Warm-up as You Log In

In-Class Exercise:

Debug the following program which is (incorrectly) attempting to run SGD for multinomial logistic regression

Buggy Program:

```
while not converged:
  for i in shuffle([1,…,N]):
    for k in [1,…,K]:
      theta[k] = theta[k] - gamma * grad(x[i], y[i], theta, k)
```

for k in 1… K

grad[k]

grad[k]

θ_1 new

θ_2 new

θ_3 old

θ_4 old

Assume: grad(x[i], y[i], theta, k) returns the gradient of the negative log-likelihood of the training example (x[i],y[i]) with respect to vector theta[k]. gamma is the learning rate. N = # of examples. K = # of output classes. M = # of features. theta is a K by M matrix.

Announcements

Assignments

- HW4
  - Wed, 10/14, 11:59 pm

Midterm

- Almost finished, grades will hopefully be out later today
- Come talk to us if you’re not happy with how you are doing in the course

Plan

Today

- Wrap-up multi-class logistic regression
- Feature engineering
  - Make our linear methods more powerful!!
- Regularization
  - Make sure they aren’t too powerful ☺

Wrap-up Logistic Regression

Logistic regression slides

Feedback Survey

See Piazza for link

Introduction to Machine Learning

Feature Engineering and Regularization

Instructor: Pat Virtue

How Do We Deal with Real-world Problems

SPAM Classification

How Do We Deal with Real-world Problems

Predicting Rating from Written Movie Review

$x = \{ \text{tex} \}$

$\phi(x) \in \mathbb{R}^m$

$$\begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ \vdots \\ \phi_m(x) \end{bmatrix} \leftarrow \text{\# times fantastic}$$

$$\hat{y} = h(\vec{x}) = g(\vec{\theta}^T \vec{x})$$

$$= g(\vec{\theta}^T \phi(x))$$

How Do We Deal with Real-world Problems

Images: Handwritten Digits

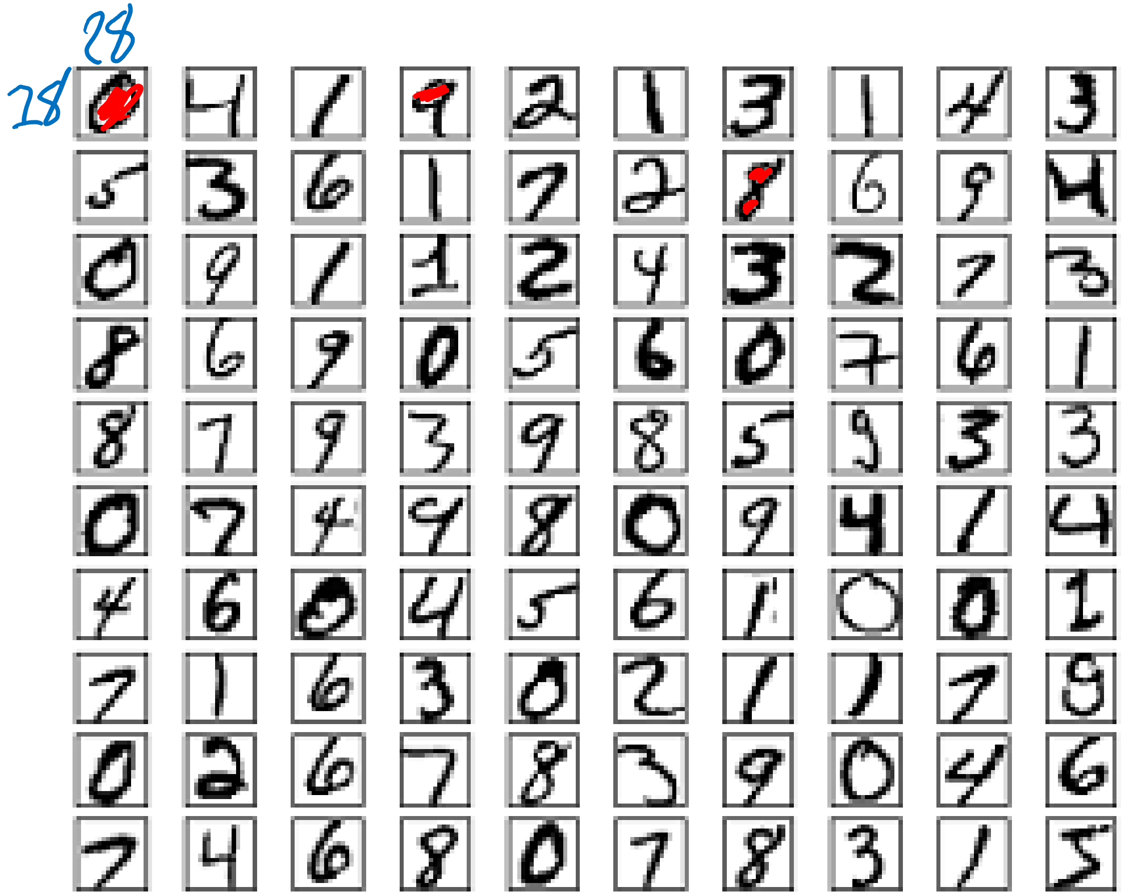
$$\vec{x} \in \mathbb{R}^{784}$$

$$g(\vec{\theta}^T \vec{x})$$

$$\begin{bmatrix} \phi_1(\vec{x}) \\ \phi_2(\vec{x}) \\ \\ \phi_m(\vec{x}) \end{bmatrix}$$

