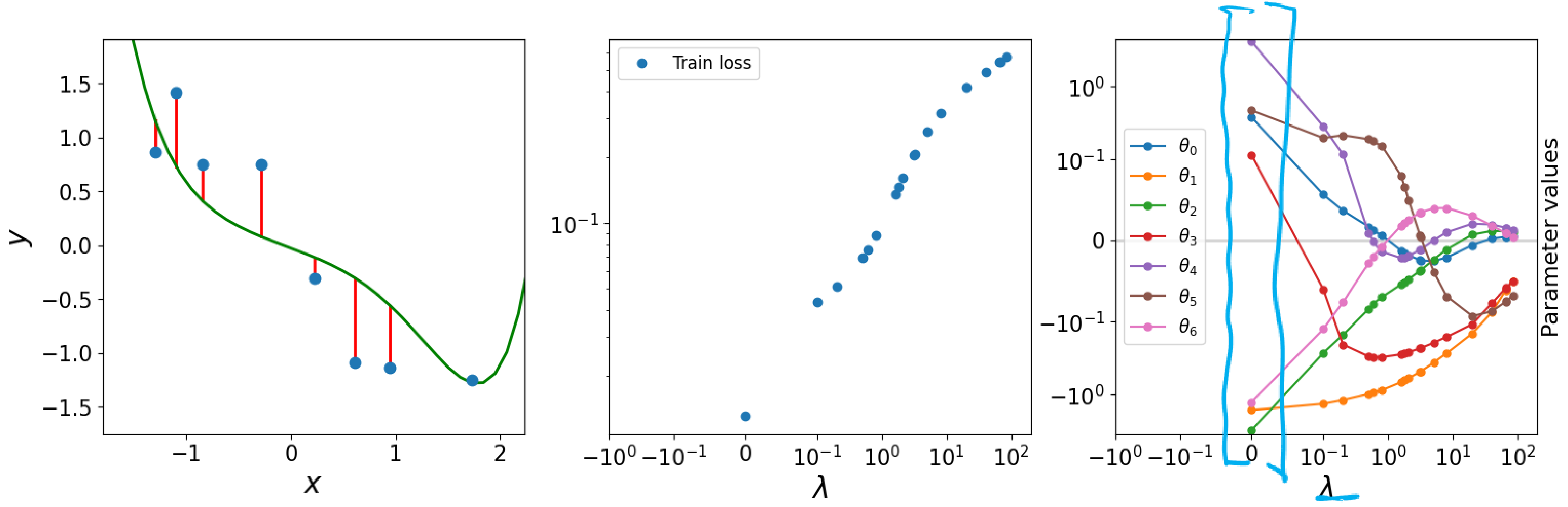
Poll 3

Notebook demo: regression_regularization.ipynb on course website

What is the best value for lambda?

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

$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3$

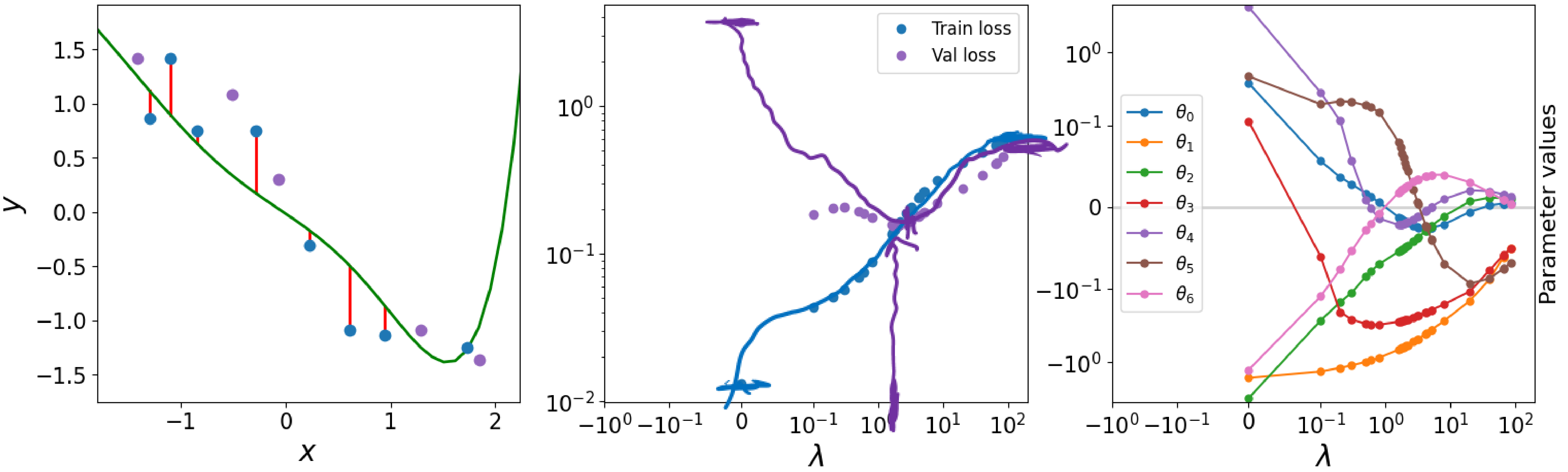


Poll 3

Notebook demo: regression_regularization.ipynb on course website

What is the best value for lambda?

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



Regularization

Given objective function: $J(\theta)$

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

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

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

Slide credit: CMU MLD Matt Gormley

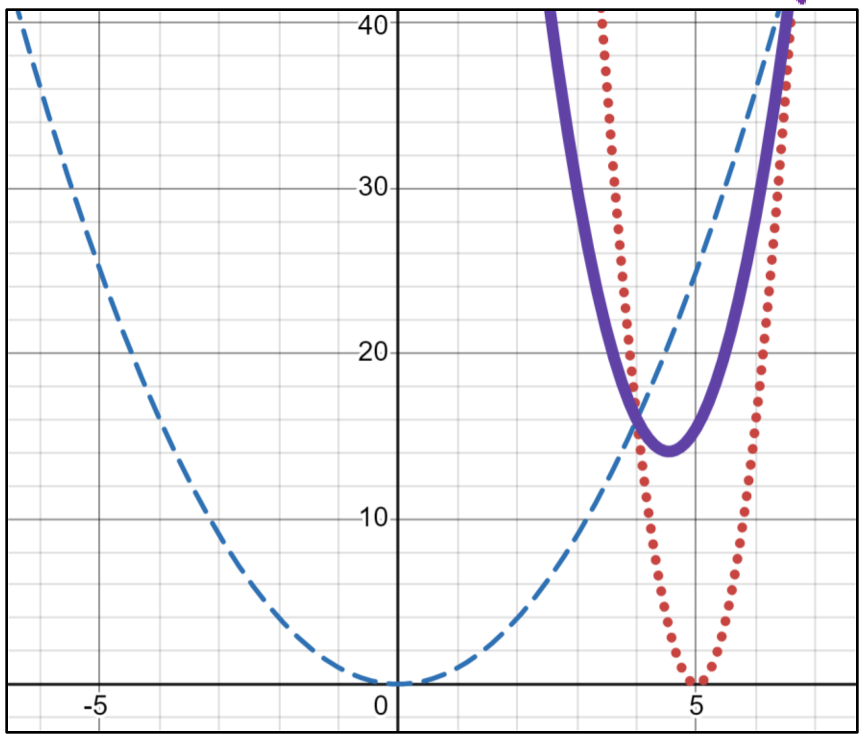
Regularization

L2 Demos

$J'(\theta) =$

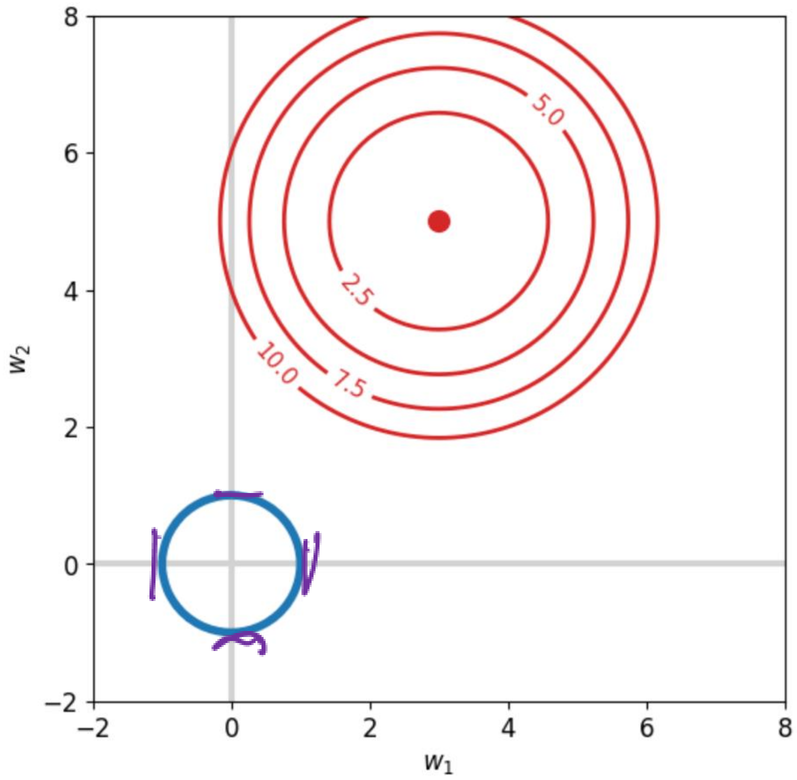
Desmos: 1-D

Regularization Interpolation

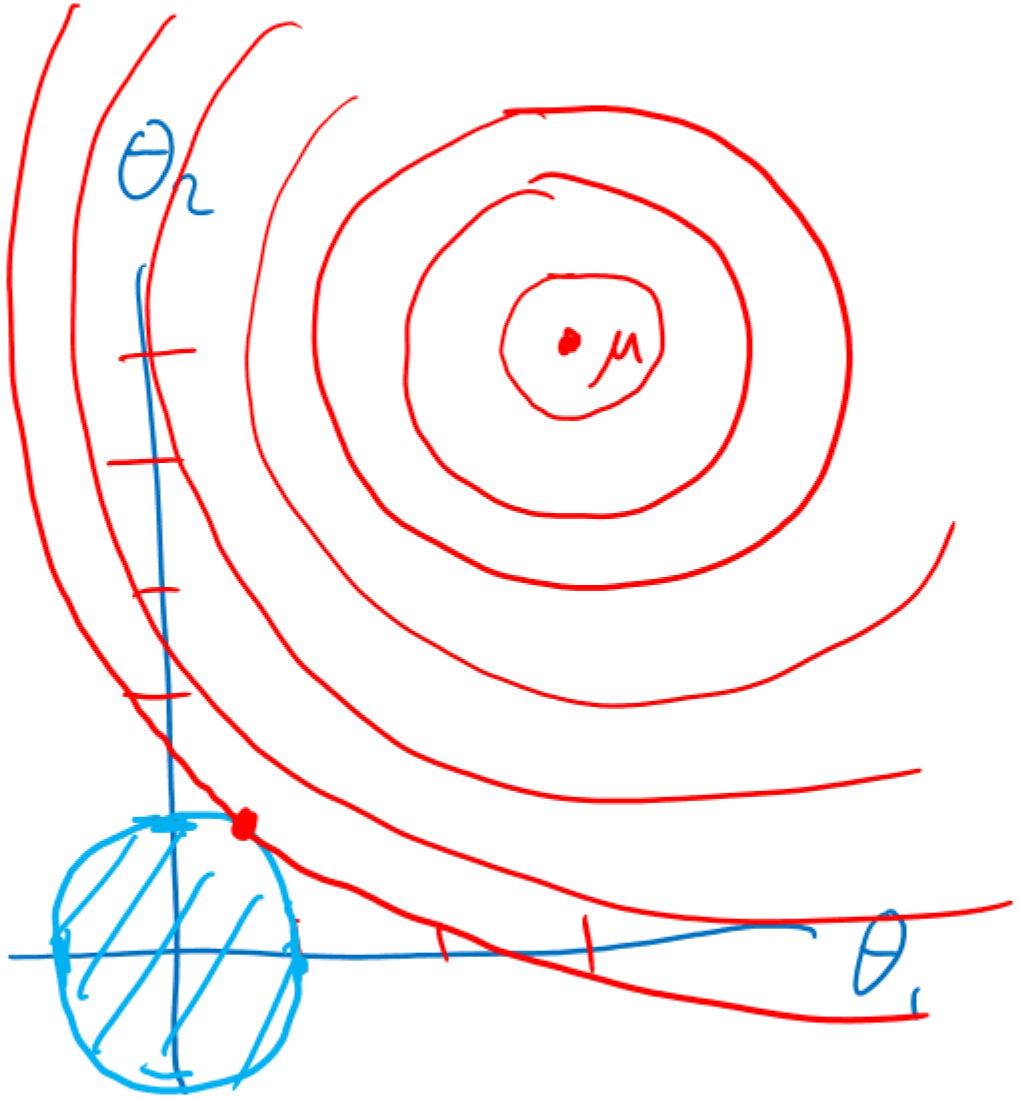


Notebook: 2-D

L1_sparsity.ipynb (L2 part for now)



Regularization



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

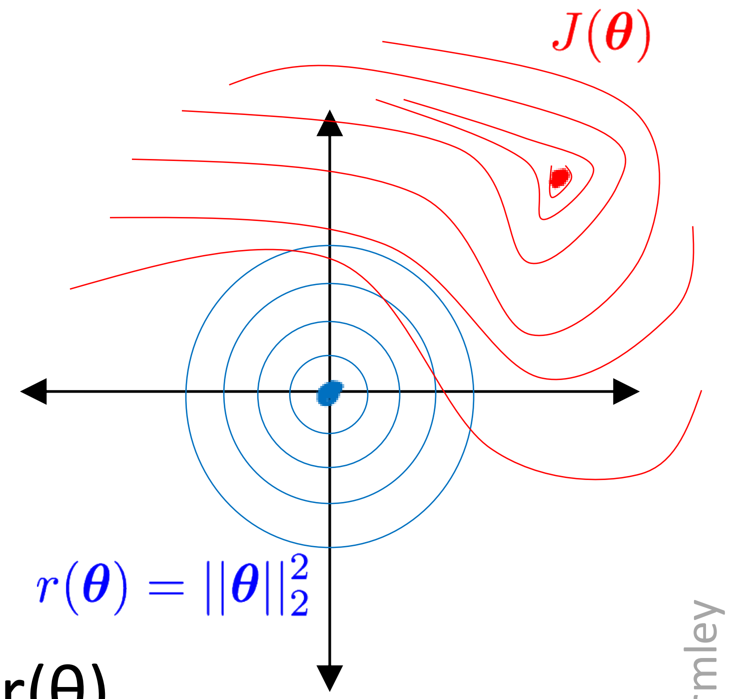
$$\min_{\theta} \; J(\theta_1, \theta_2)$$

$$\text{s.t.} \quad \|\theta\|_2 \leq 1$$

Poll 4

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

Regularization Exercise

In-class Exercise

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

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

Slide credit: CMU MLD Matt Gormley

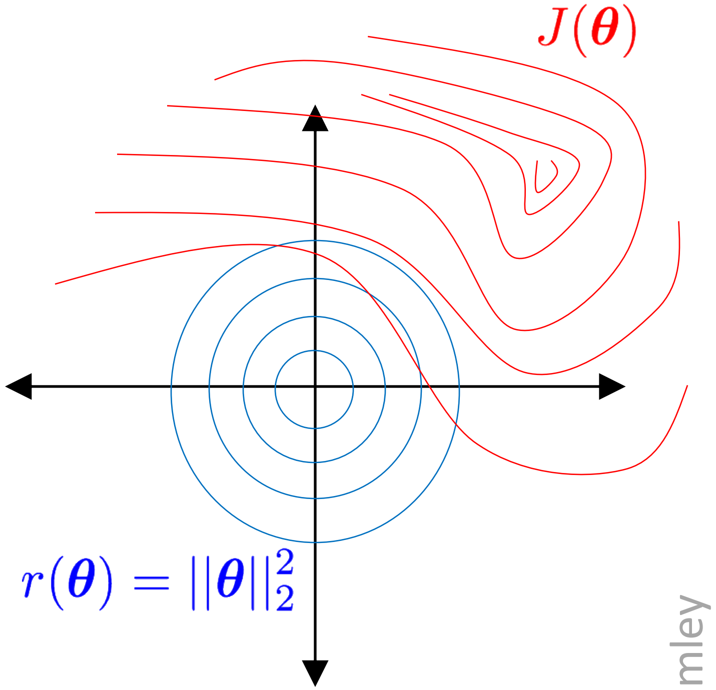
Poll 5

Suppose we are minimizing $J'(\theta)$ 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