$\leftarrow$ number of holes

$\leftarrow$ ratio fg/bg



How Do We Deal with Real-world Problems

Images: Face Recognition

How Do We Deal with Real-world Problems

Images: Animal Classification

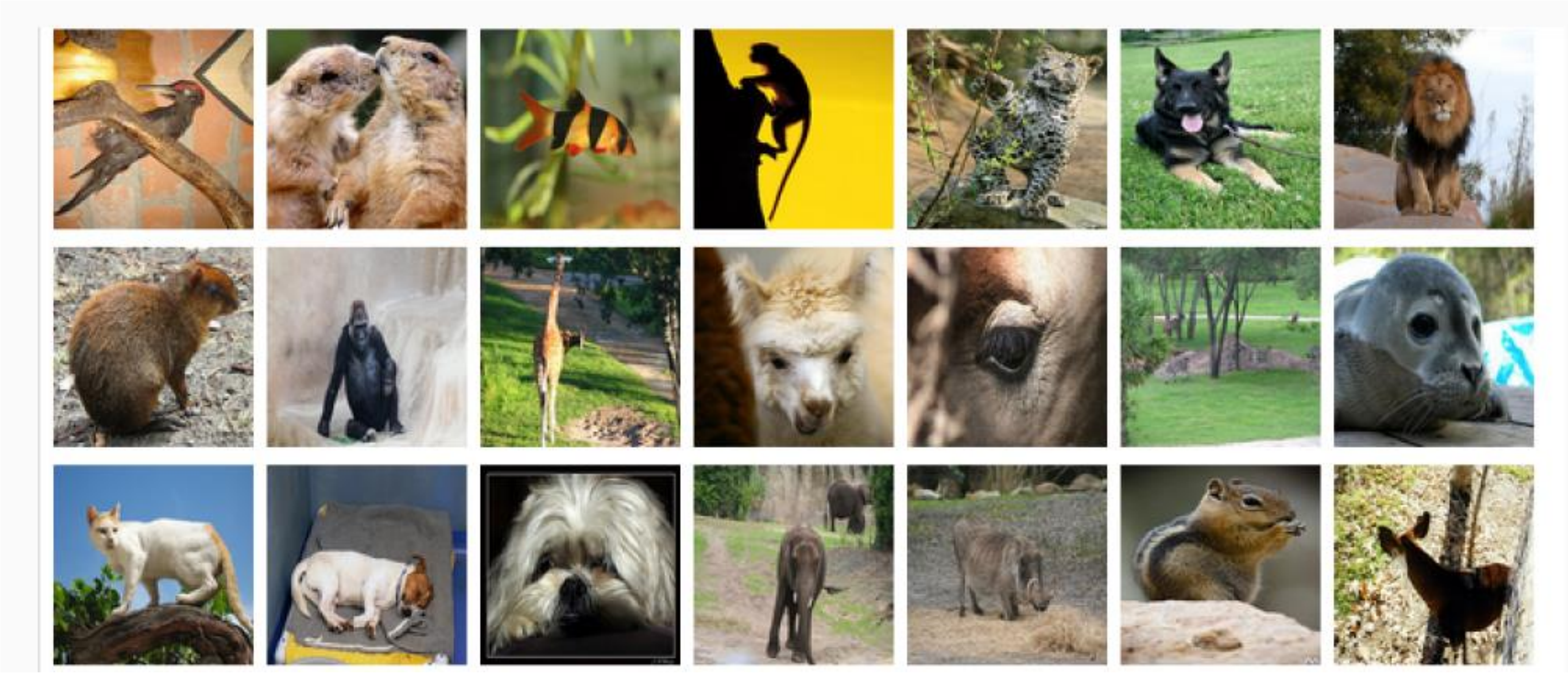


Image: ImageNet

How Do We Deal with Real-world Problems

Images: Animal Classification

input image

visualization of HOG features

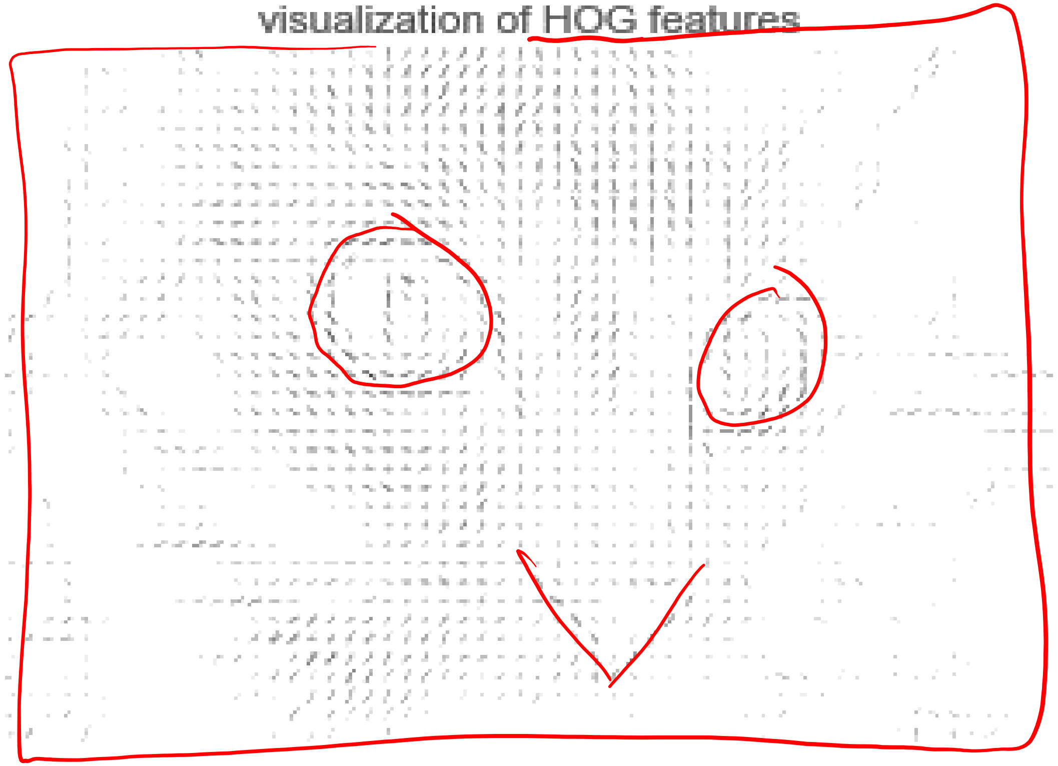


Image: https://jakevdp.github.io/PythonDataScienceHandbook/05.14-image-features.html

How Do We Deal with Real-world Problems

Preview: Neural Networks

x $\xrightarrow{\phi(x)}$ z → linear classification (SVM)

input image

visualization of HOG features

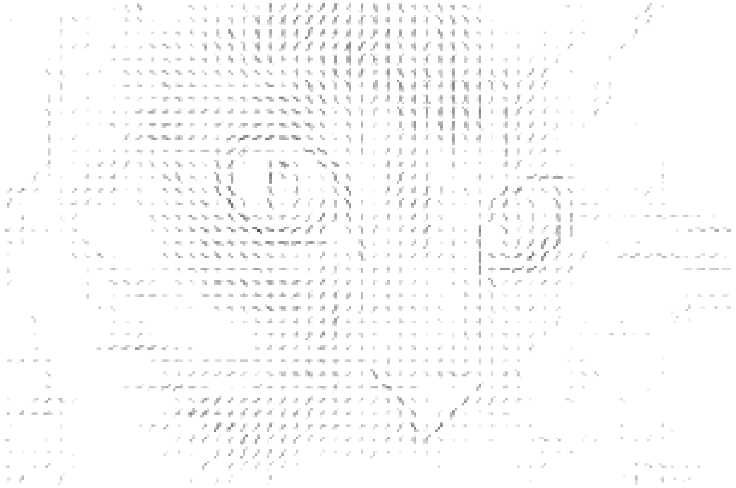
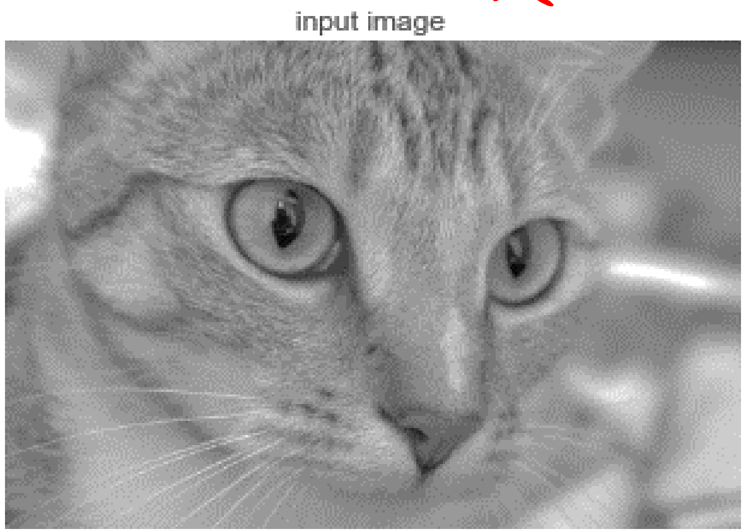


Image: https://jakevdp.github.io/PythonDataScienceHandbook/05.14-image-features.html

How Do We Deal with Real-world Problems

Preview: Neural Networks



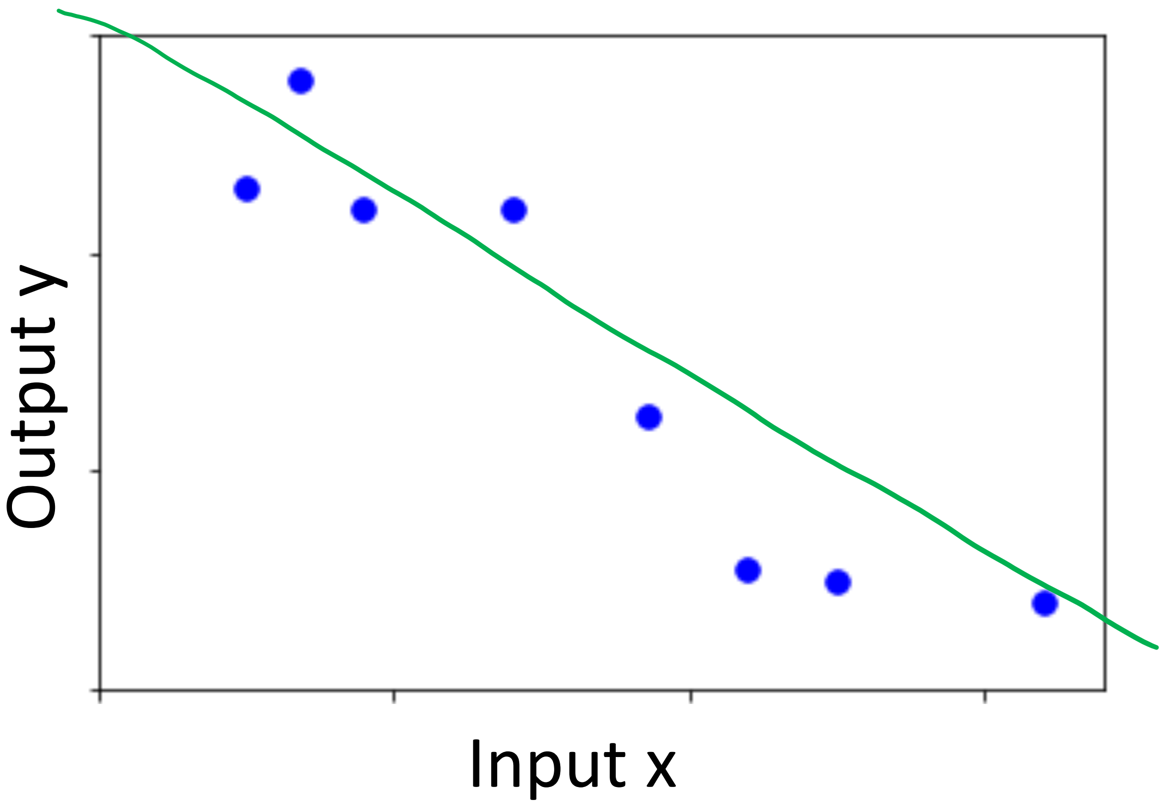
$x \xrightarrow{\phi_A(x)} z_A \xrightarrow{\phi(z_B)} z_B \xrightarrow{\phi(z_B)} z_C \rightarrow$

Lin. Class.
Multiclass
Log. Reg.