Poll 6

As we increase λ, our model is more likely to:

A. Overfit

B. Underfit

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

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 Optimization

Linear Regression with L2 Regularization

a.k.a Ridge regression or Tychonov regression

$$\hat{y} = X\theta$$

$$J(\theta) = \ell(y, \hat{y}) + \lambda\|\theta\|_2^2$$

$$\ell(\hat{y}) = \|y - \hat{y}\|_2^2$$

$$J(\theta) = \|\vec{y} - X\vec{\theta}\|_2^2 + \lambda\|\theta\|_2^2$$

denom

$M \times 1$

$$\frac{\partial J}{\partial \theta} = 0 = \frac{\partial \hat{y}}{\partial \vec{\theta}}\left[\frac{\partial \ell}{\partial \hat{y}}\right] + \frac{\partial (\;)}{\partial \theta}$$

$$\begin{bmatrix} \theta \\ \theta \\ \theta \\ \theta \end{bmatrix} = X^T\left[-2(\vec{y} - X\vec{\theta})\right] + 2\lambda\vec{\theta}$$

$$0 = -\cancel{2}X^T y + \cancel{2}X^T X\theta + \cancel{2}\lambda\theta$$

$$X^T y = X^T X\theta + \lambda\theta = \cancel{(X^T X + \lambda)\theta} = (X^T X + \lambda I_M)\theta$$

$$(X^T X + \lambda I_M)^{-1} X^T y = \theta$$

Linear Algebra Timeout

$Av + cv$

Distribution of multiplication and addition with scalar involved

Original

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 5 \end{bmatrix} + 3\begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

$$Av + cv$$

Broken

$$\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + 3\right)\begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

$$(A + c)v$$

Fixed

$$\left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + 3\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)\begin{bmatrix} 2 \\ 5 \end{bmatrix}$$

$$(A + cI)v$$

Regularization with L1 norm

Model Preference

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

Model structure (for both models):

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

Model parameters:

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

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

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

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

Motivation: Regularization

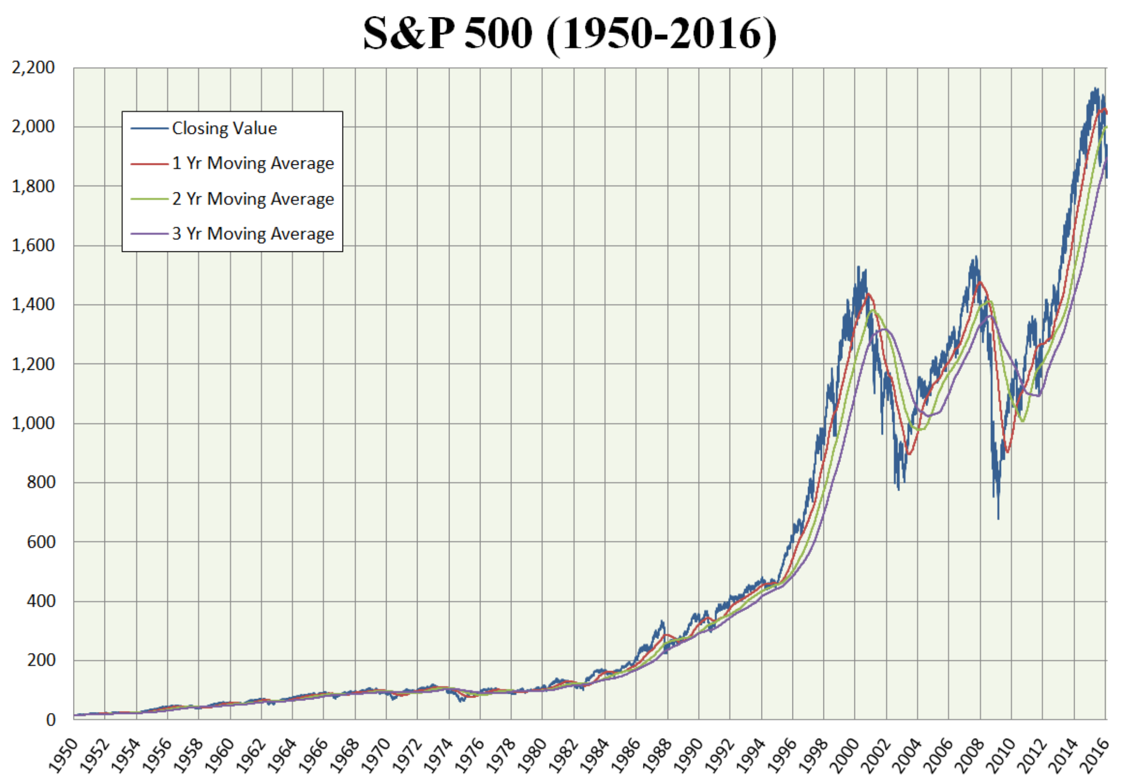
Example: Stock Prices

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?



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 $\|\theta\|_2^2 = \sum \theta_j^2$

L1 norm: sum of absolute values $\|\theta\|_1 = \sum |\theta_j|$

L0 norm: count of non-zero values $\leftarrow \|\theta\|_0 = \sum \mathbb{1}(\theta_j \neq 0)$

Regularization

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

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

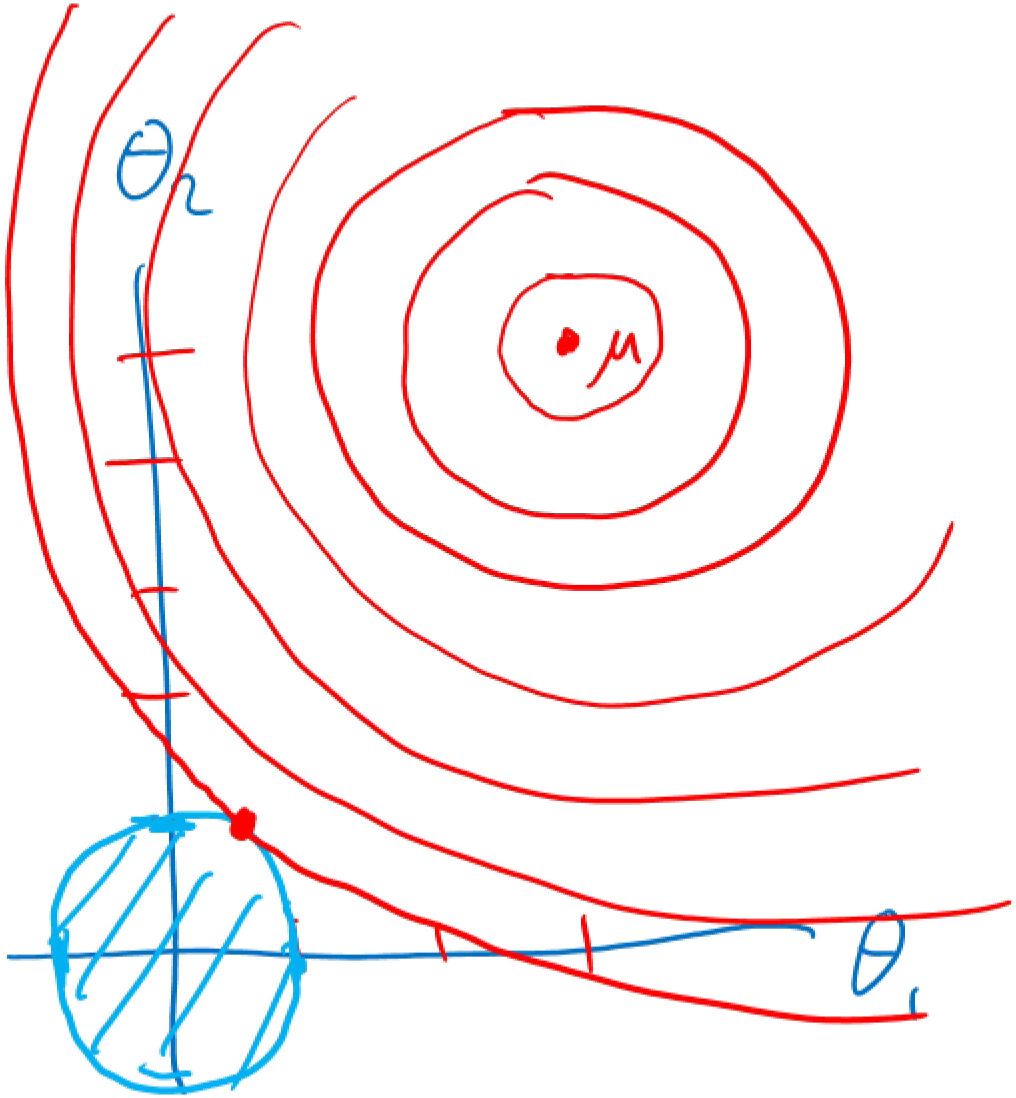
Choose form of $r(\boldsymbol{\theta})$:

- 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	$\lVert\boldsymbol{\theta}\rVert_0 = \sum \mathbb{1}(\theta_m \neq 0)$	zero values	L0 reg.	no good computational solutions
1	$\lVert\boldsymbol{\theta}\rVert_1 = \sum \lvert\theta_m\rvert$	zero values	L1 reg.	subdifferentiable
2	$(\lVert\boldsymbol{\theta}\rVert_2)^2 = \sum \theta_m^2$	small values	L2 reg.	differentiable

Regularization



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

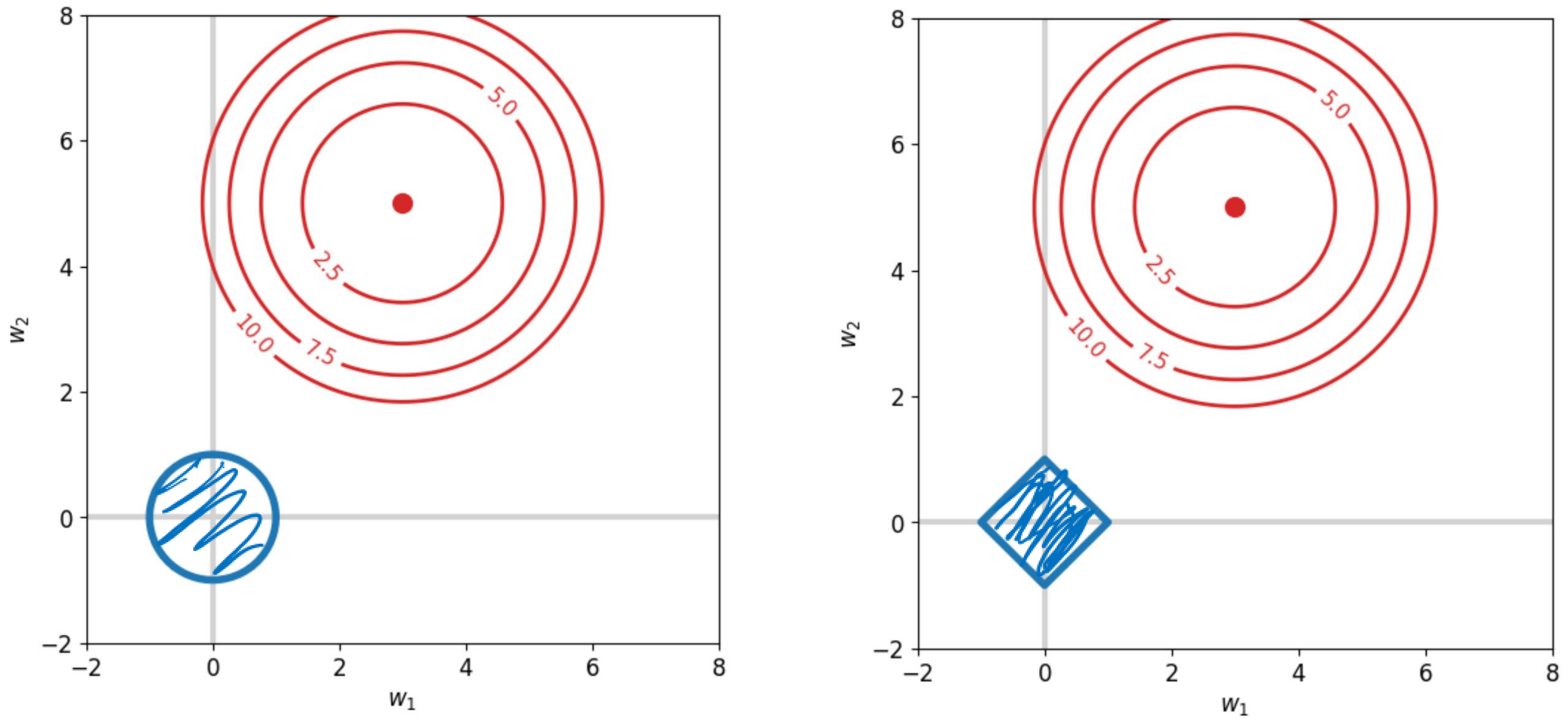
$$\min_{\theta} \; J(\theta_1, \theta_2)$$

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

Regularization

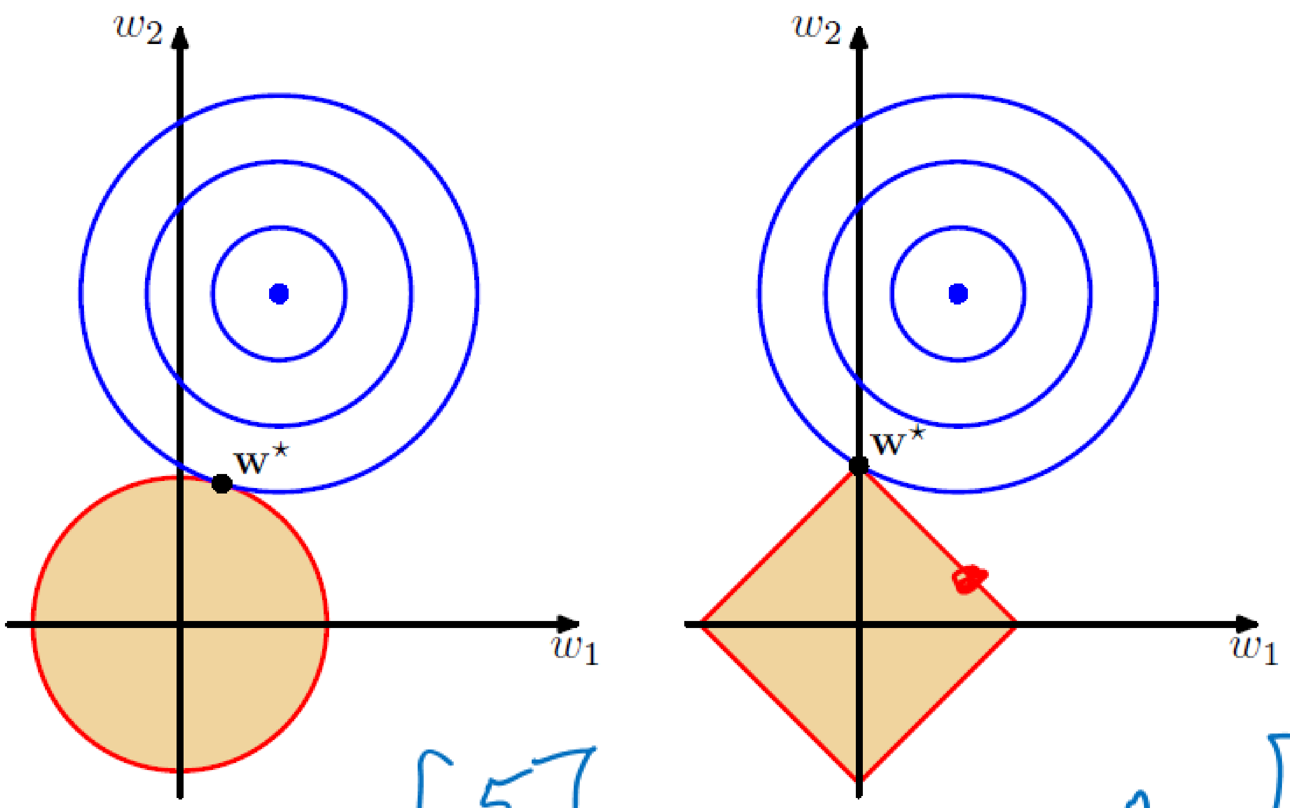
L1 demo: L1_sparsity.ipynb

$$\underset{\theta}{\arg\min}\ J(\theta)$$
$$\text{s.t. } \|\theta\|_2^2 \leq 1$$



L2 vs L1 Regularization

Combine original objective with penalty on parameters



$\|\theta\|_0 =$ count on nonzeros

$\theta = \begin{bmatrix} .5 \\ 1.0 \end{bmatrix}$

$\theta = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Figures: Bishop, Ch 3.1.4

L2 vs L1: Housing Price Example

Predict housing price from several features

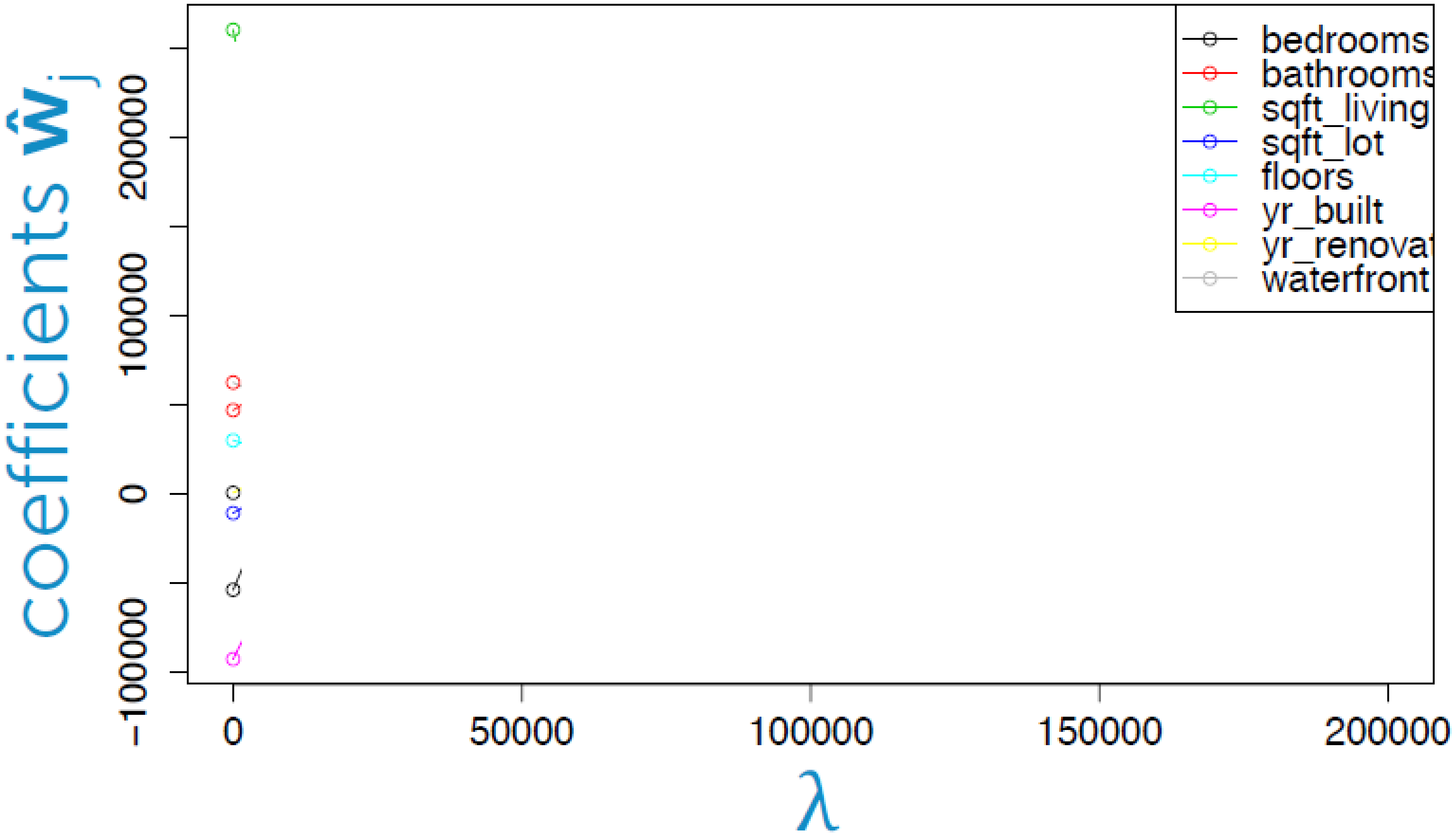


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

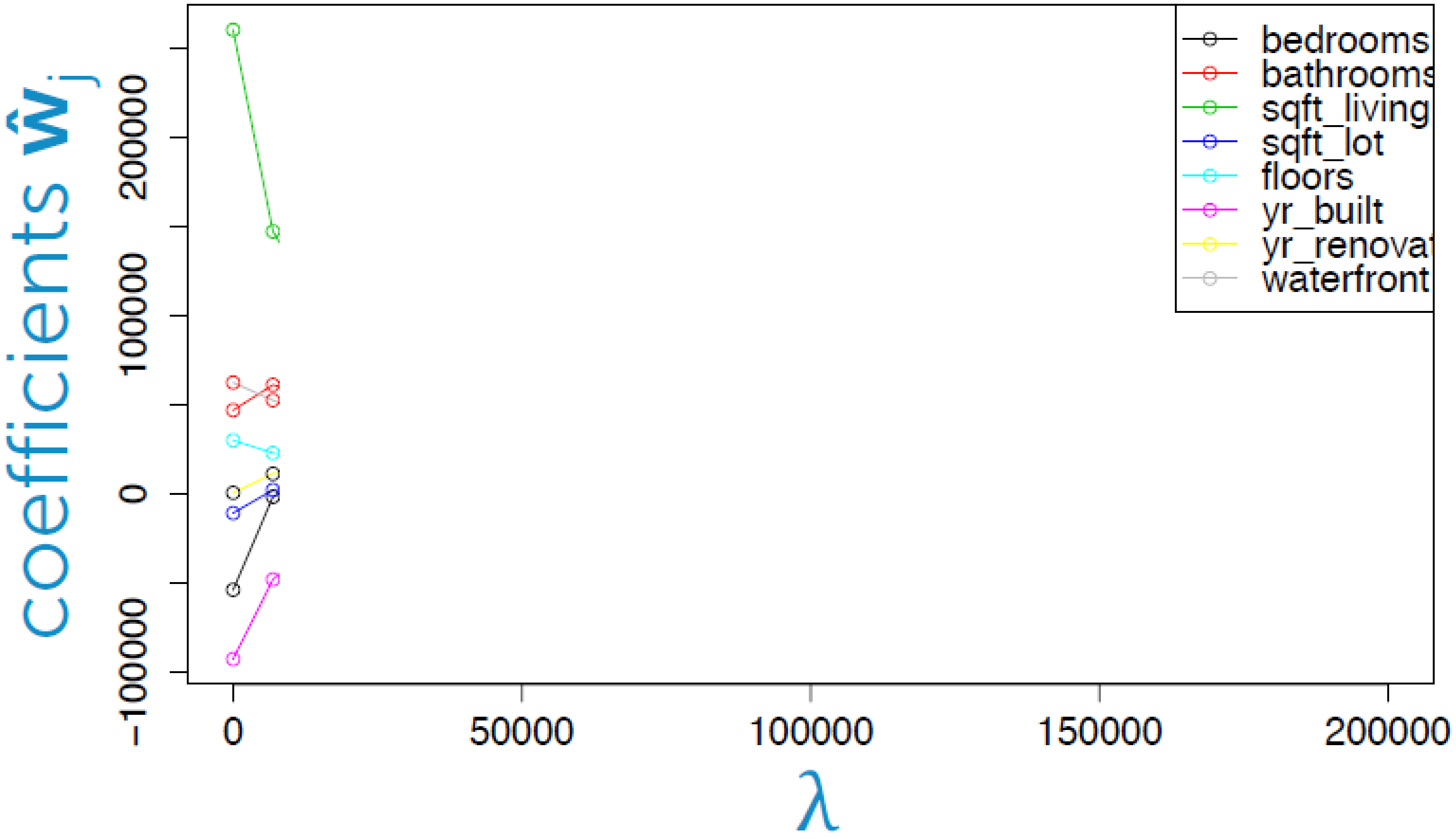


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

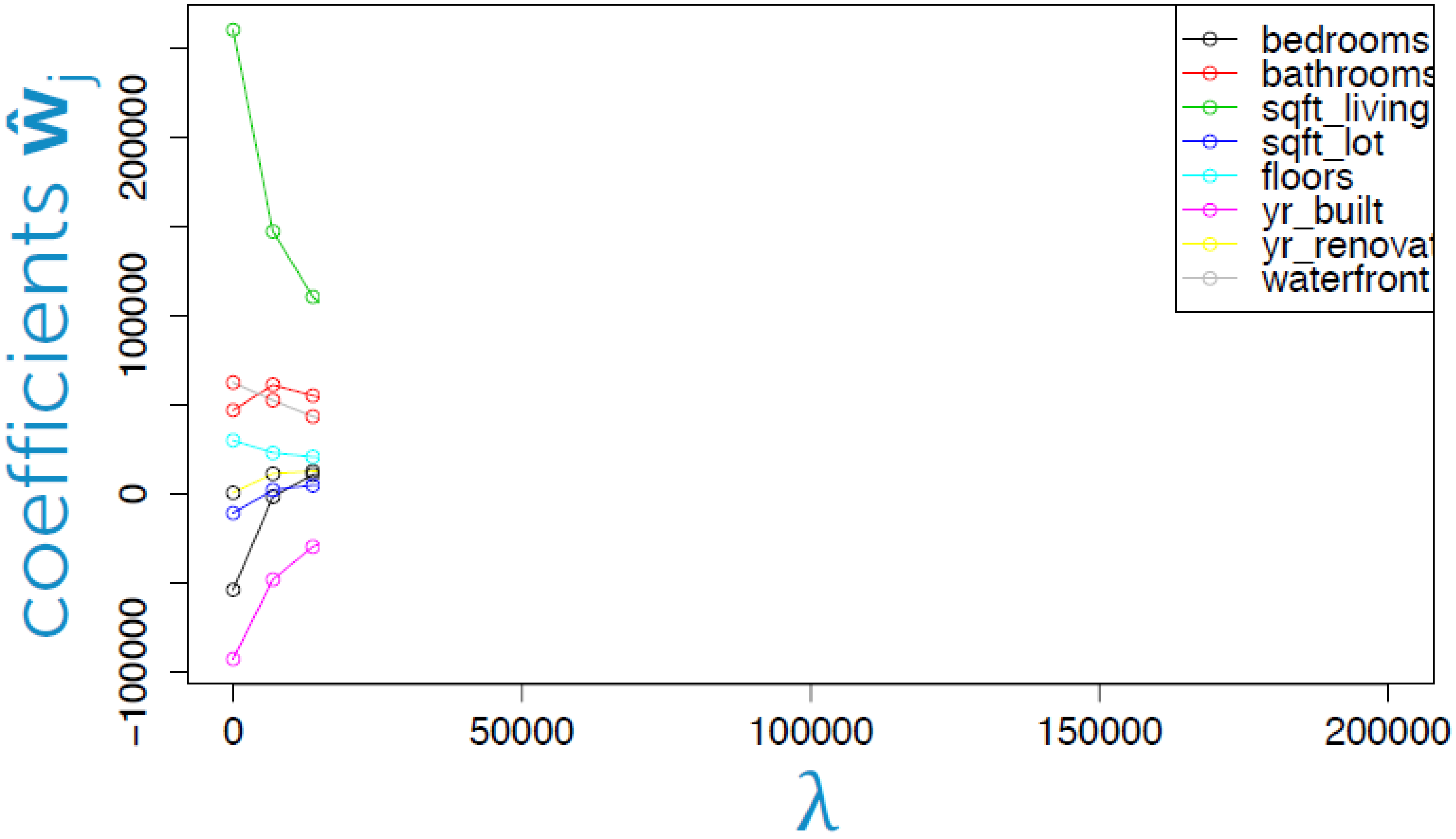


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