Image: https://jakevdp.github.io/PythonDataScienceHandbook/05.14-image-features.html

Linear Data

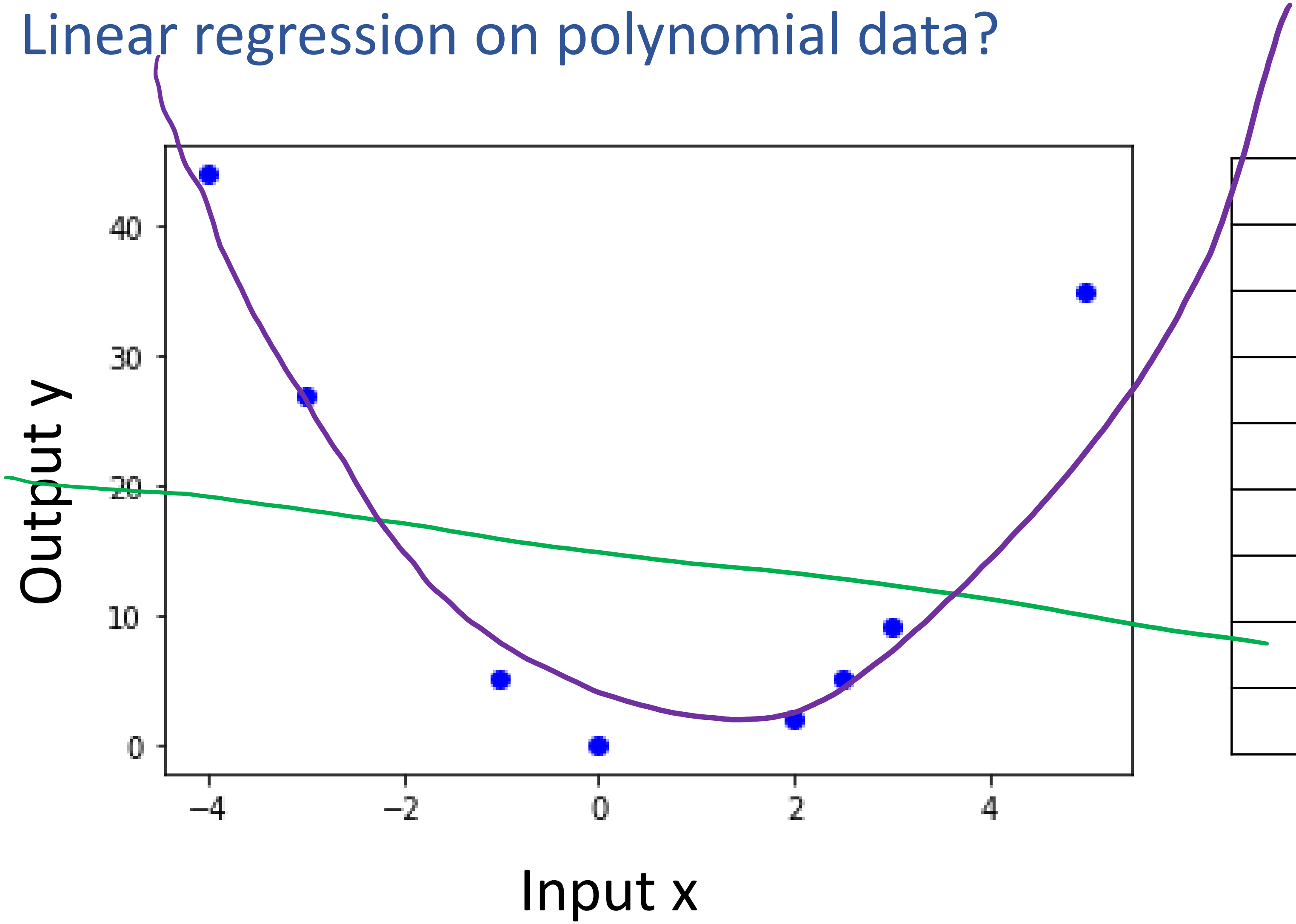
Regression on simple linear dataset



Non-linear Data

Linear regression on polynomial data?