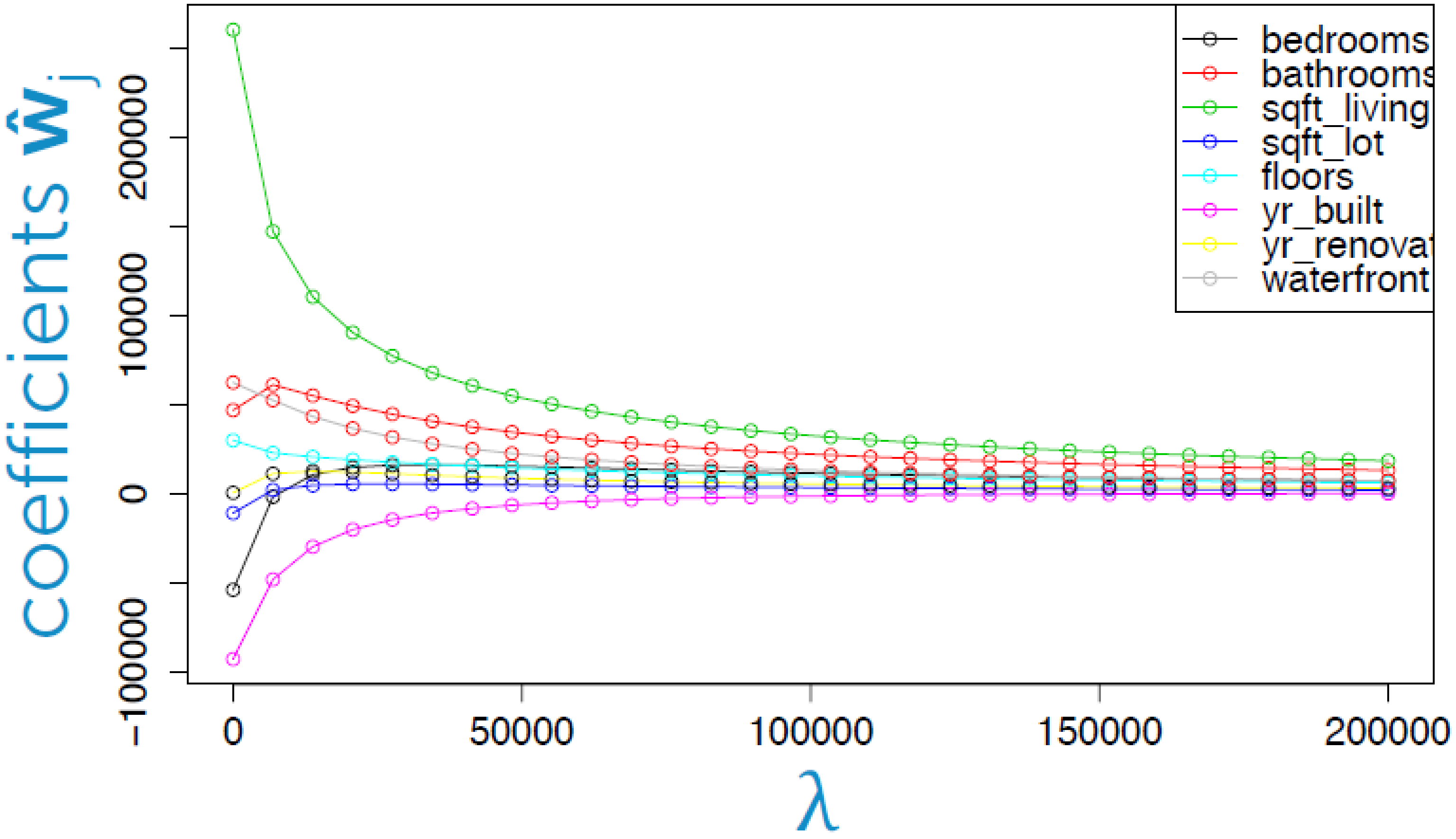


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

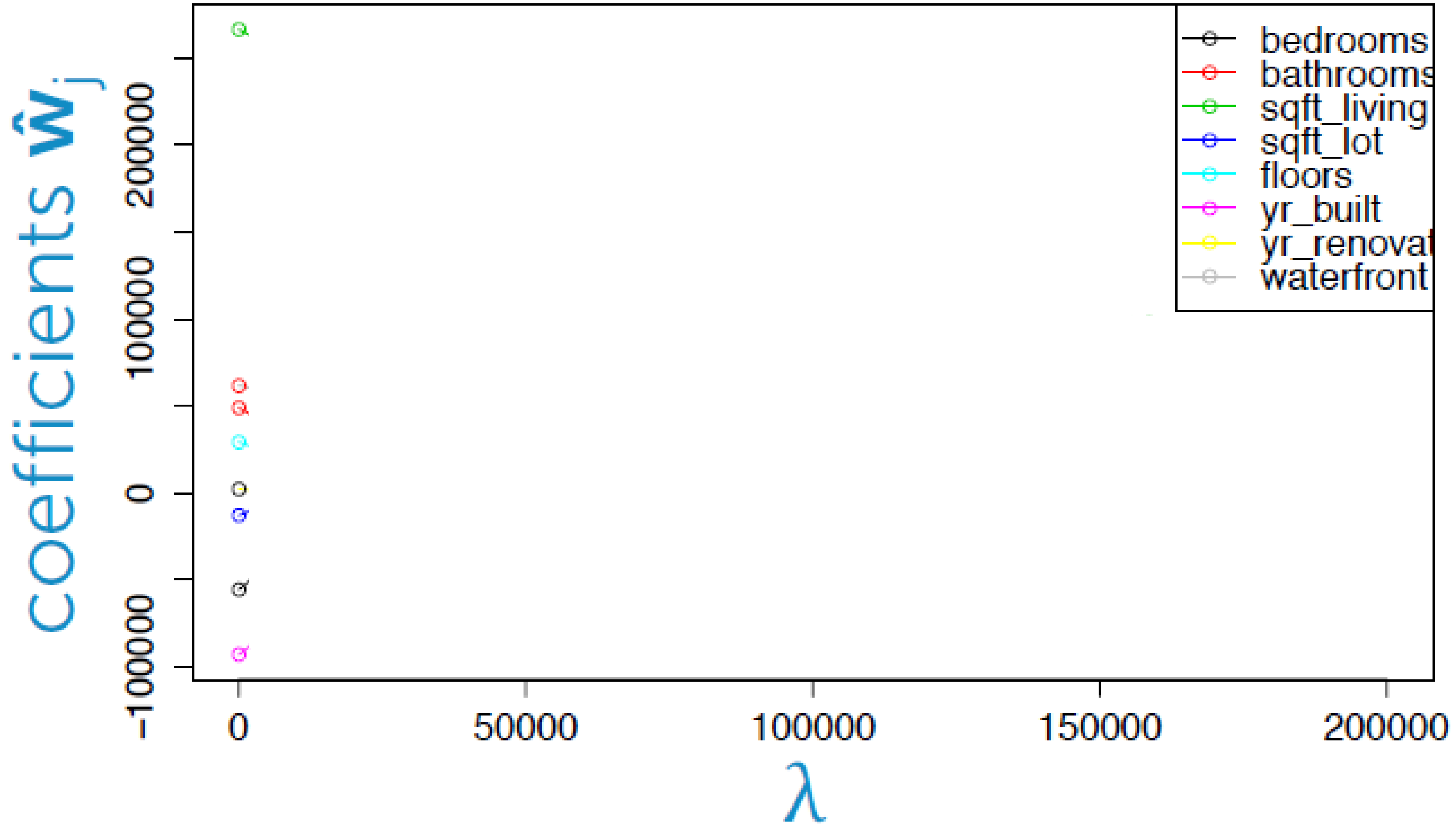


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

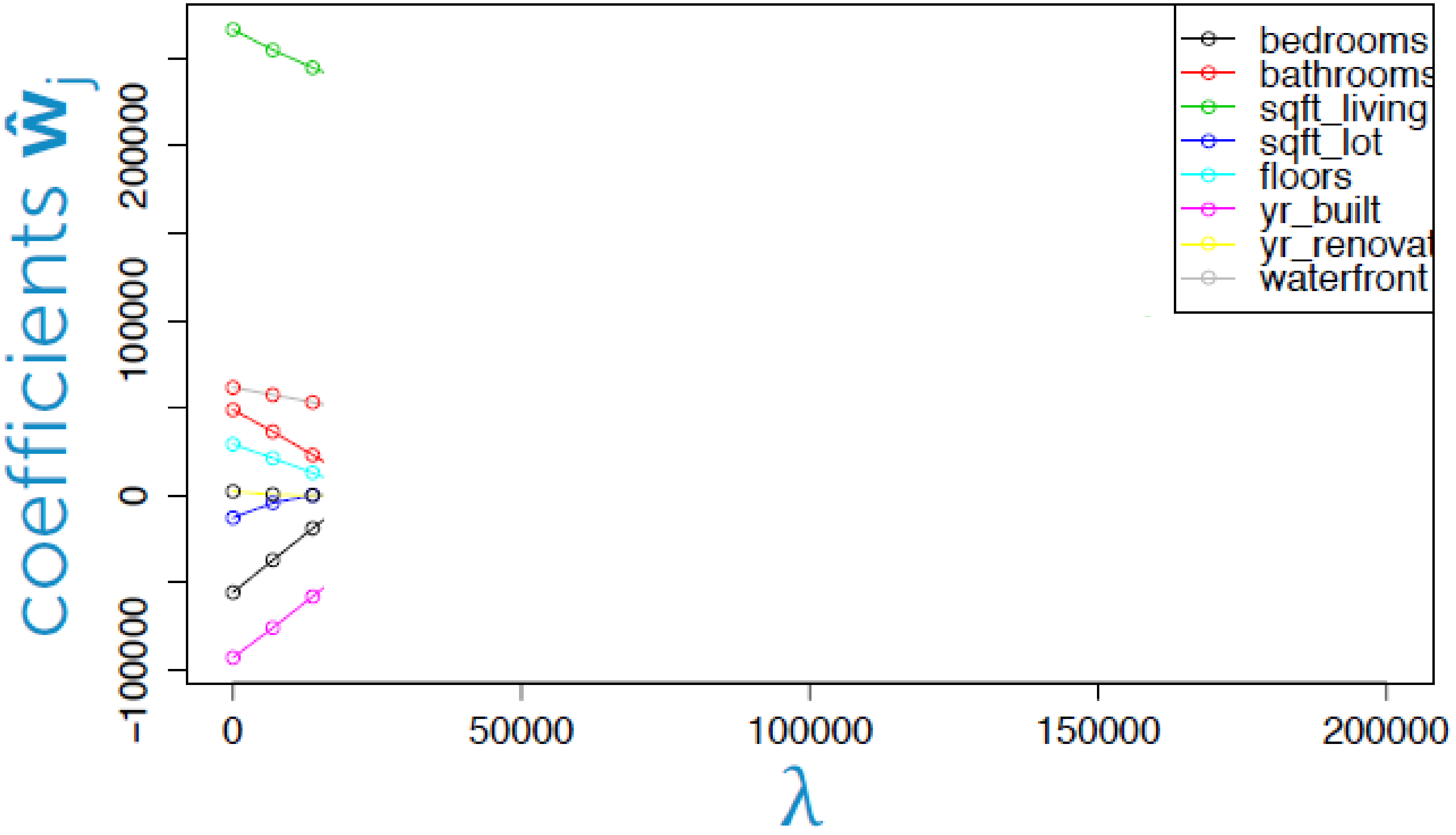


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

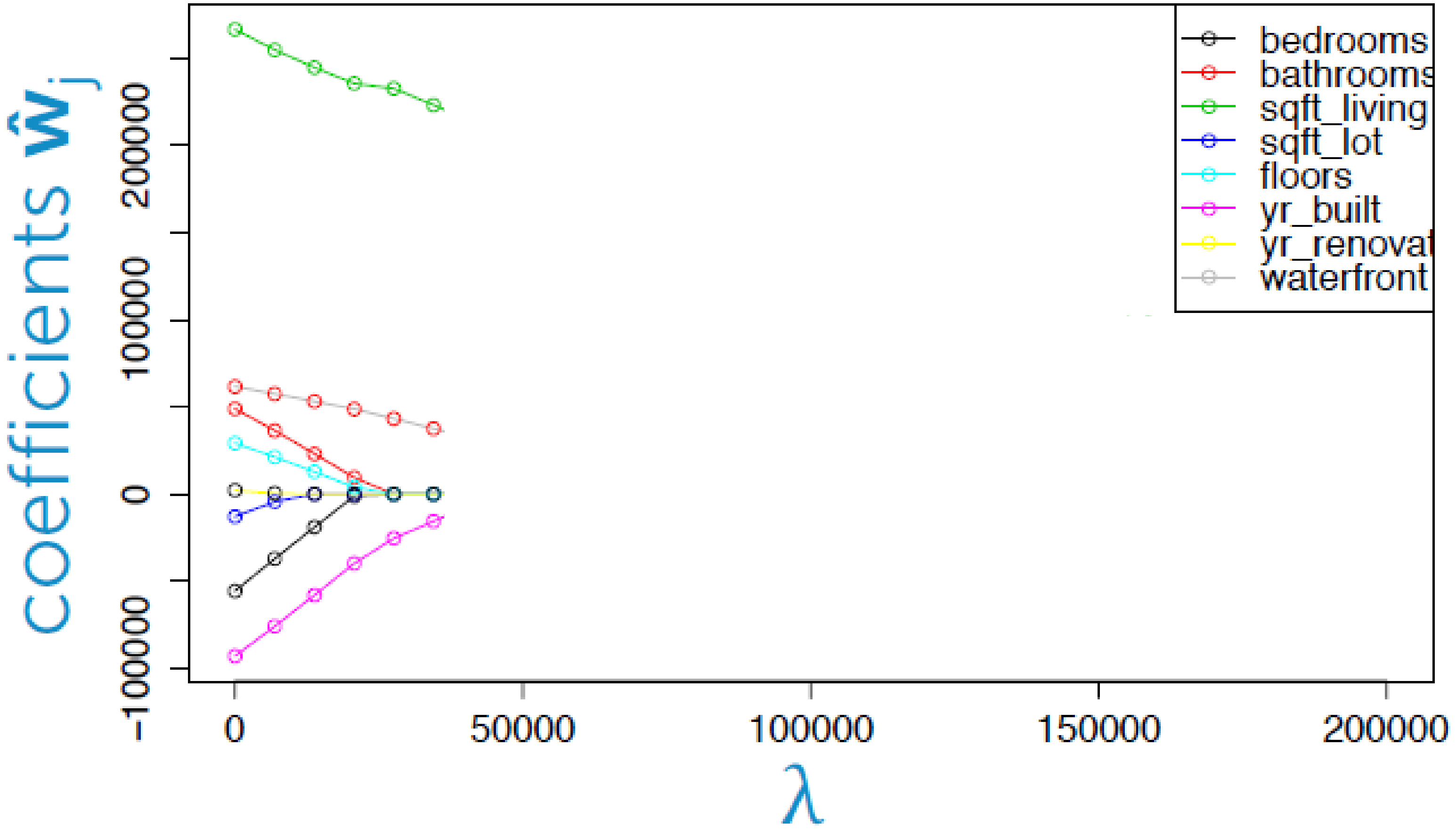


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

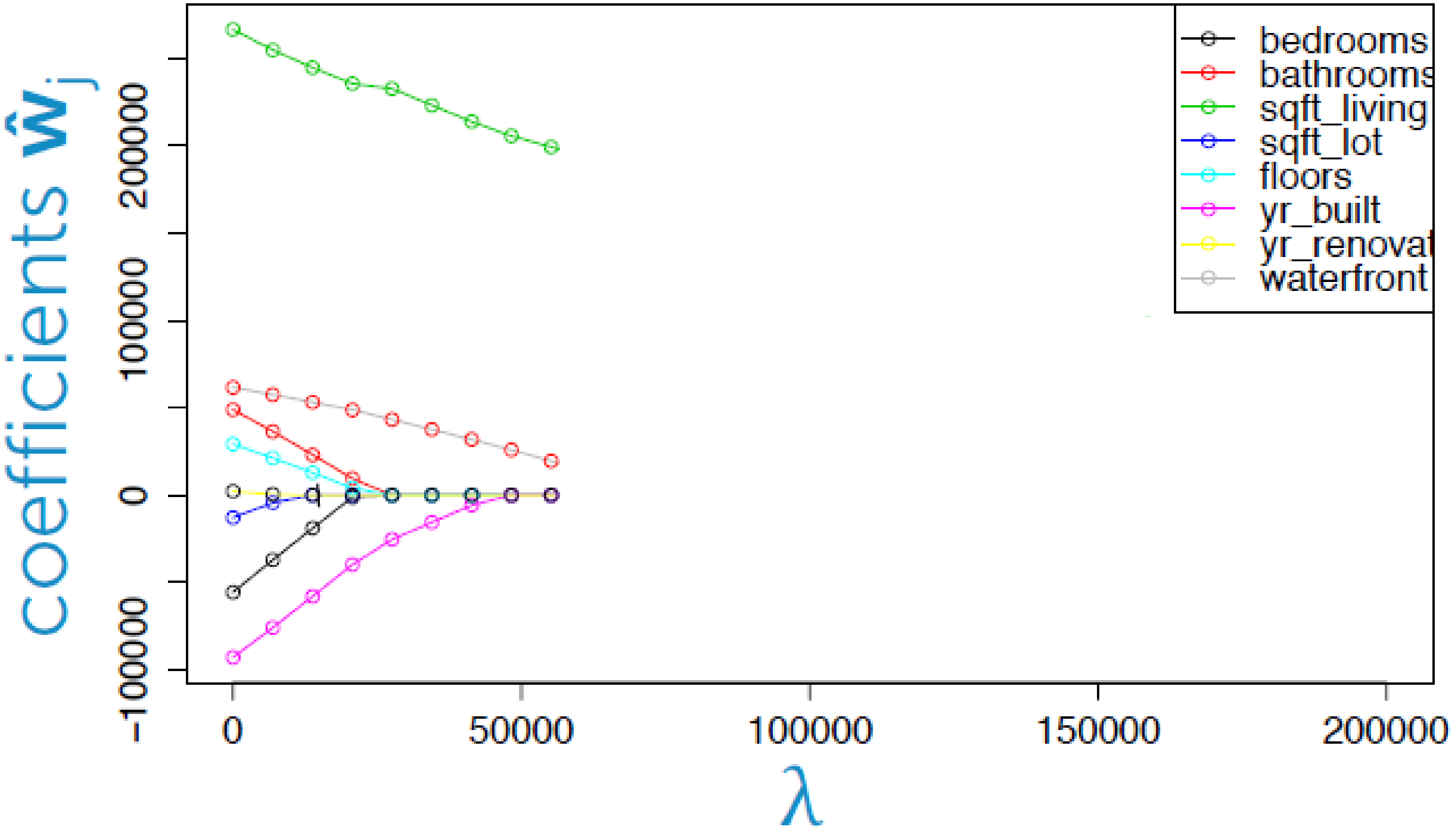


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

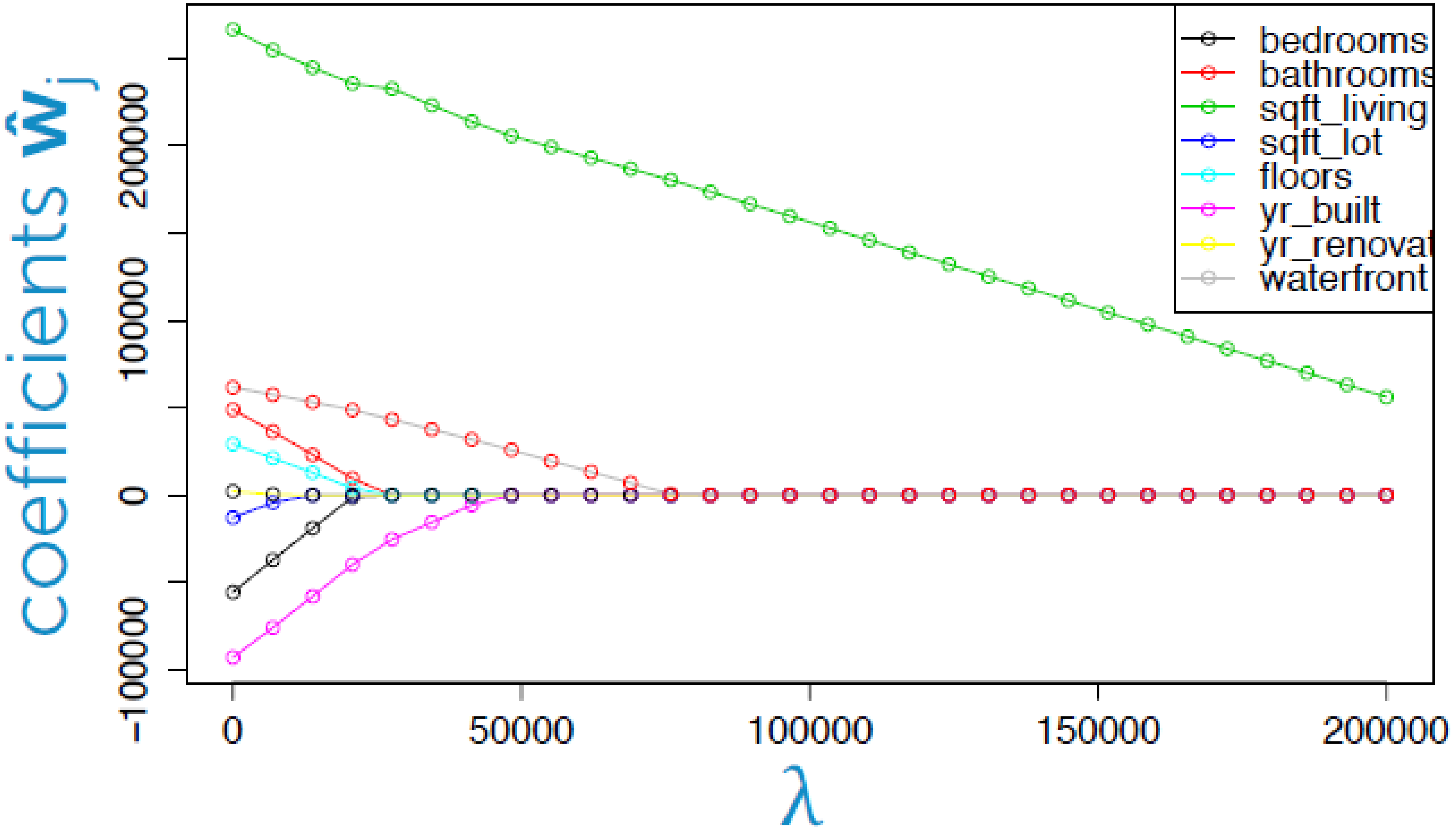


Figure: Emily Fox, University of Washington

Regularization as MAP

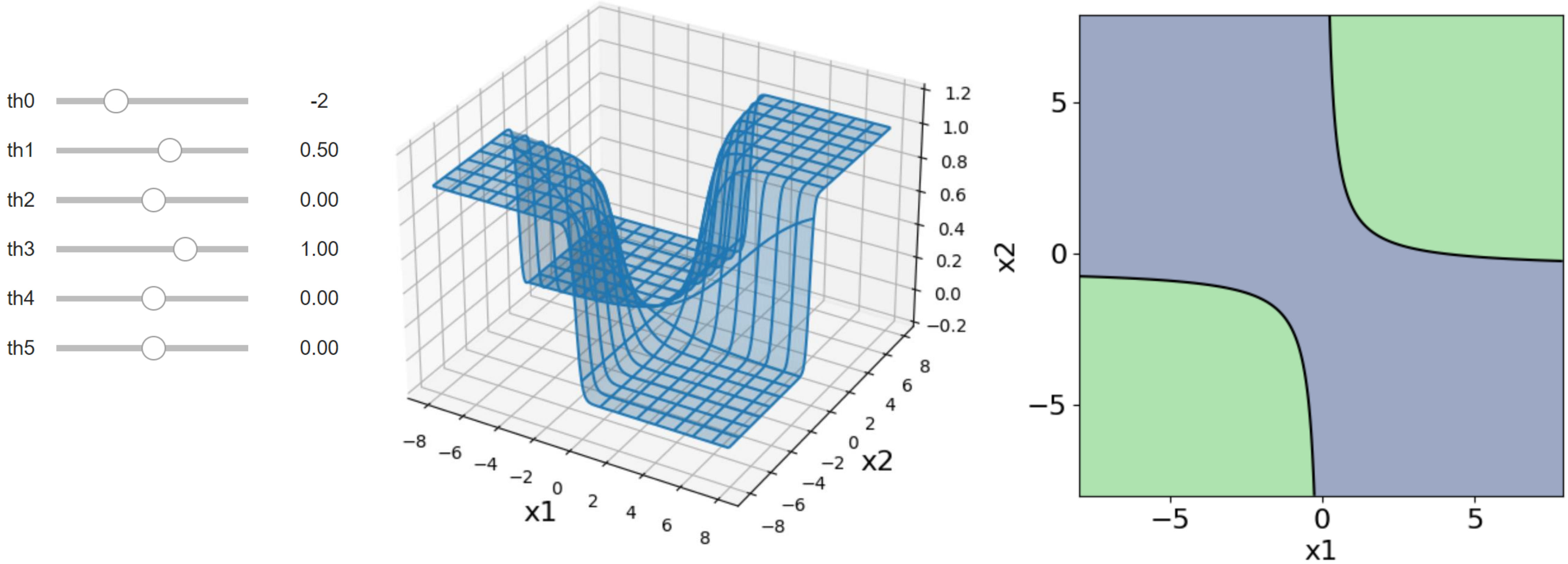
L1 and L2 regularization can be interpreted as **maximum a-posteriori (MAP) estimation** of the parameters

To be discussed later in the course…

Additional Slides

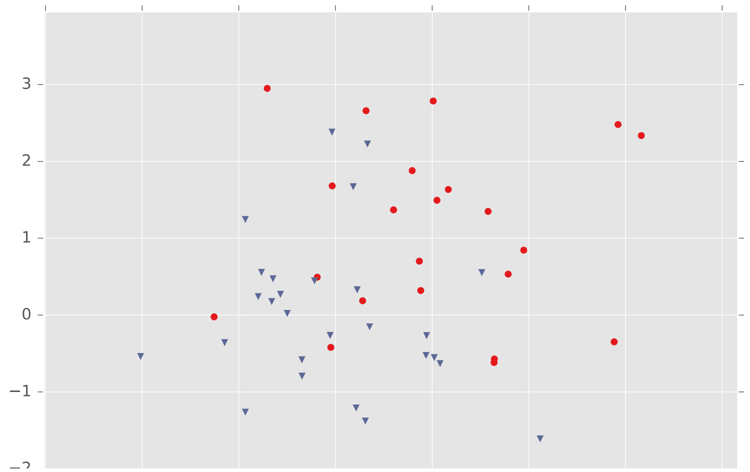
Logistic Regression with Nonlinear Features

Jupyter notebook demo: [quadratic_logistic.ipynb]

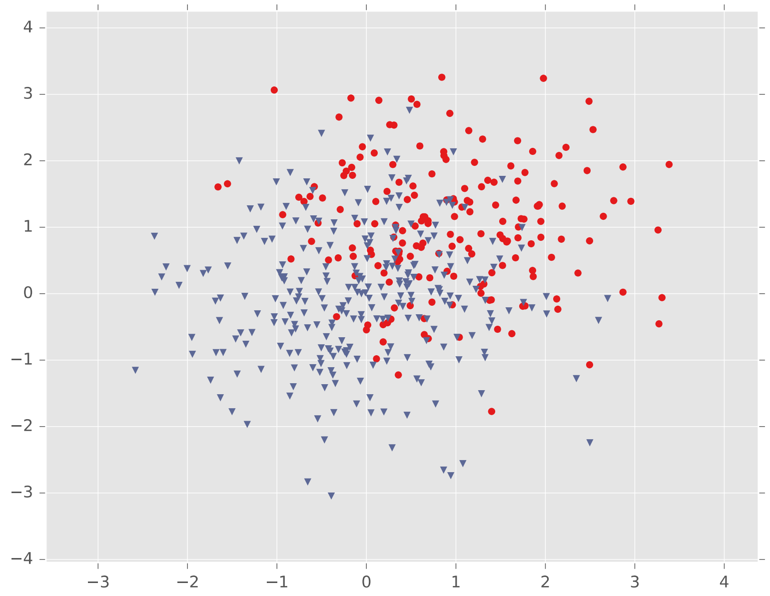


Example: Logistic Regression

Training Data



Test Data



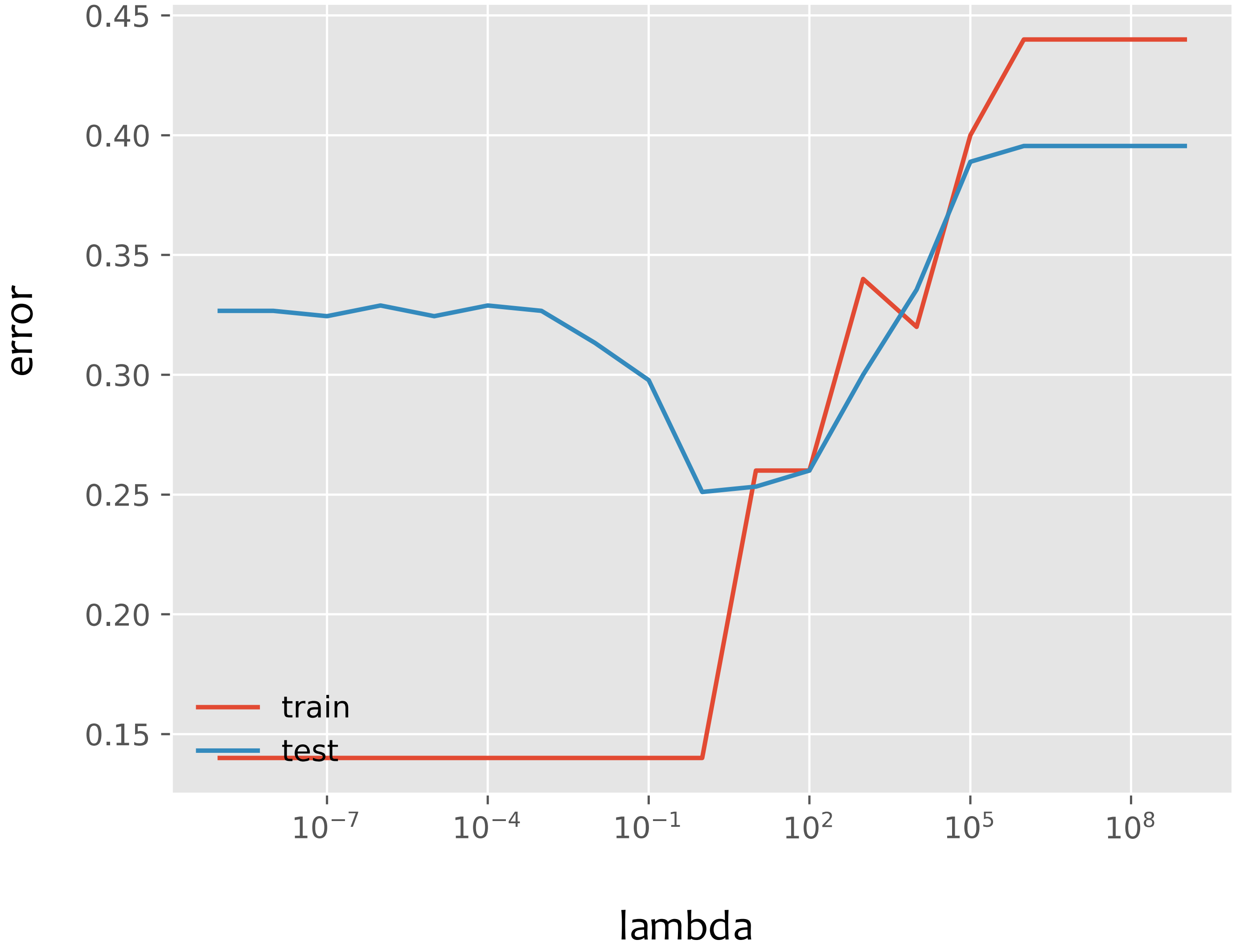
For this example, we construct **nonlinear features** (i.e. feature engineering)

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)

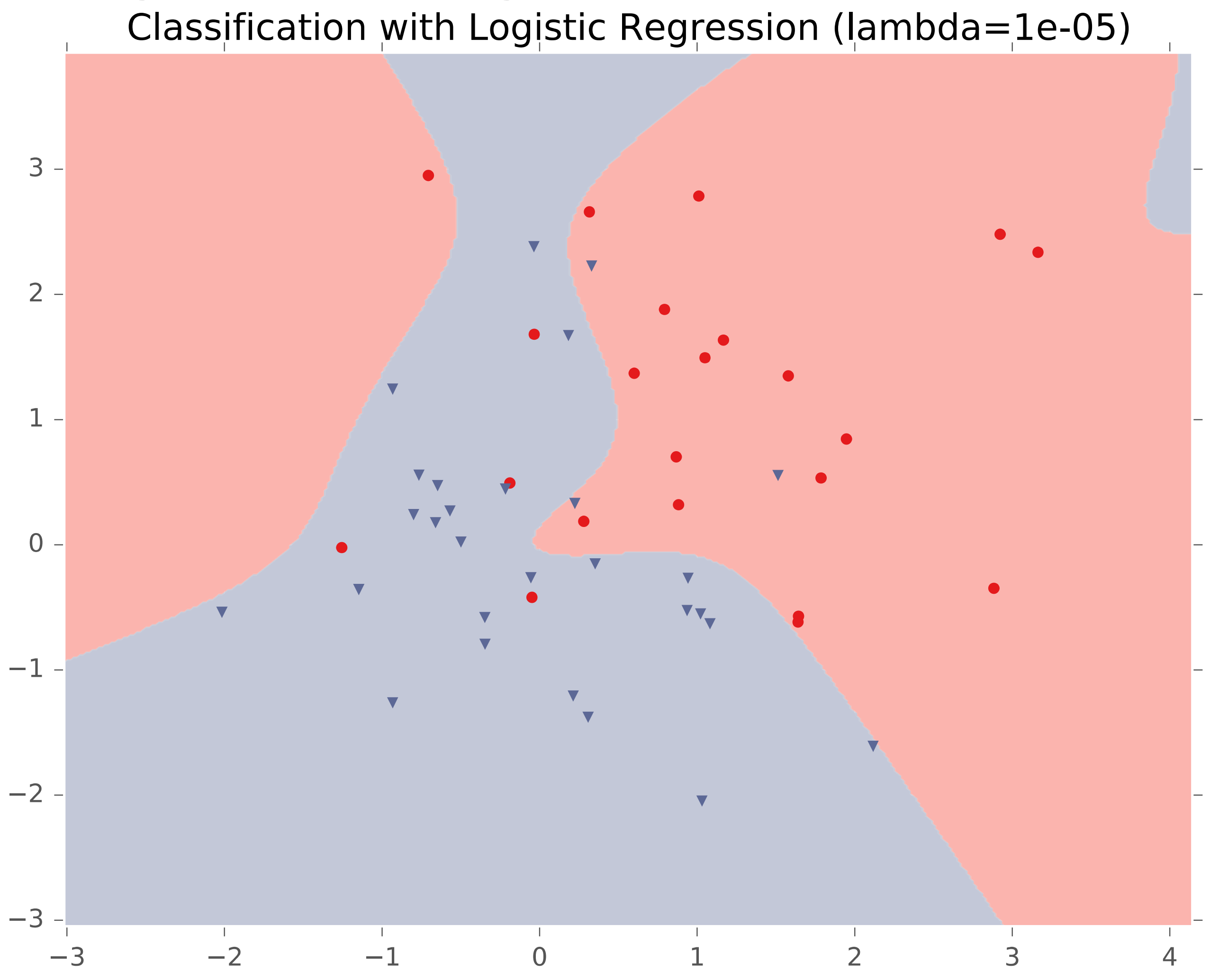
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