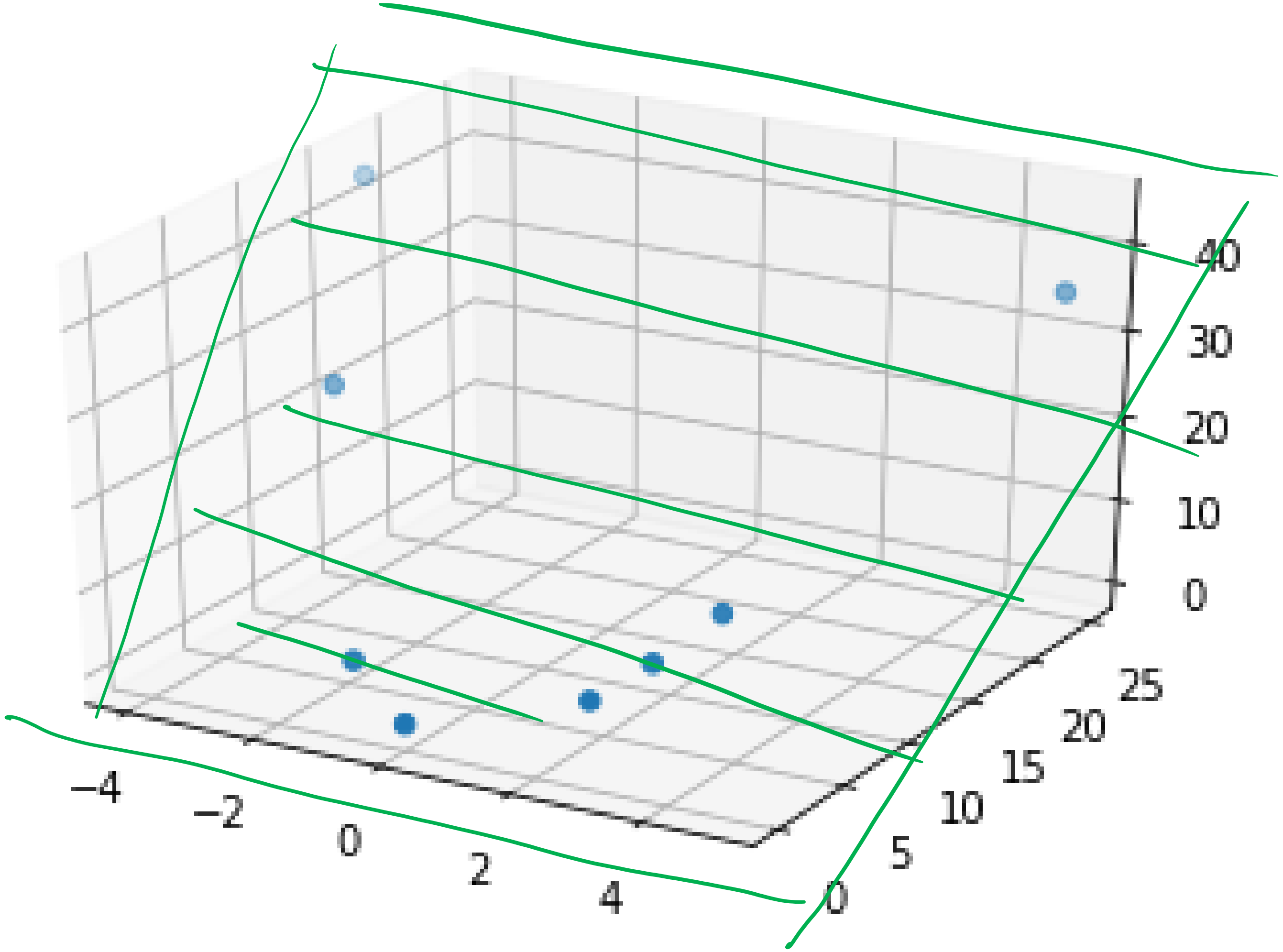
$D_1 \rightarrow D_2$



y	x_1
44	-4
27	-3
5	-1
0	0
2	2
5	2.5
9	3
35	5

Non-linear Data

Regression on simple linear dataset



D_2

y	x_1	x_2
44	-4	16
27	-3	9
5	-1	1
0	0	0
2	2	4
5	2.5	6.25
9	3	9
35	5	25

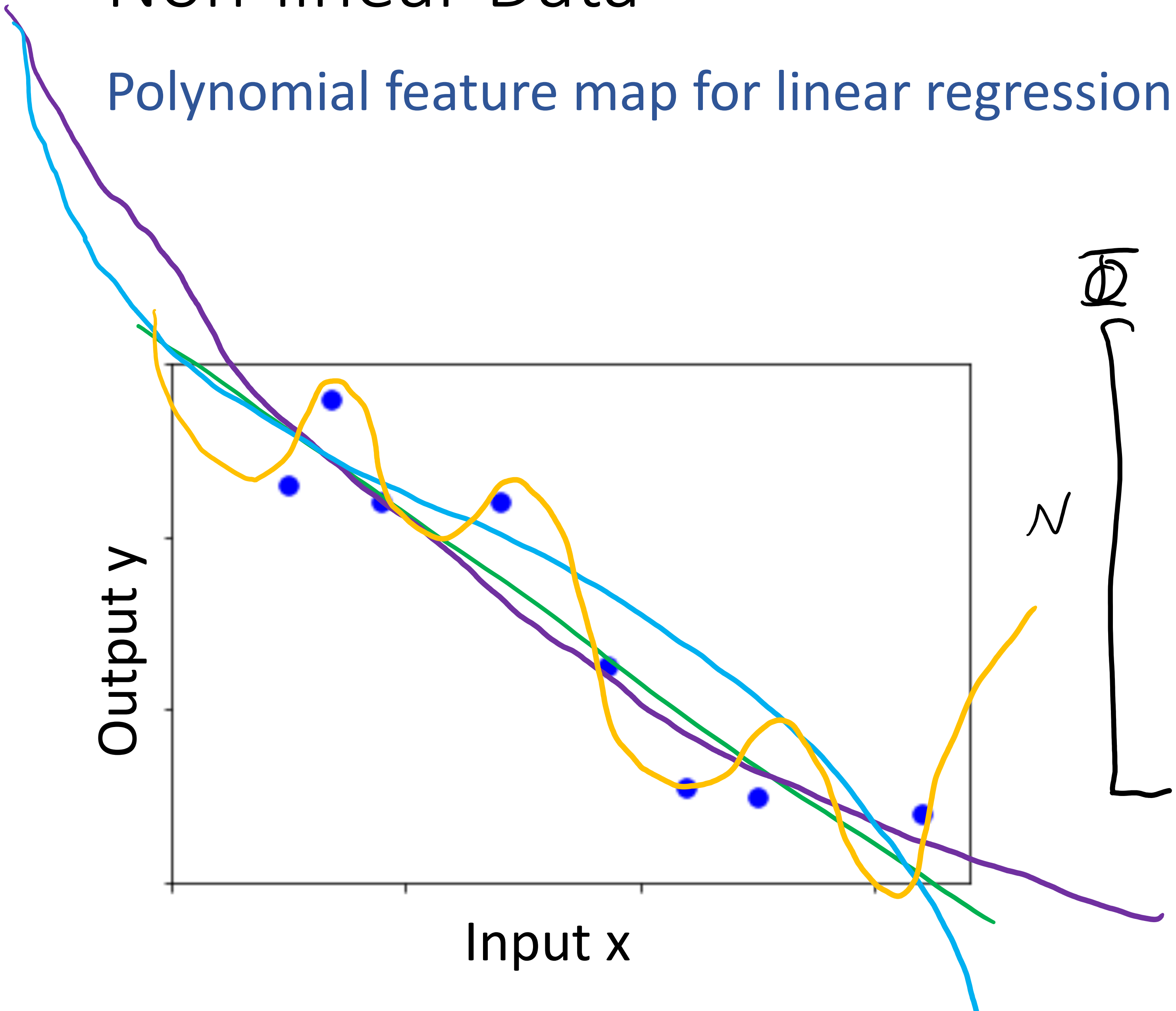
x_1^2

$y = w_1 x_1 + w_2 x_2$

x_1^2

Non-linear Data

Polynomial feature map for linear regression



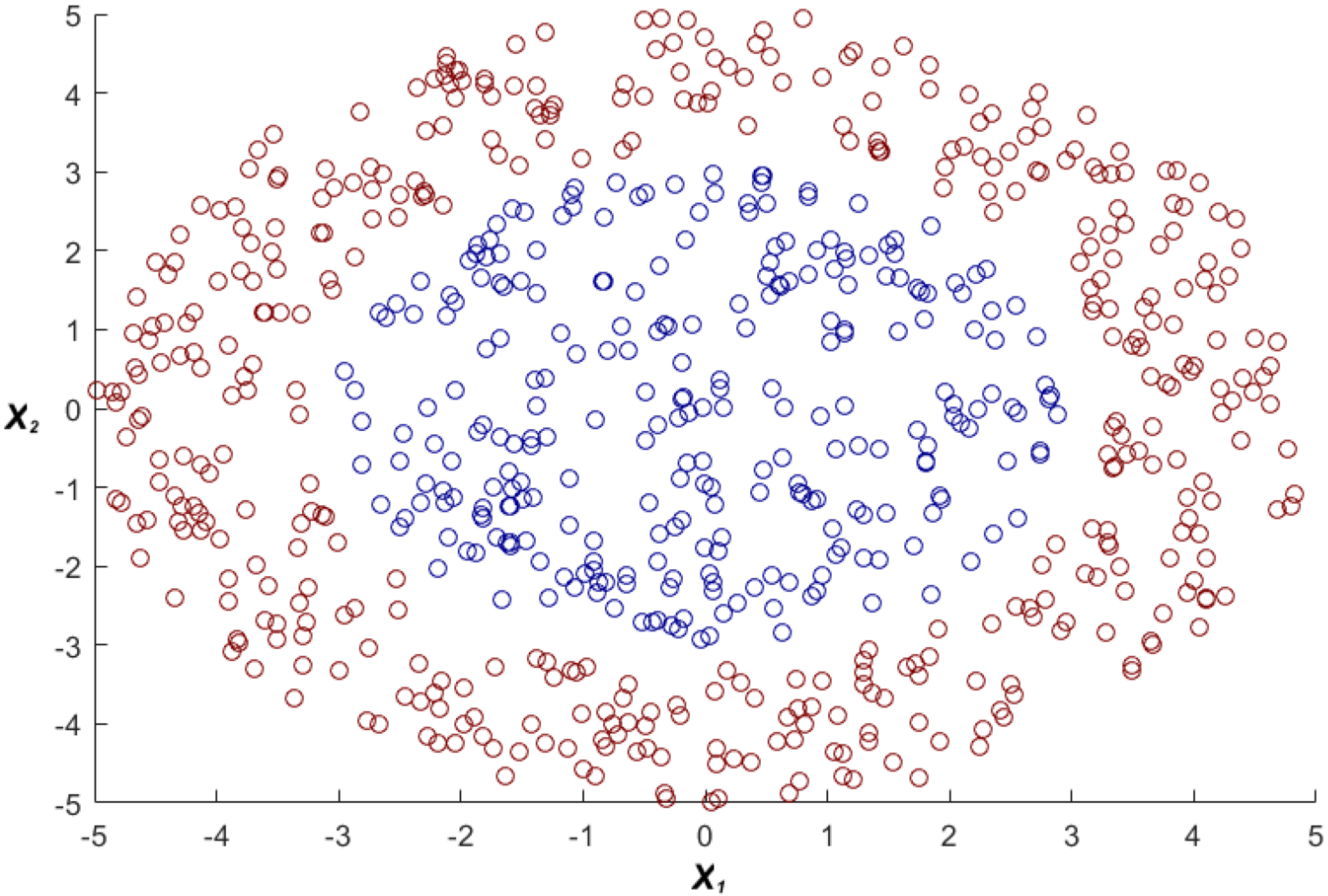
$x \longrightarrow \phi(x)$

$X \longrightarrow \Phi$

$$\Phi = N\left[\begin{array}{cccccc} 1 & x^{(1)} & {x^{(1)}}^2 & {x^{(1)}}^3 & \cdots & {x^{(1)}}^5 \\ 1 & x^{(2)} & {x^{(2)}}^2 & {x^{(2)}}^3 & & \\ & \vdots & & & & \\ 1 & x^{(N)} & {x^{(N)}}^2 & & & \end{array}\right]$$

Non-linear Data

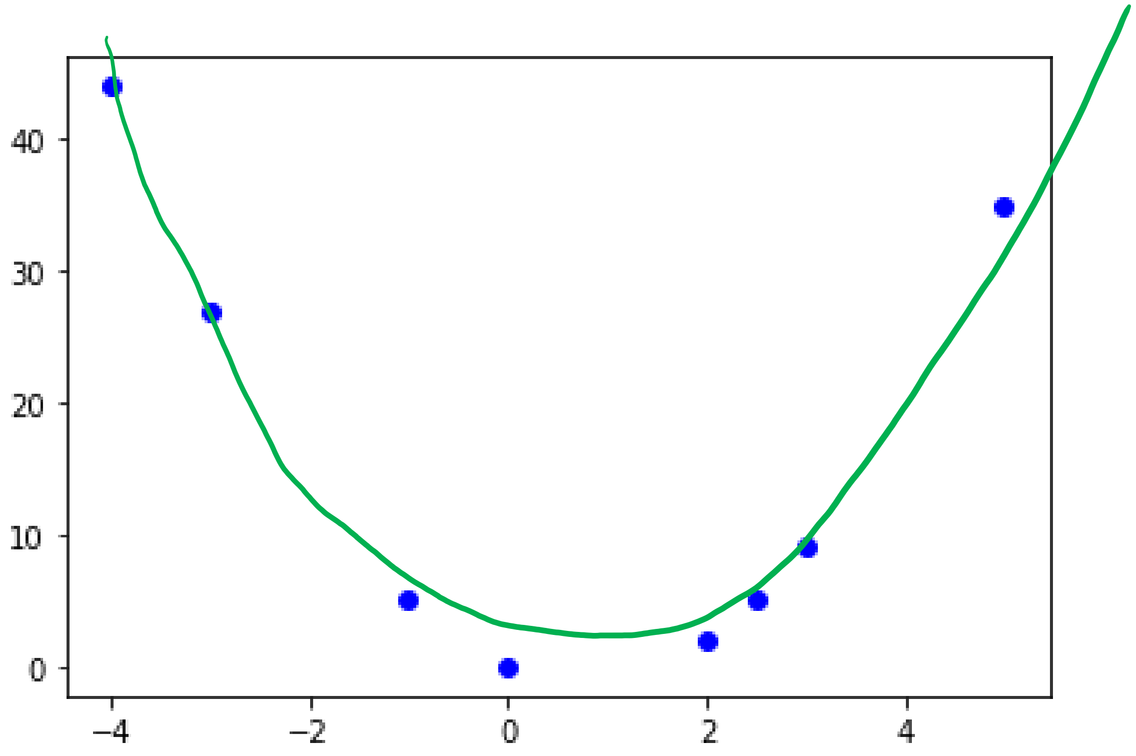
Polynomial feature map for linear classification