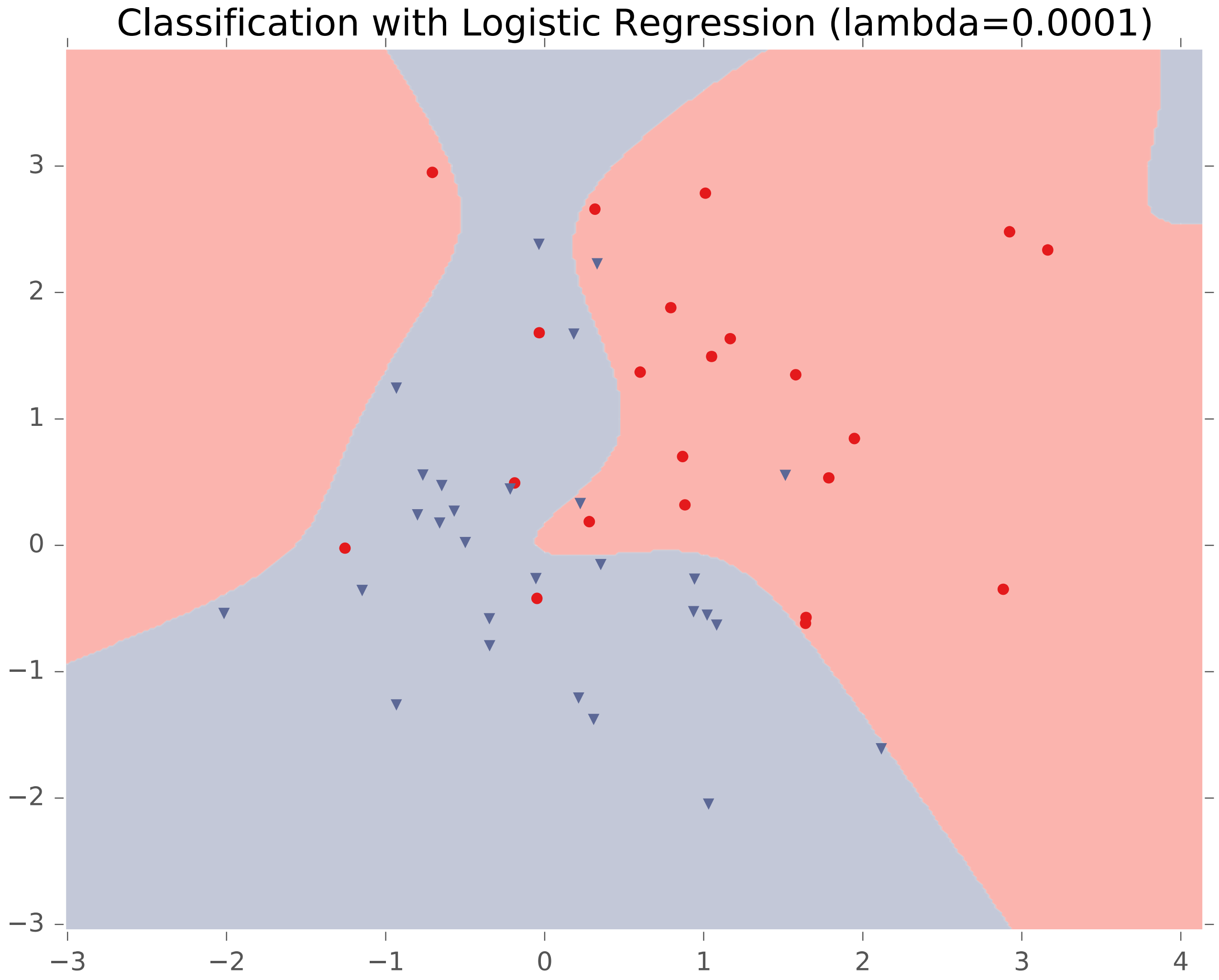
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

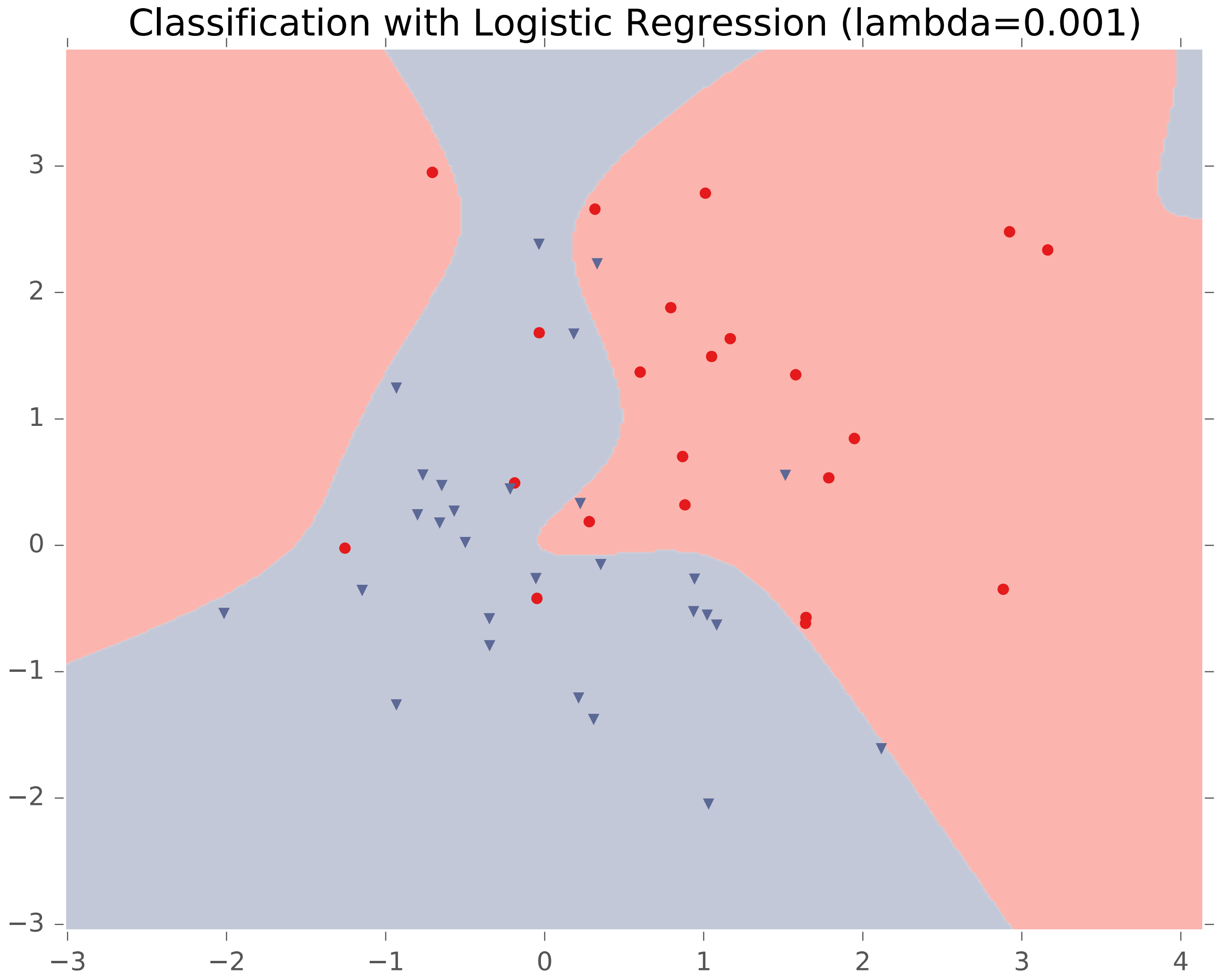


Slide credit: CMU MLD Matt Gormley

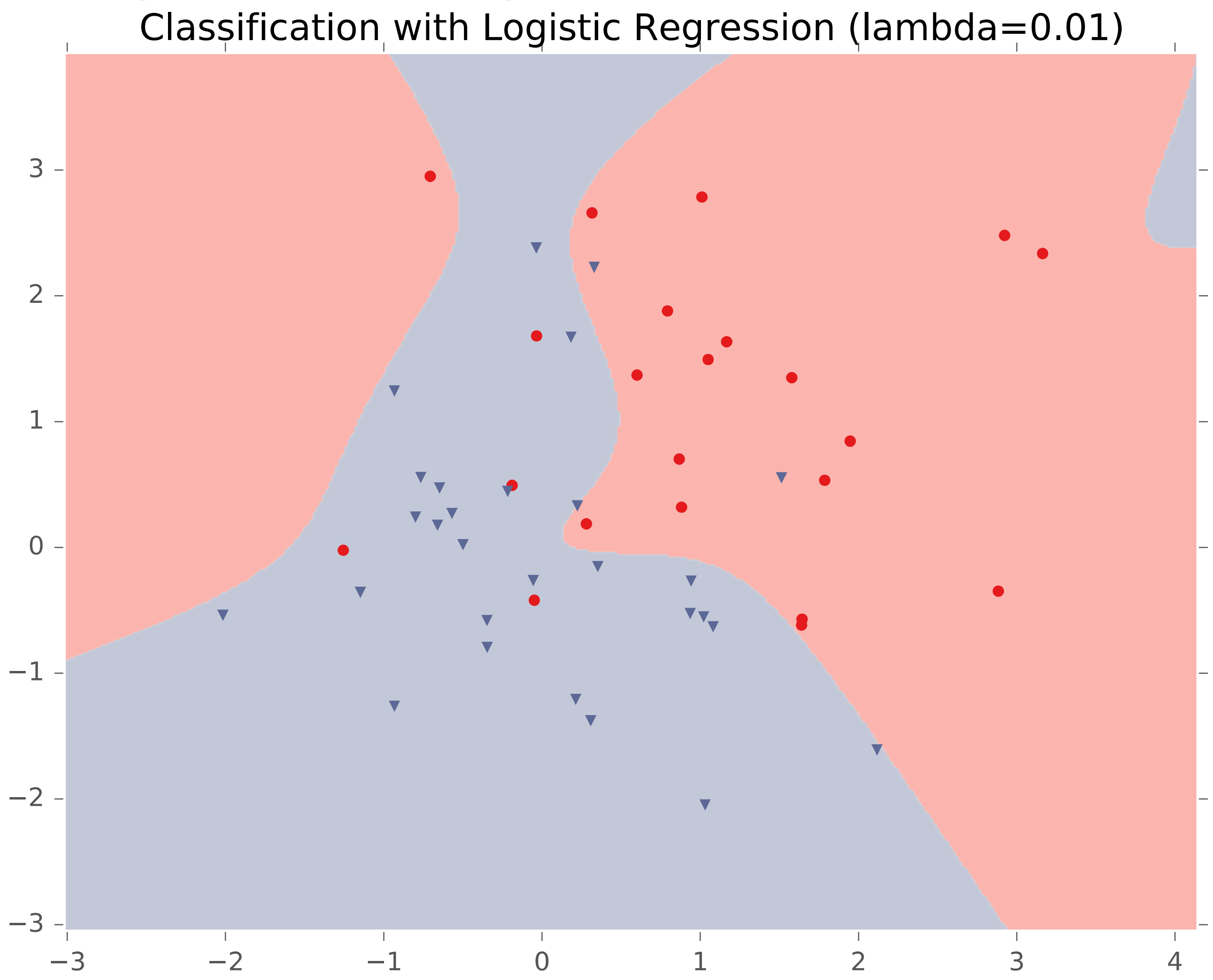
Example: Logistic Regression



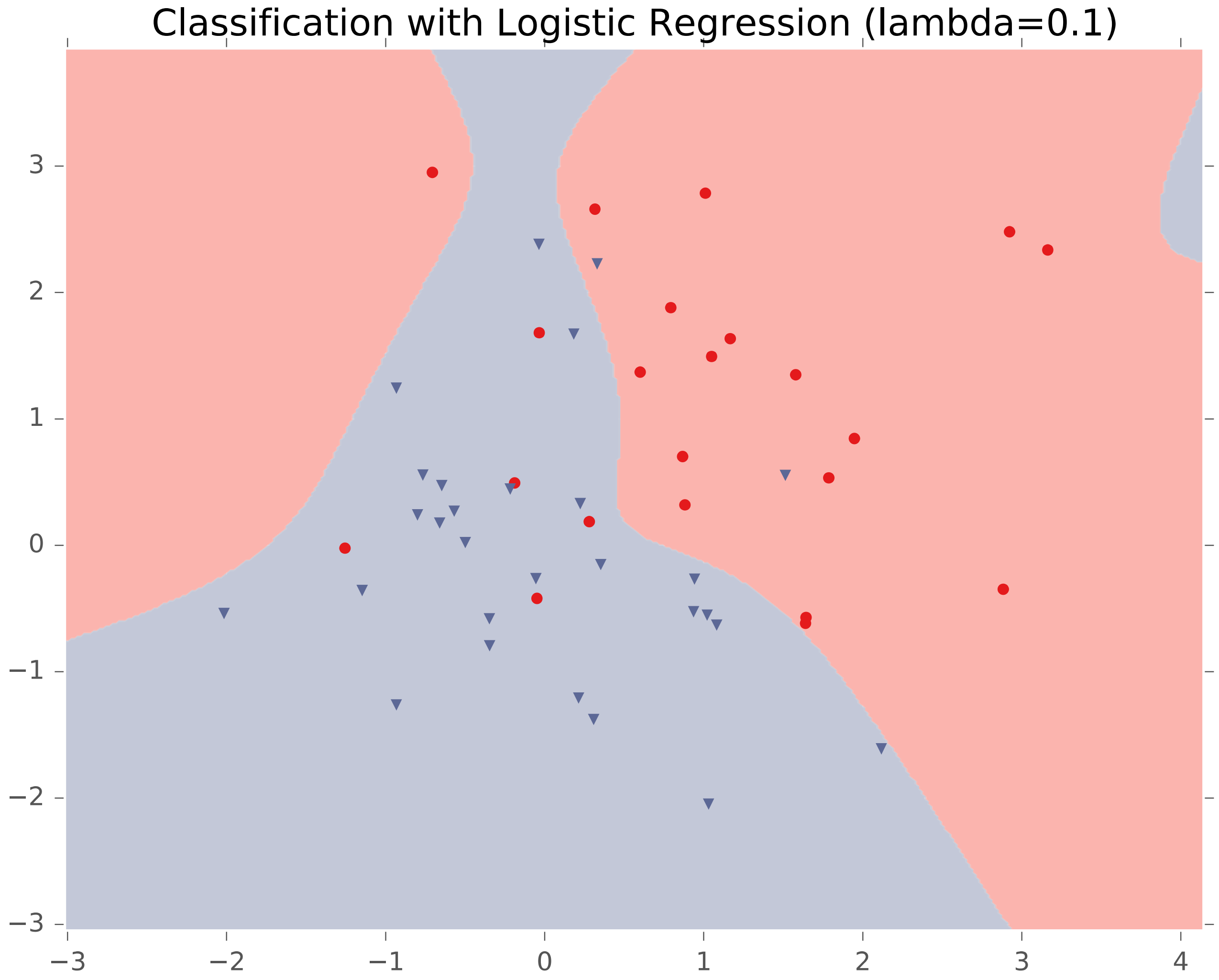
Example: Logistic Regression



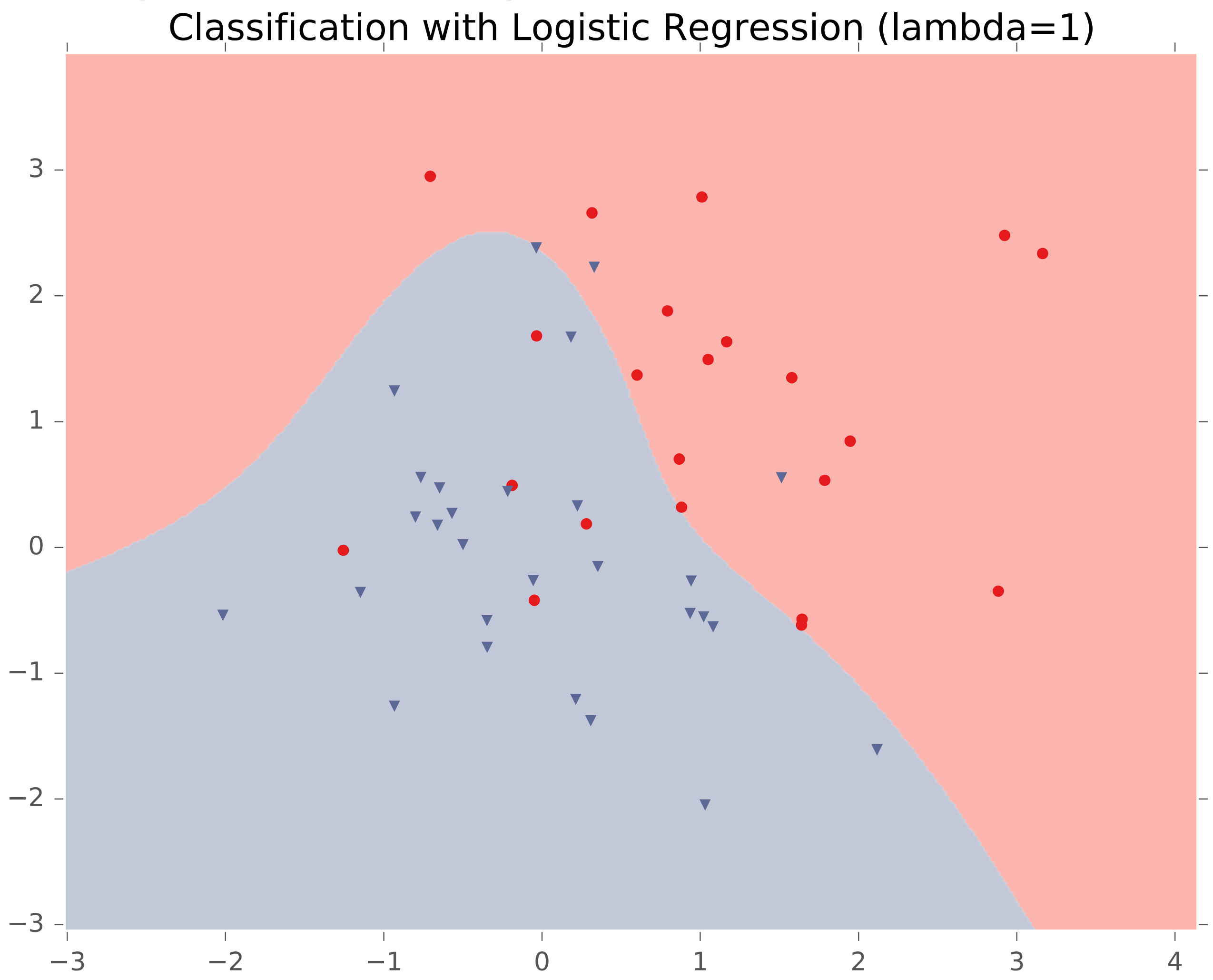
Example: Logistic Regression



Example: Logistic Regression

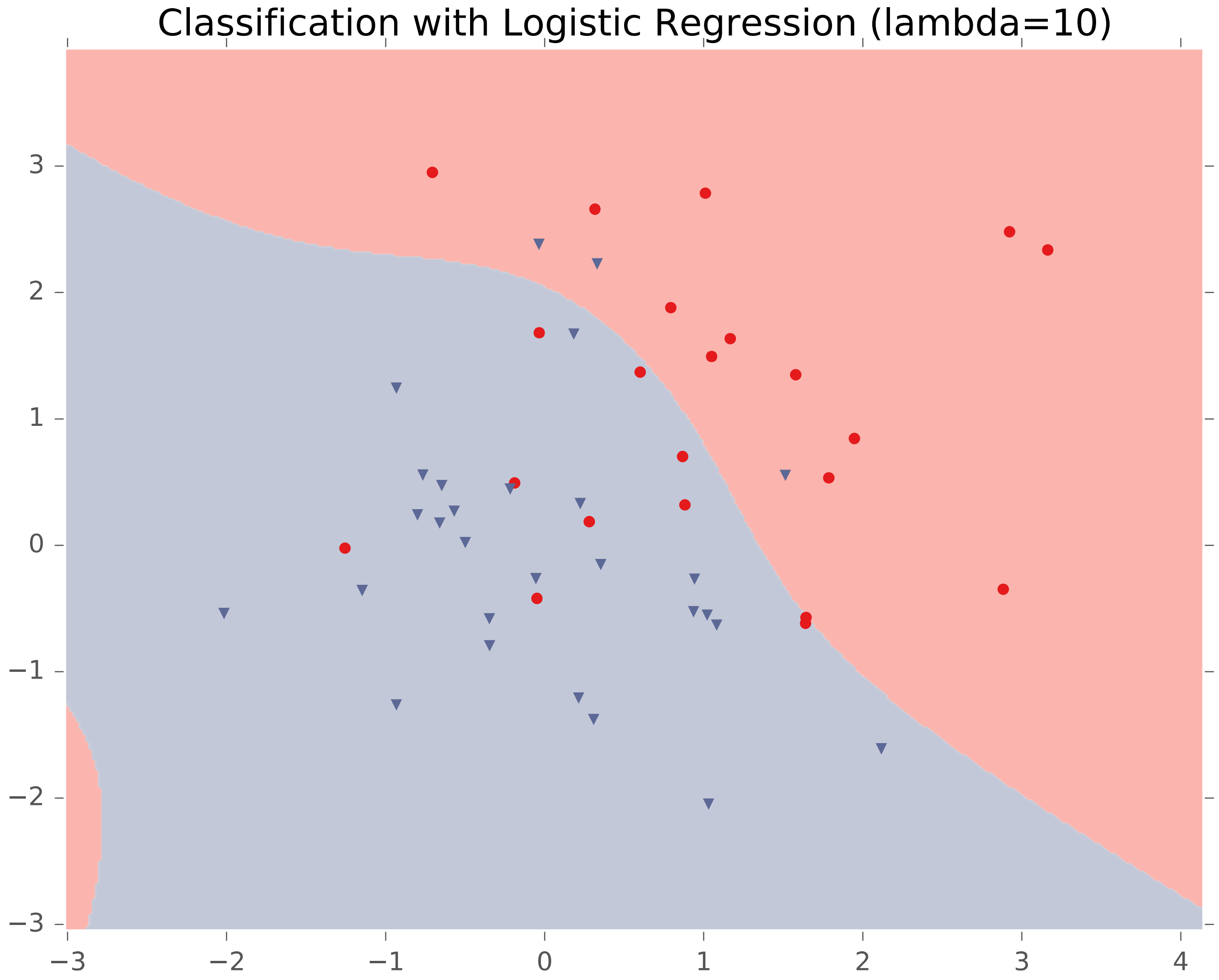


Example: Logistic Regression

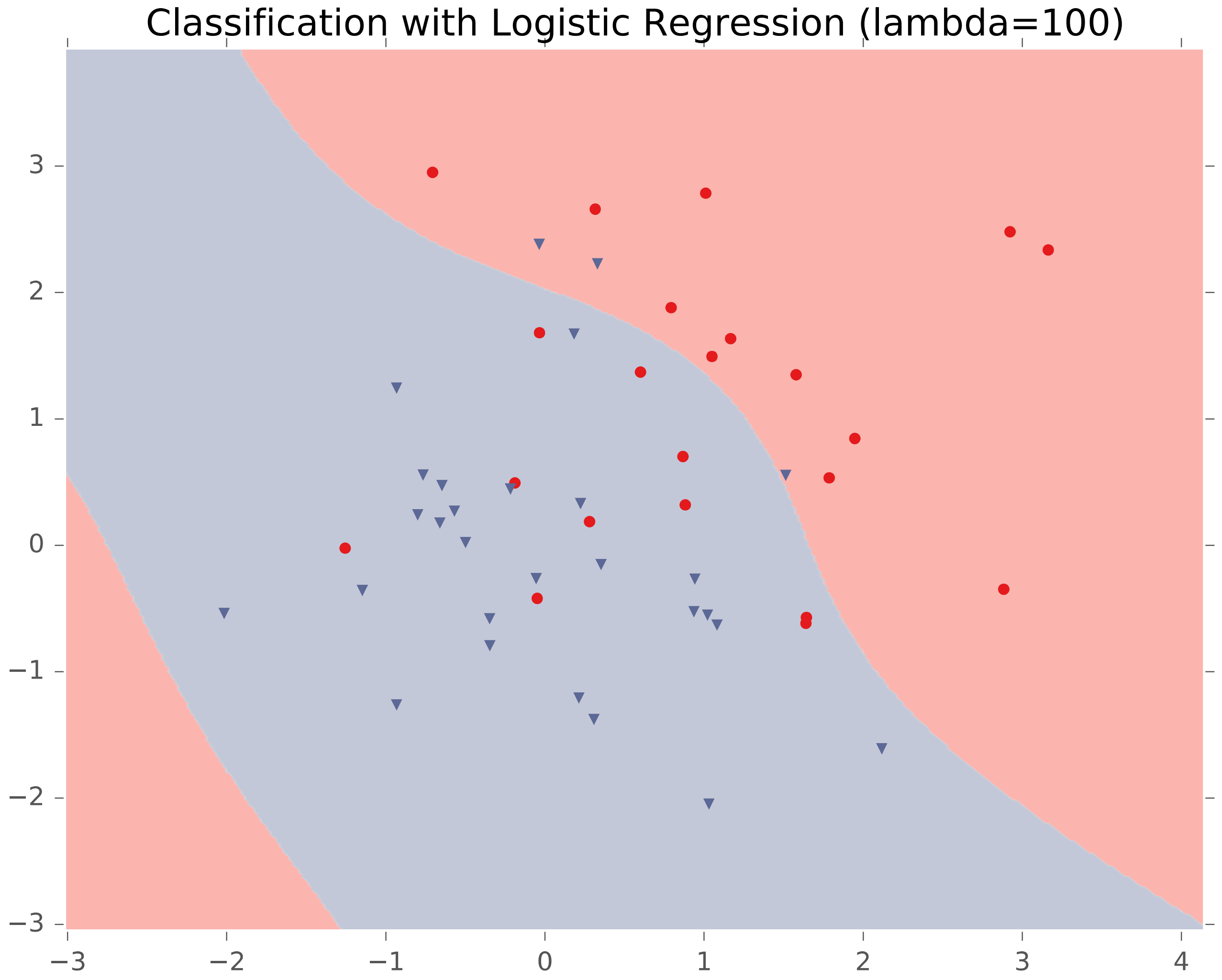


Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

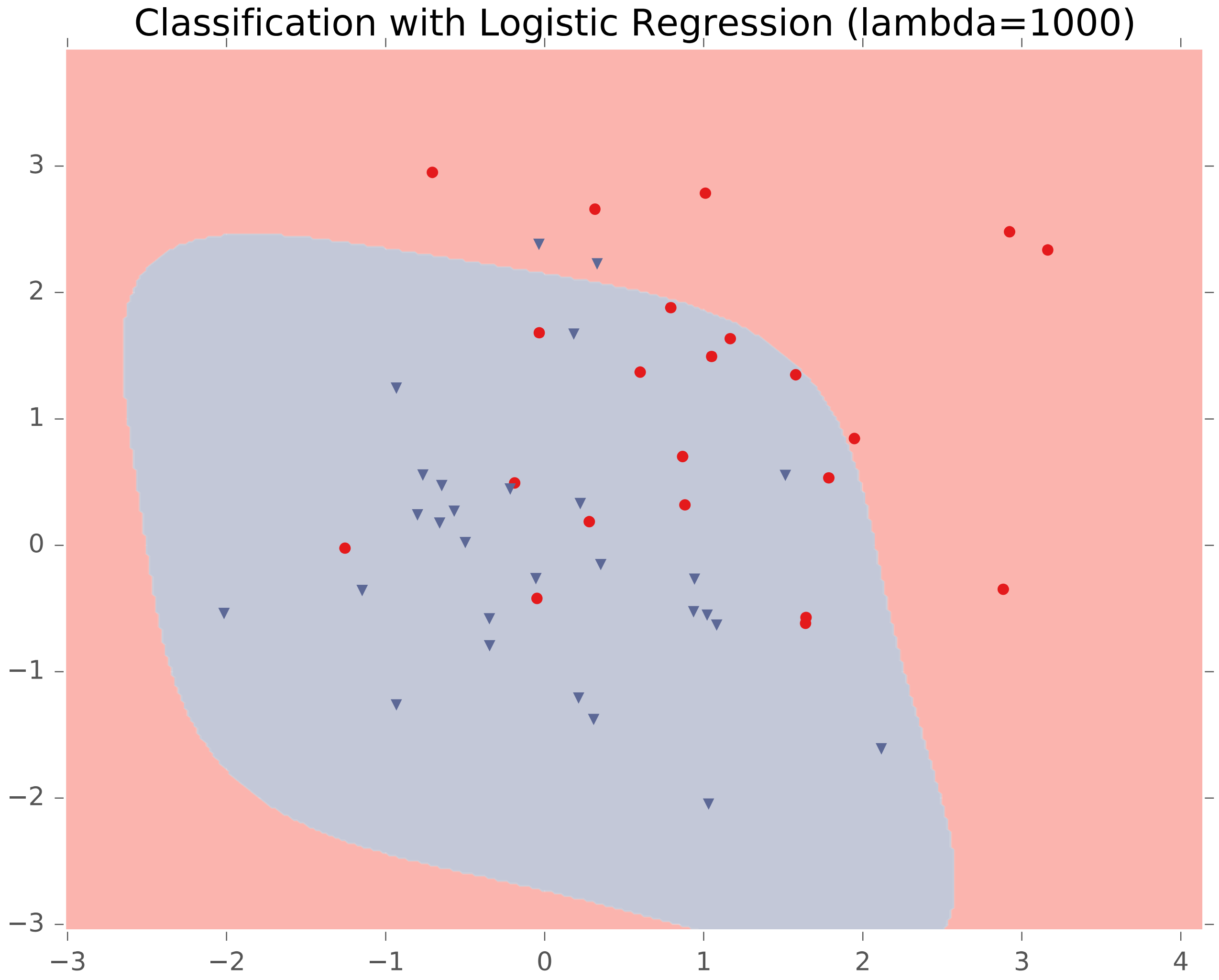