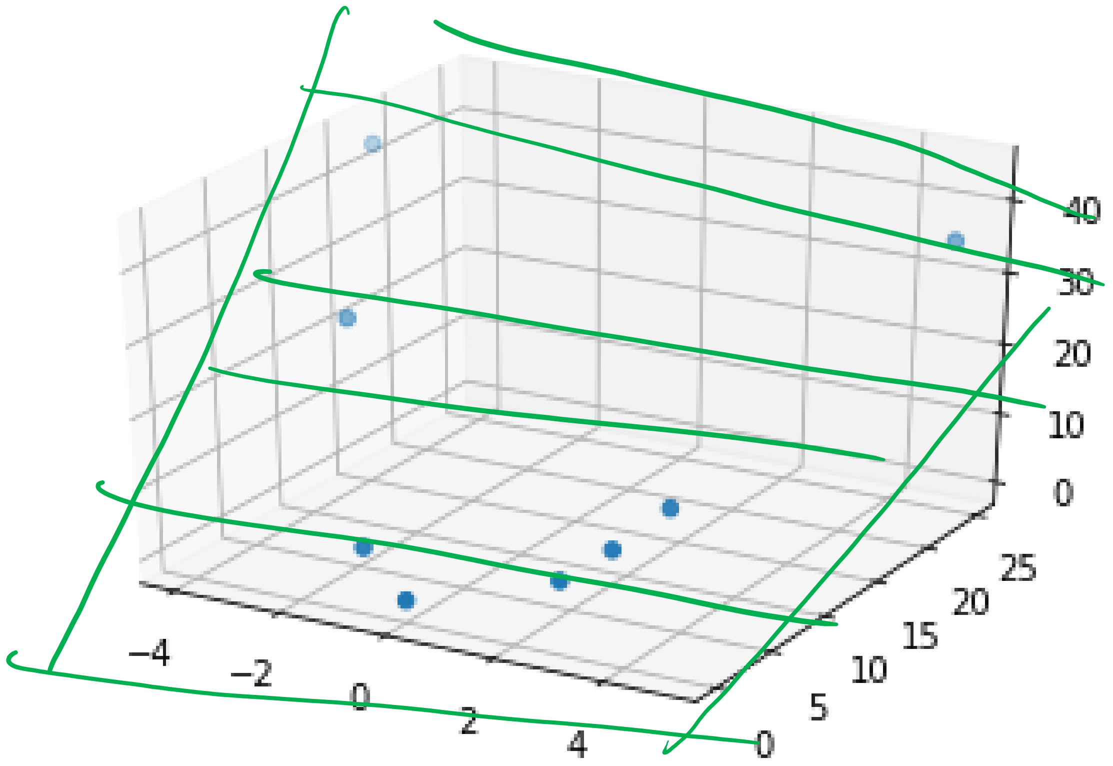


https://www.youtube.com/watch?v=3liCbRZPrZA

Feature Maps

Two ways to think about it



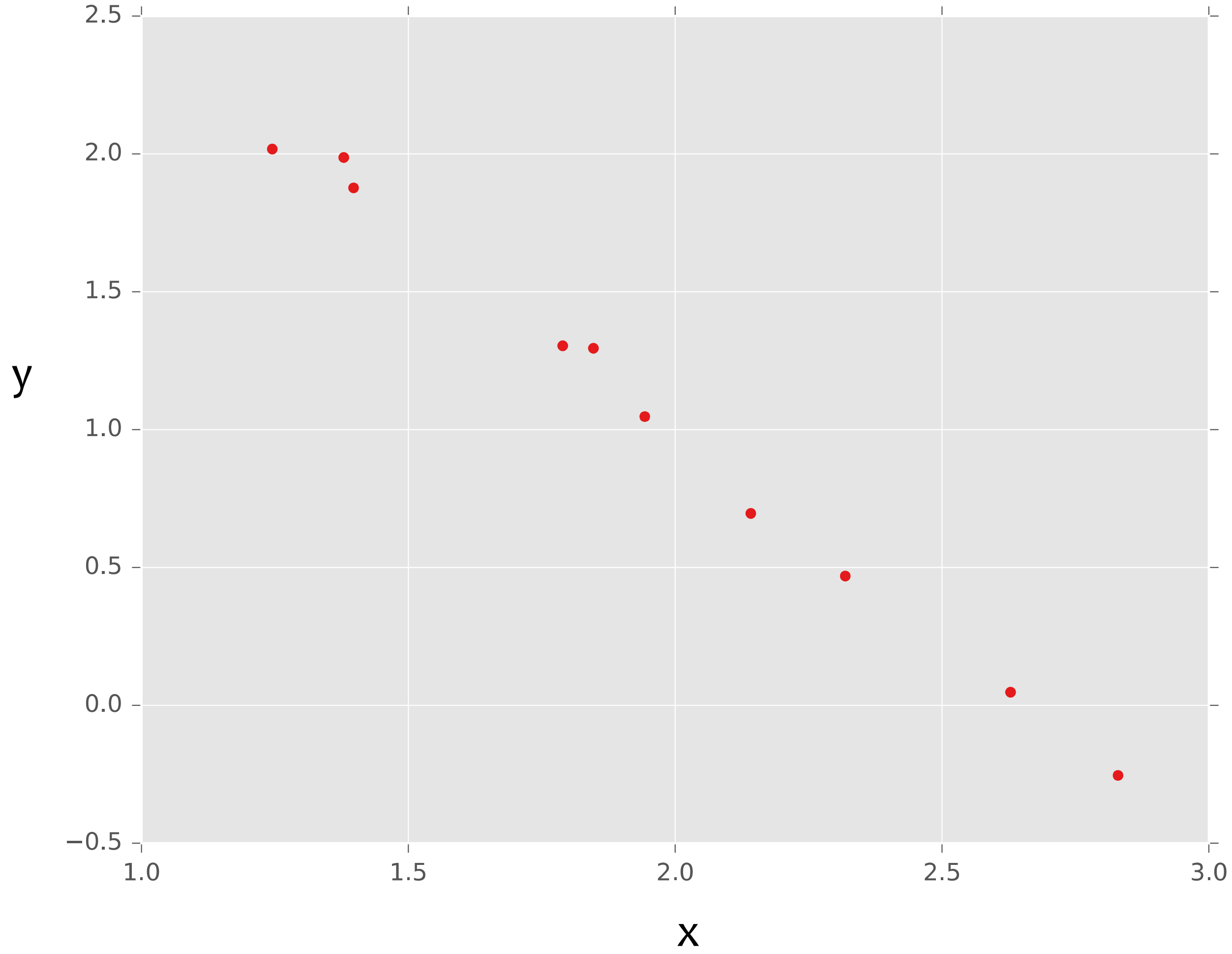


Example: Linear Regression

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

y	x
2.0	1.2
1.3	1.7
0.1	2.7
1.1	1.9

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

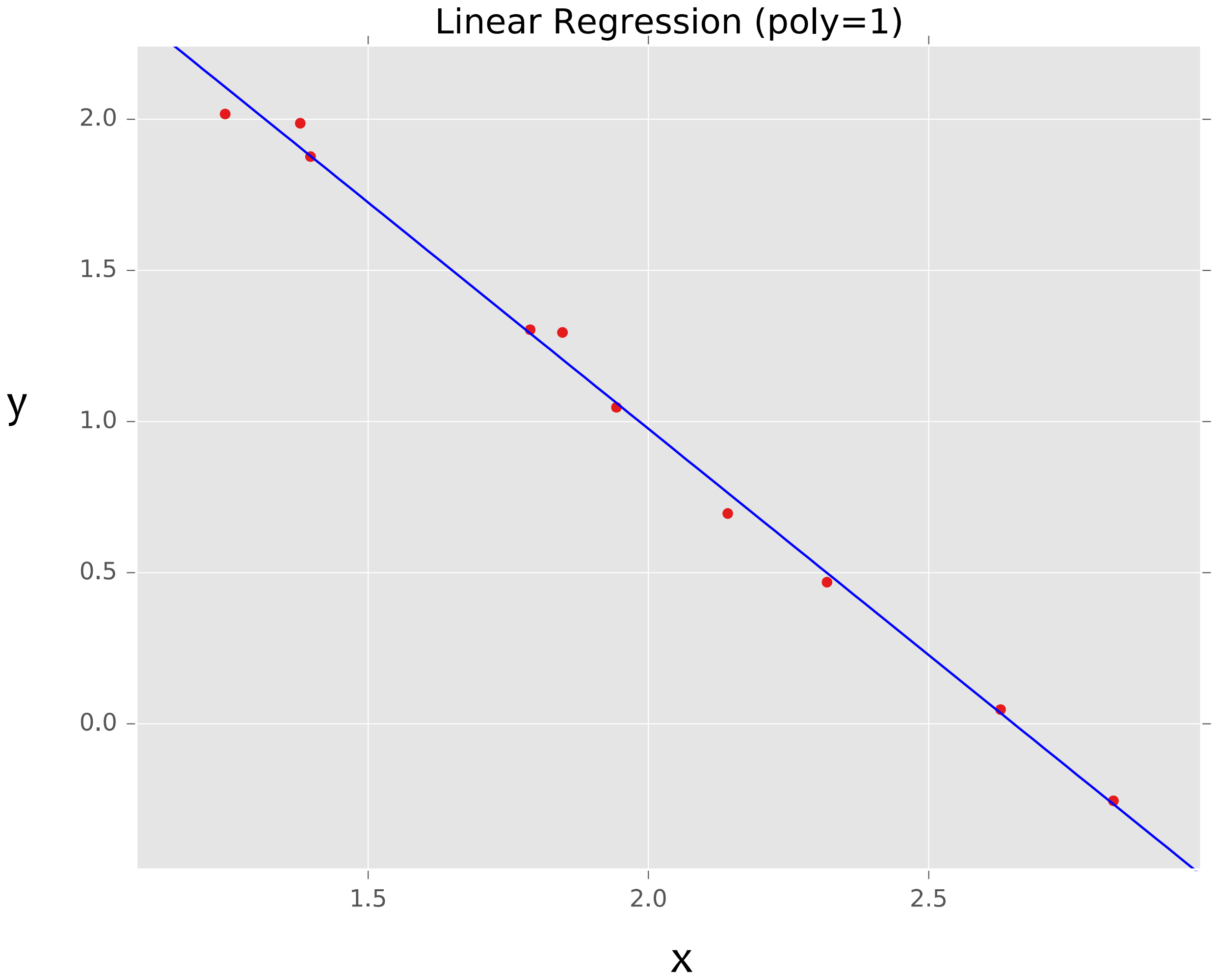


Example: Linear Regression

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

y	x
2.0	1.2
1.3	1.7
0.1	2.7
1.1	1.9

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