Example: Logistic Regression

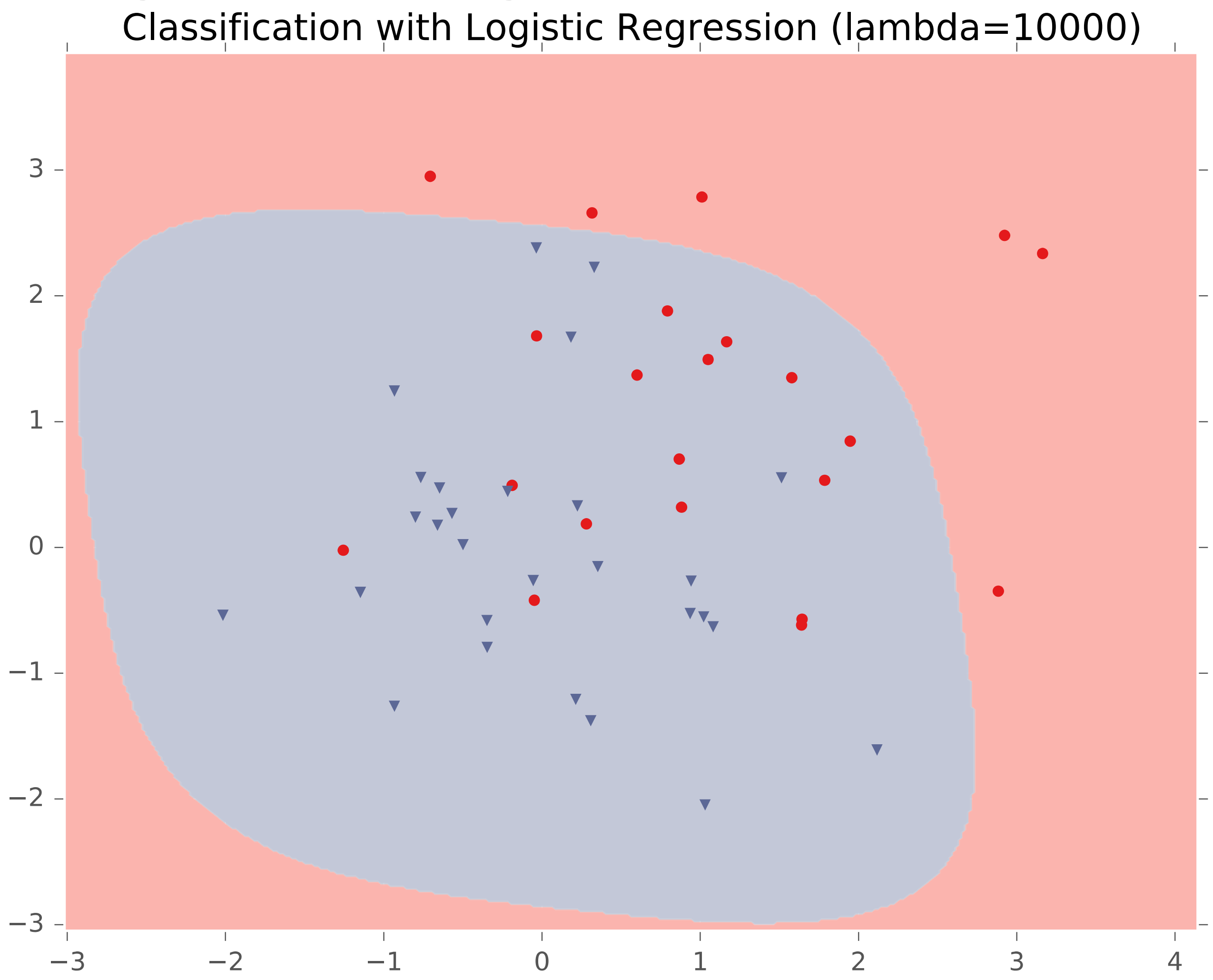


Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

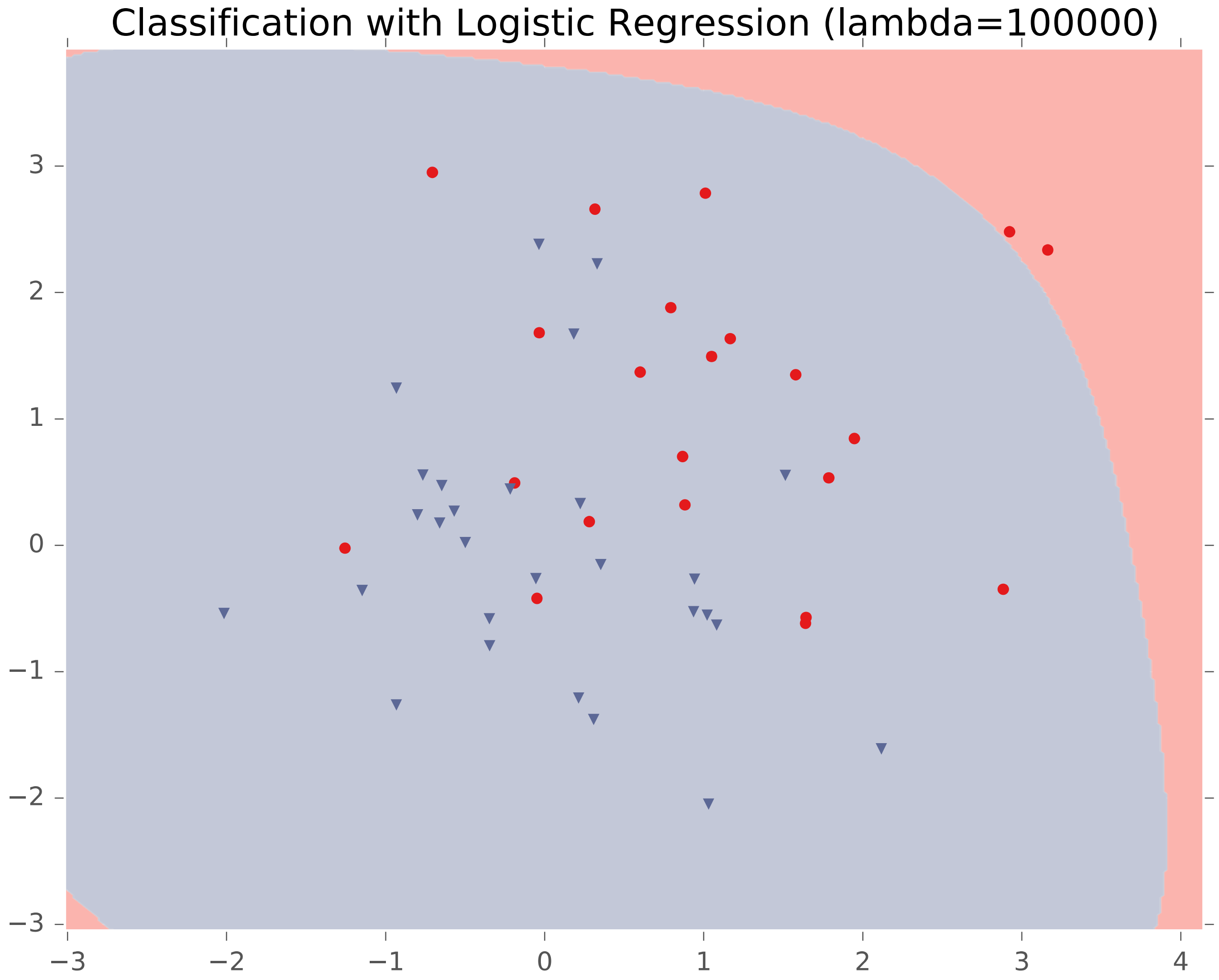


Example: Logistic Regression



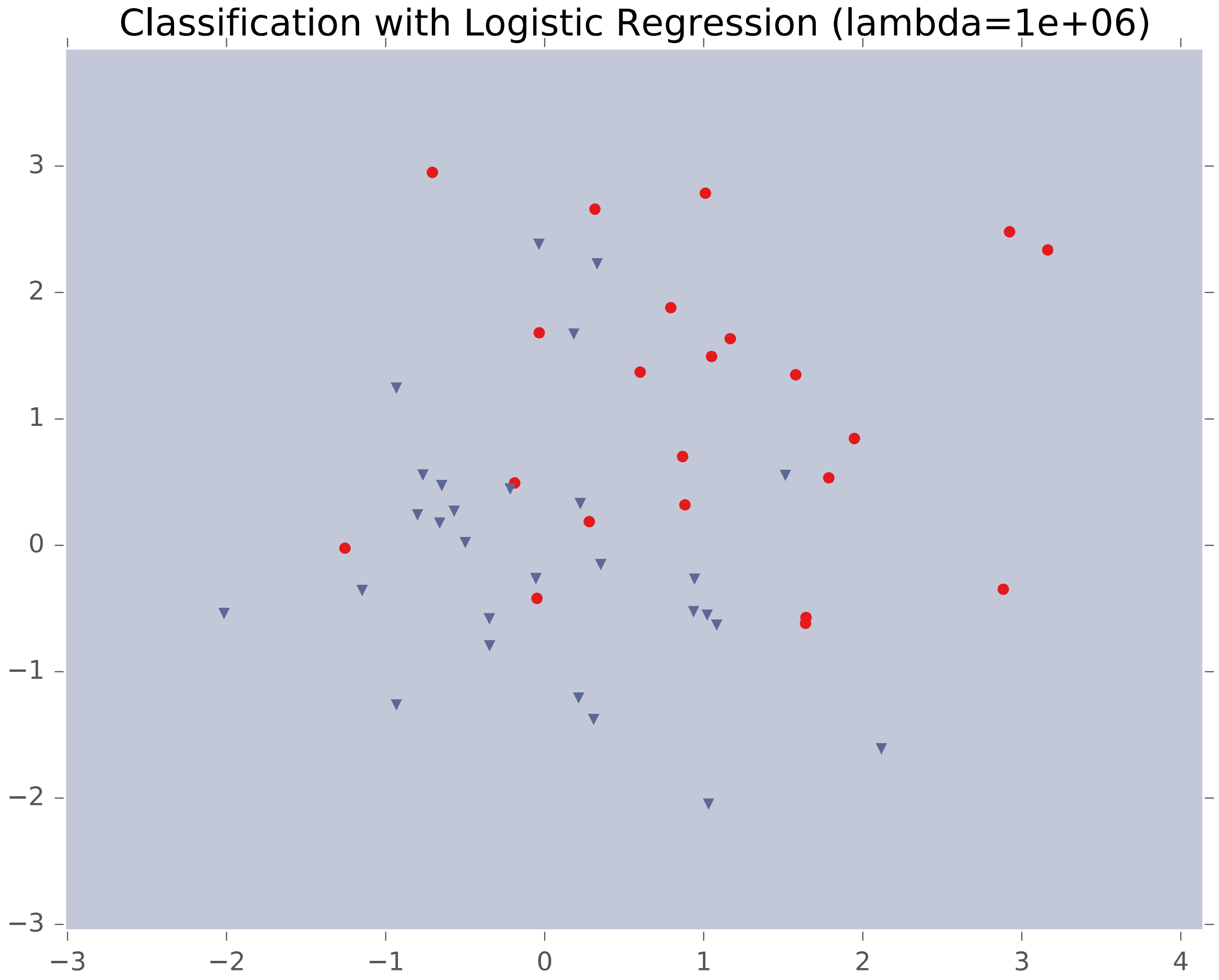
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

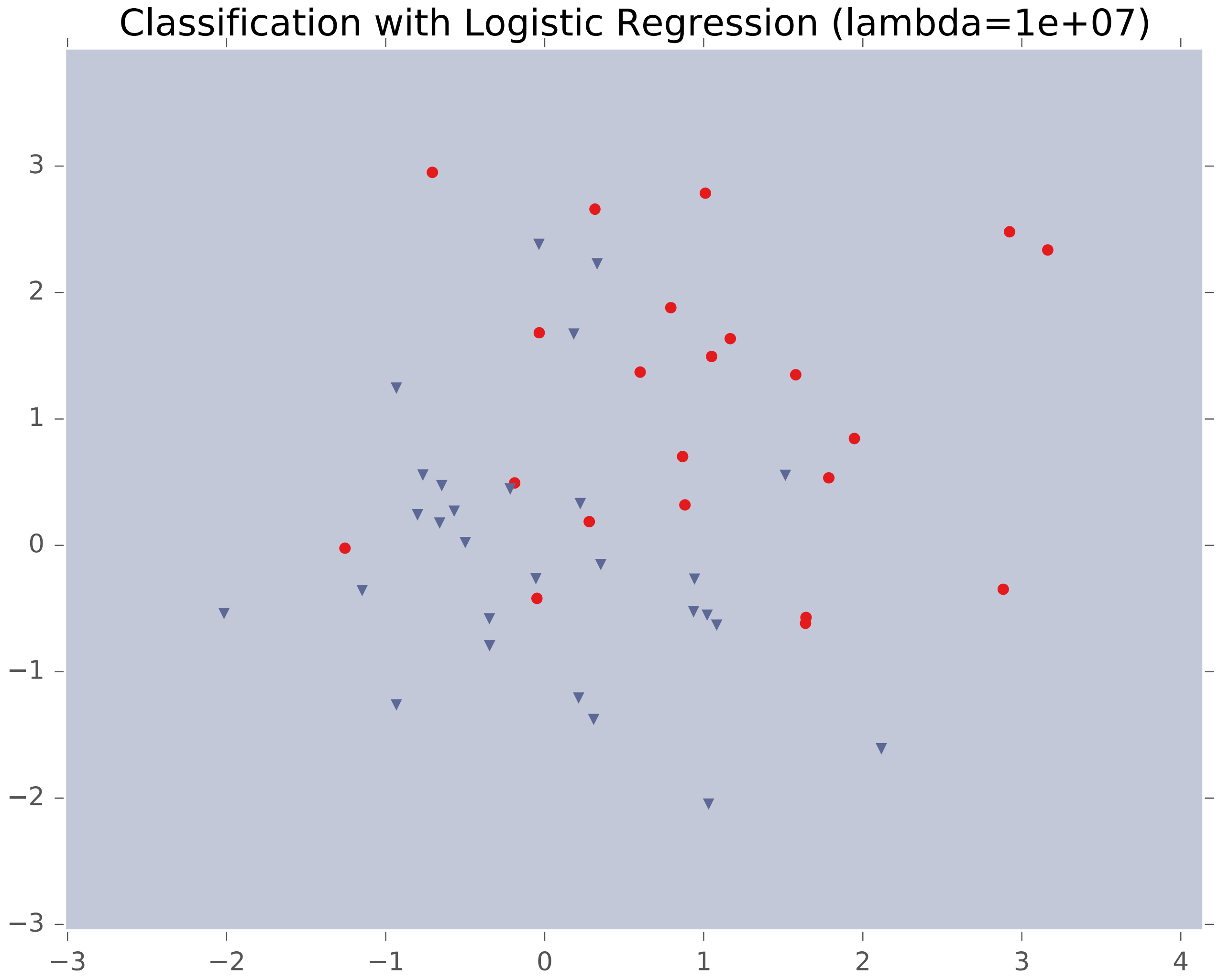


Slide credit: CMU MLD Matt Gormley

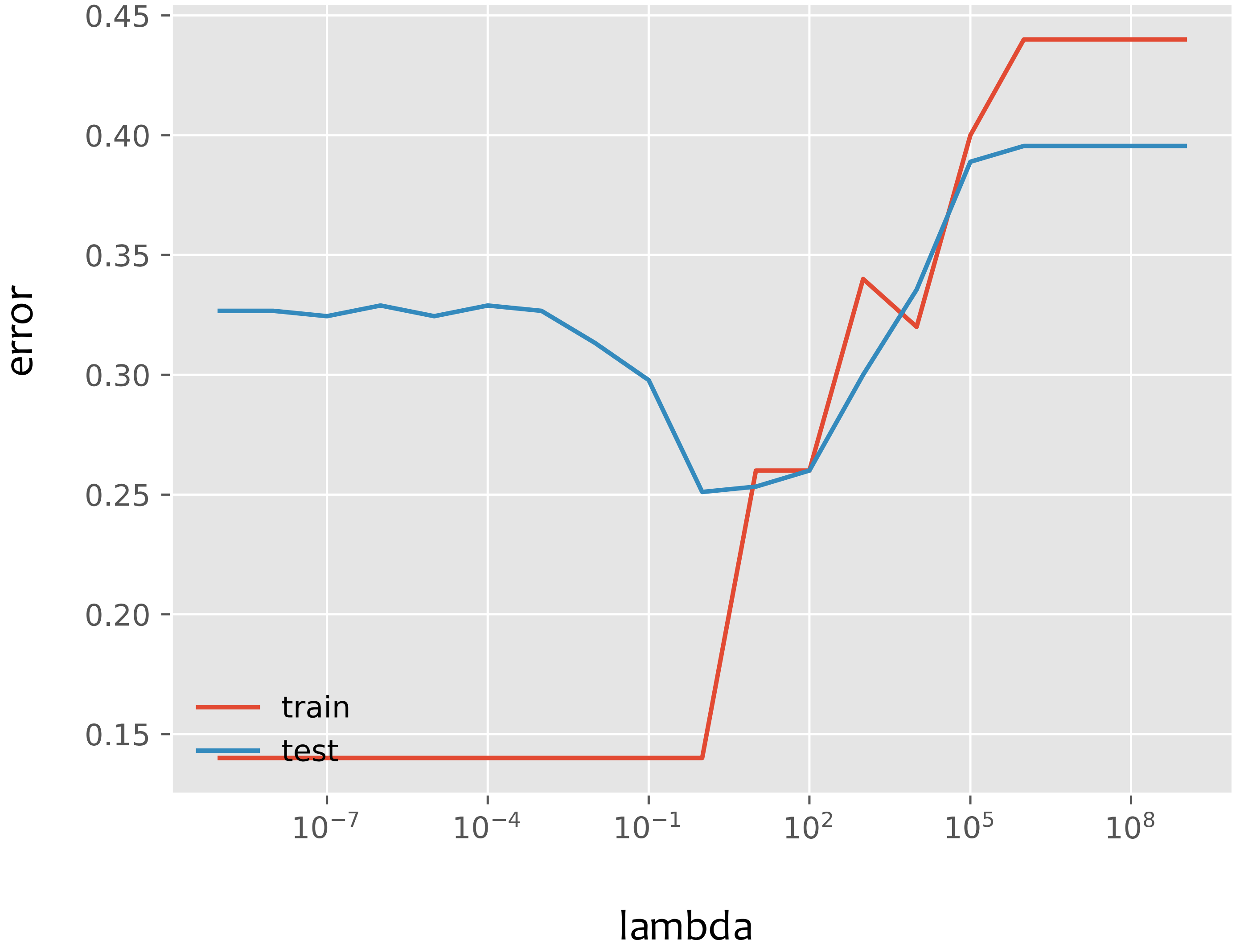
Example: Logistic Regression



Example: Logistic Regression



Example: Logistic Regression



Slide credit: CMU MLD Matt Gormley