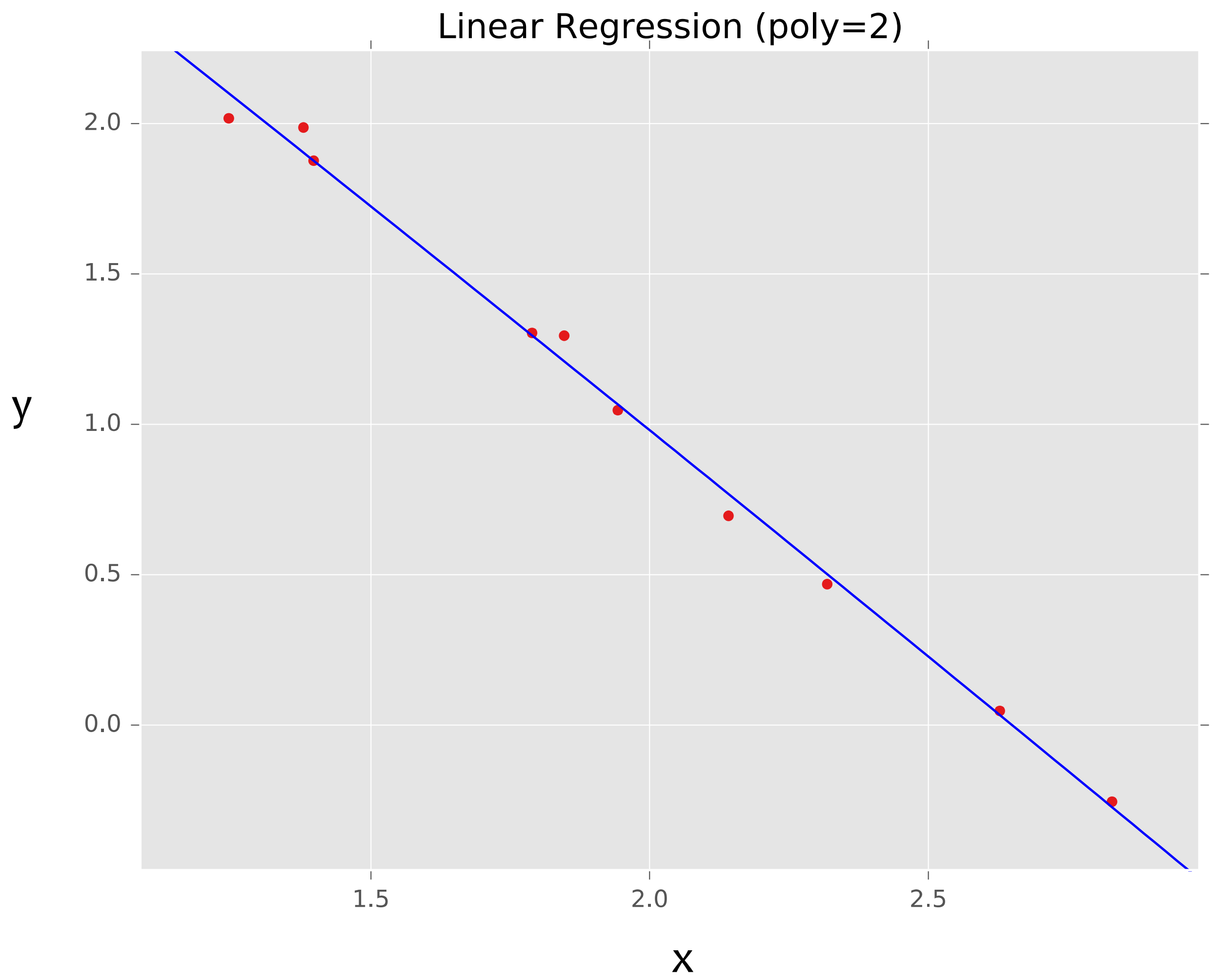
Slide credit: CMU MLD Matt Gormley

Example: Linear Regression

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

y	x	x^2
2.0	1.2	$(1.2)^2$
1.3	1.7	$(1.7)^2$
0.1	2.7	$(2.7)^2$
1.1	1.9	$(1.9)^2$

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



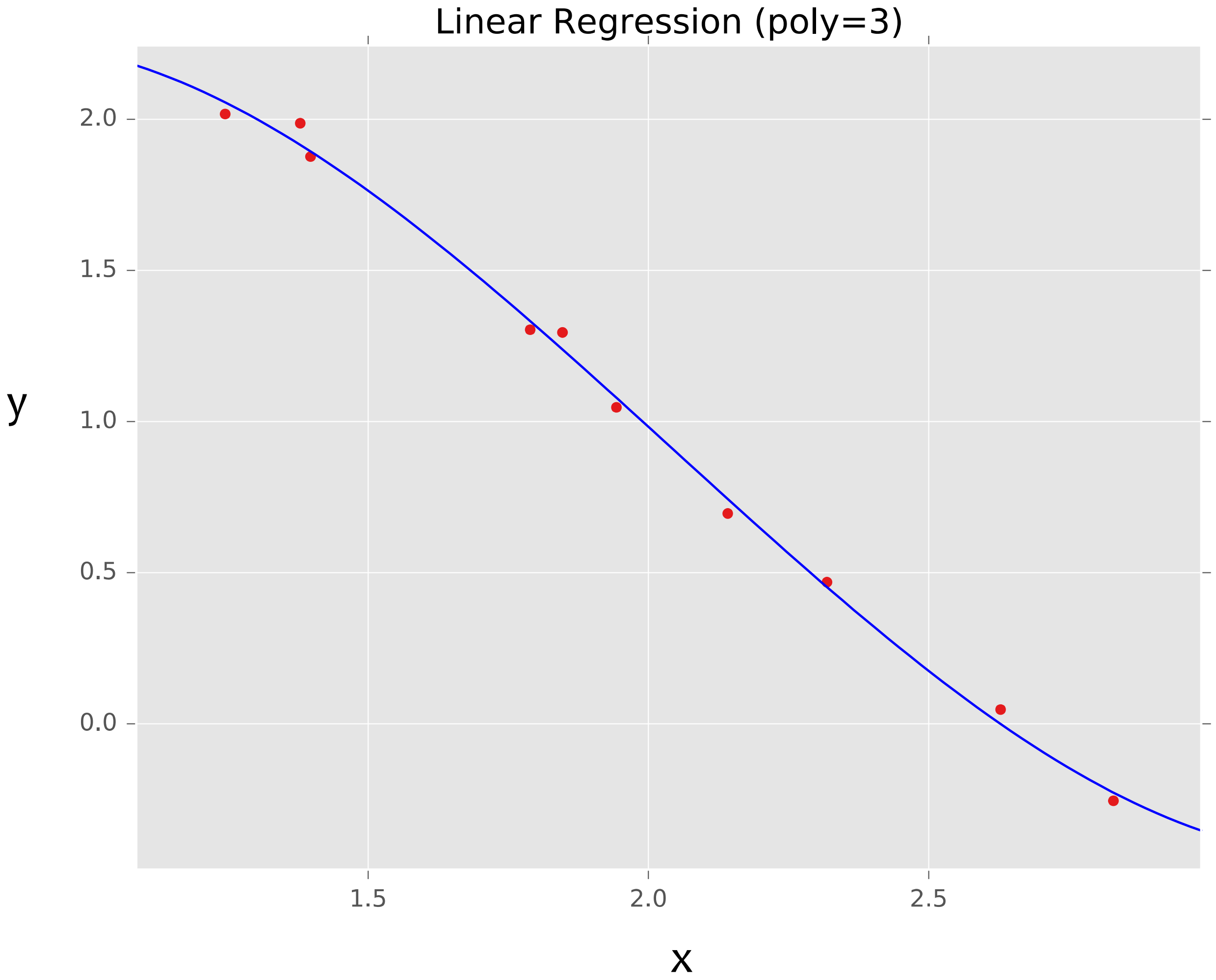
Slide credit: CMU MLD Matt Gormley

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^3
2.0	1.2	$(1.2)^2$	$(1.2)^3$
1.3	1.7	$(1.7)^2$	$(1.7)^3$
0.1	2.7	$(2.7)^2$	$(2.7)^3$
1.1	1.9	$(1.9)^2$	$(1.9)^3$

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



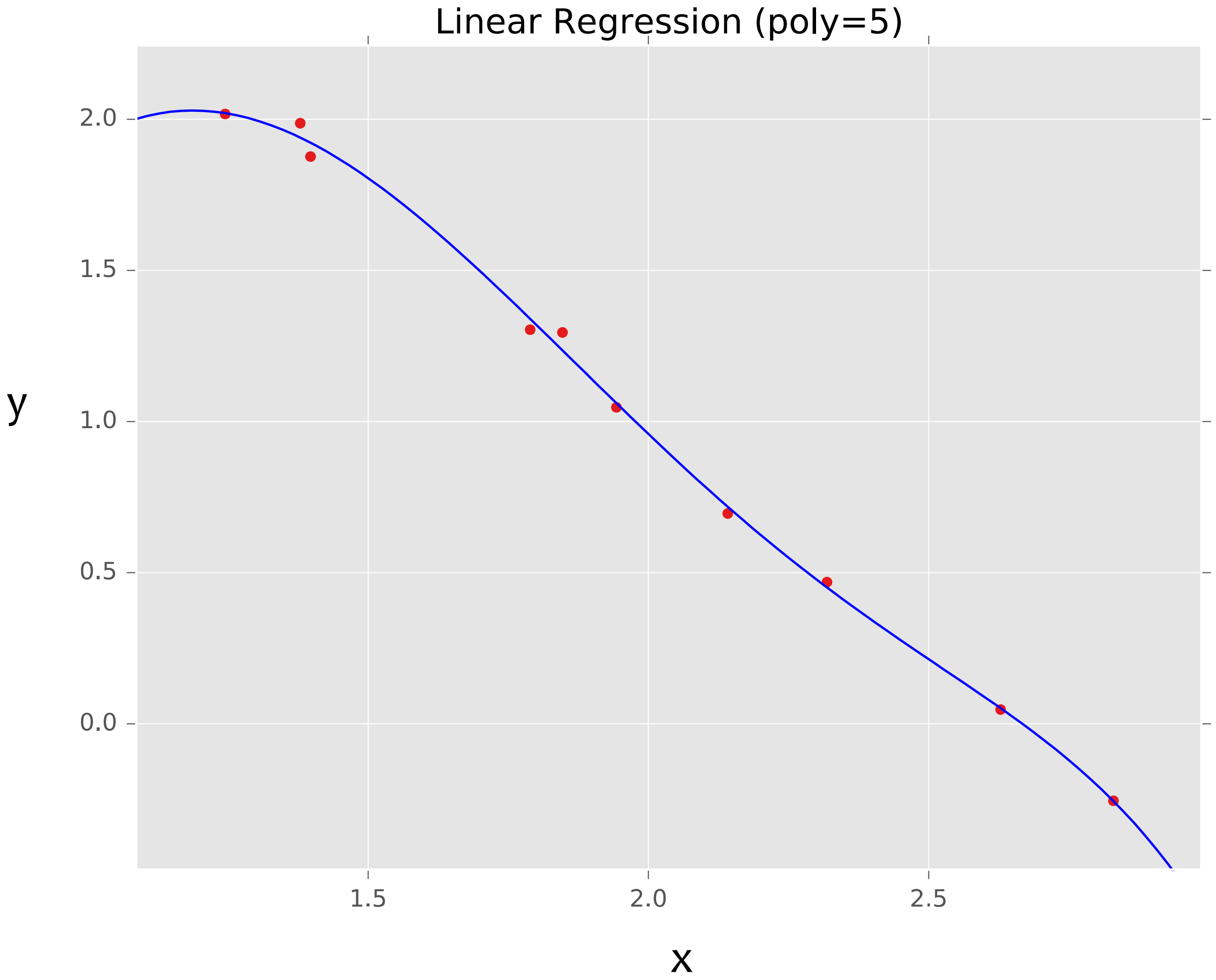
Slide credit: CMU MLD Matt Gormley

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^5
2.0	1.2	$(1.2)^2$	…	$(1.2)^5$
1.3	1.7	$(1.7)^2$	…	$(1.7)^5$
0.1	2.7	$(2.7)^2$	…	$(2.7)^5$
1.1	1.9	$(1.9)^2$	…	$(1.9)^5$

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



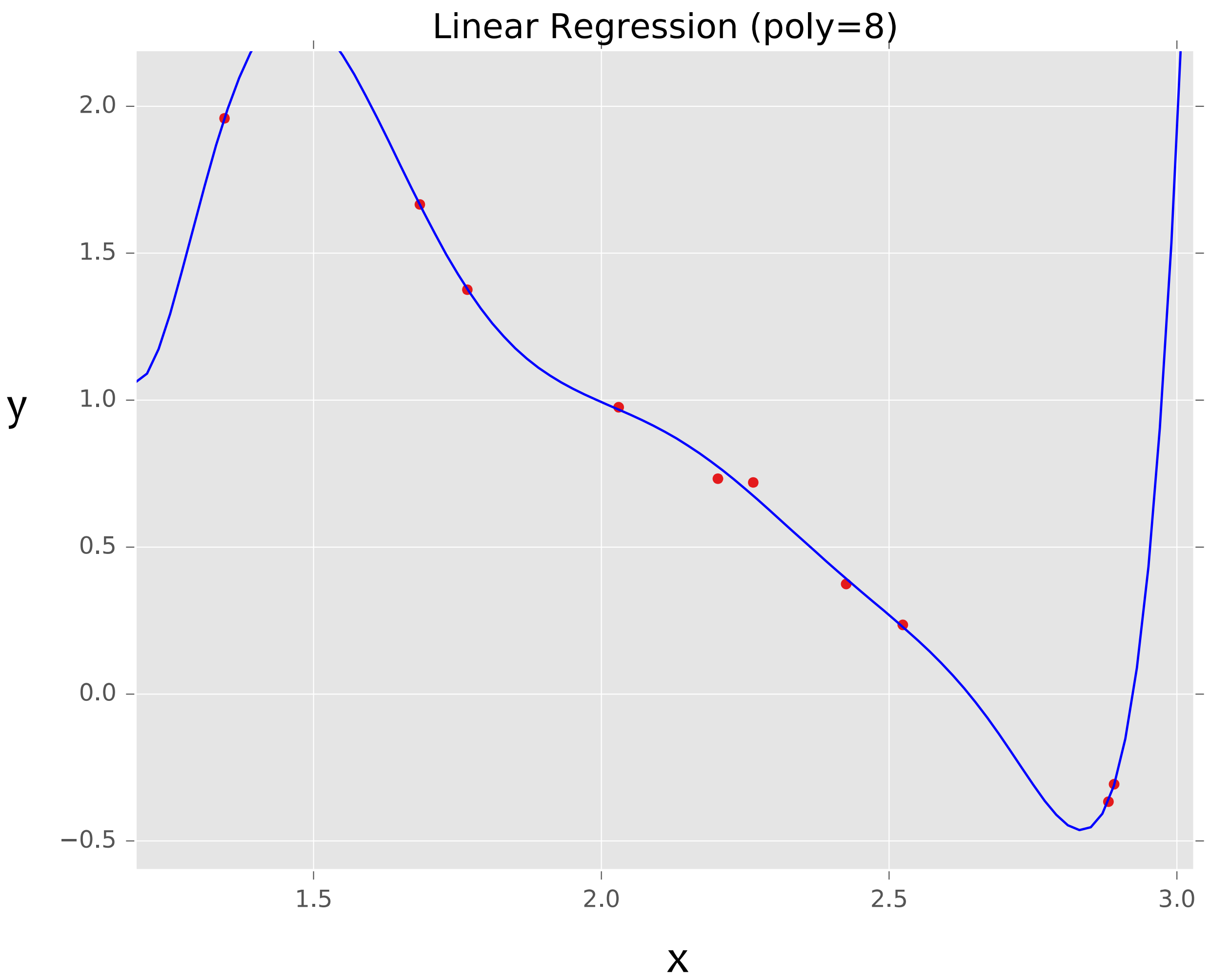
Slide credit: CMU MLD Matt Gormley

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^8
2.0	1.2	$(1.2)^2$	…	$(1.2)^8$
1.3	1.7	$(1.7)^2$	…	$(1.7)^8$
0.1	2.7	$(2.7)^2$	…	$(2.7)^8$
1.1	1.9	$(1.9)^2$	…	$(1.9)^8$

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



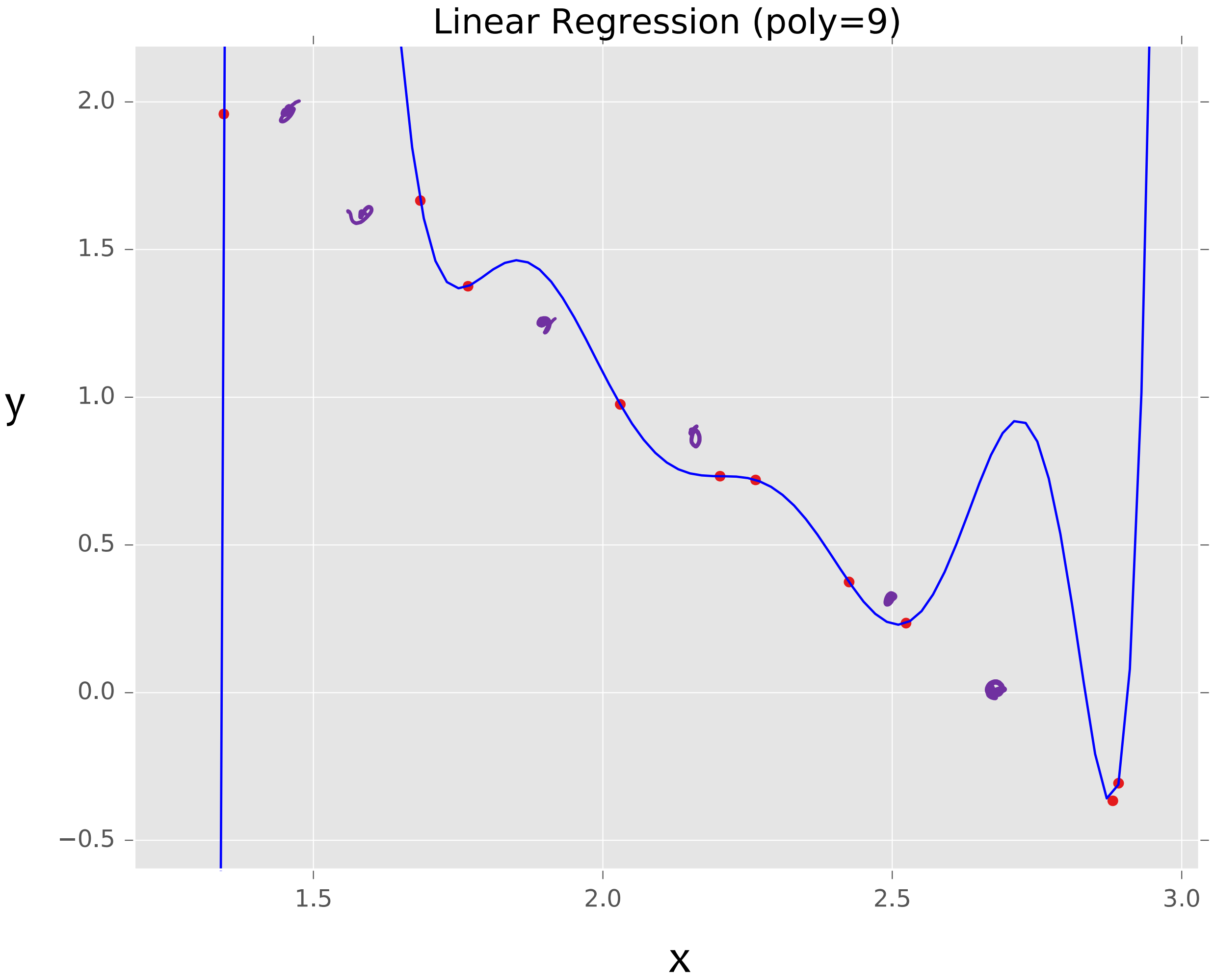
Slide credit: CMU MLD Matt Gormley

Example: Linear Regression

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

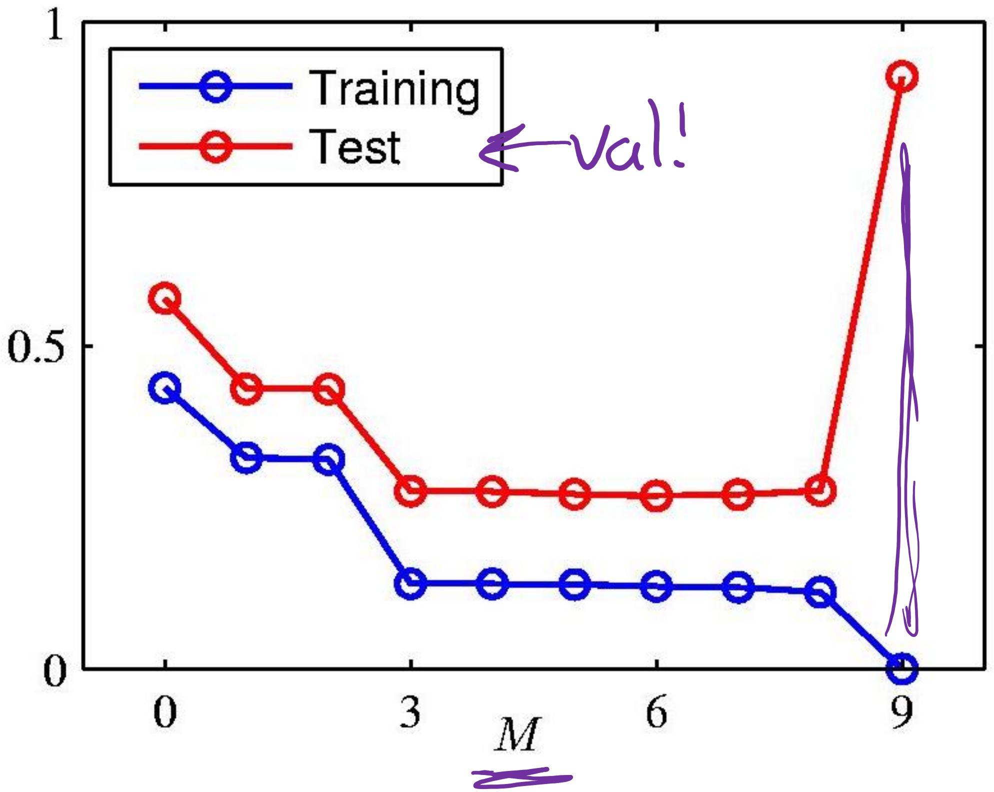
y	x	x^2	…	x^9
2.0	1.2	$(1.2)^2$	…	$(1.2)^9$
1.3	1.7	$(1.7)^2$	…	$(1.7)^9$
0.1	2.7	$(2.7)^2$	…	$(2.7)^9$
1.1	1.9	$(1.9)^2$	…	$(1.9)^9$

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



Slide credit: CMU MLD Matt Gormley

Over-fitting



Slide credit: CMU MLD William Cohen

Polynomial Coefficients

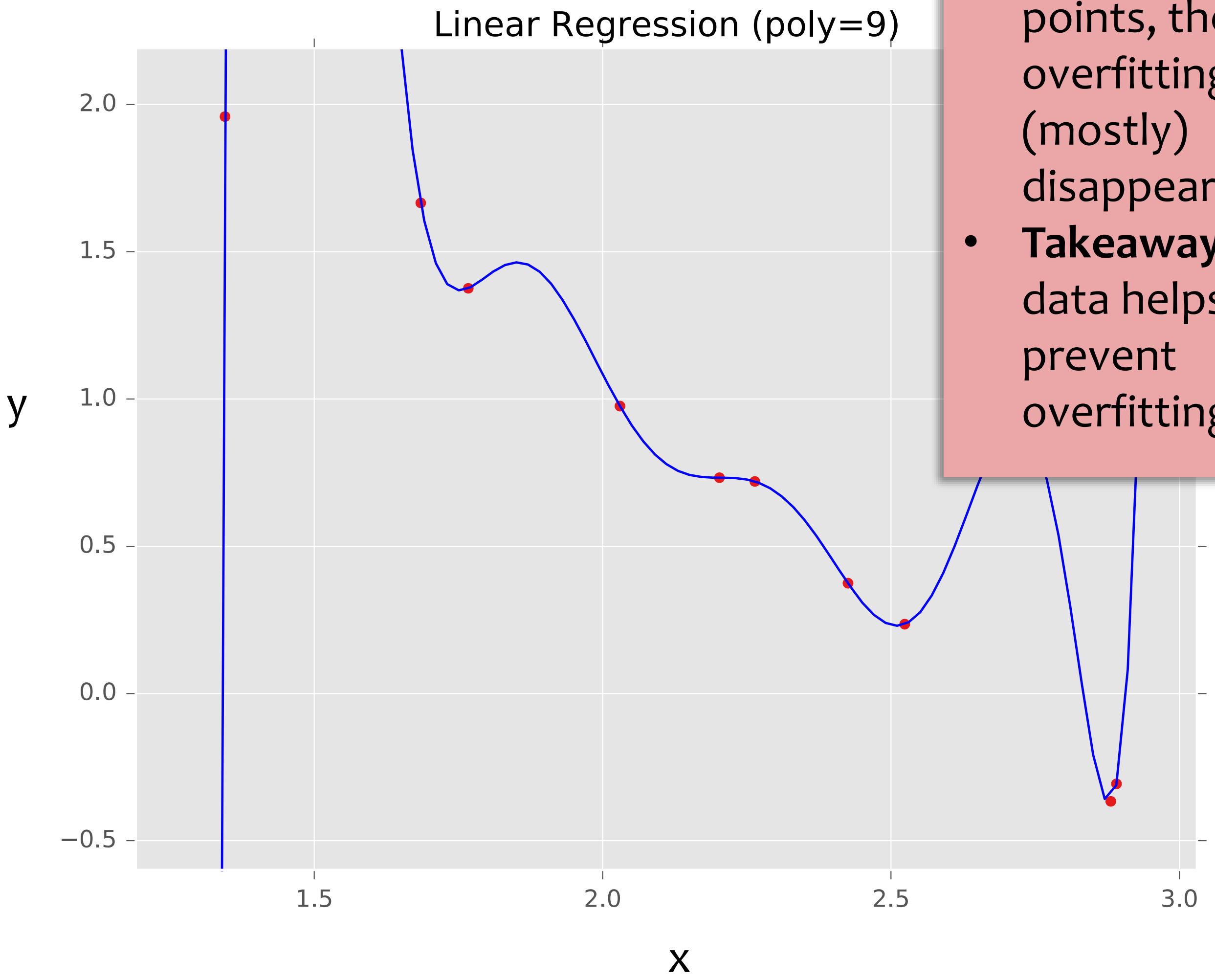
	$M=0$	$M=1$	$M=3$	$M=9$
θ_0	0.19	0.82	0.31	0.35
θ_1		-1.27	7.99	232.37
θ_2			-25.43	-5321.83
θ_3			17.37	48568.31
θ_4				-231639.30
θ_5				640042.26
θ_6				-1061800.52
θ_7				1042400.18
θ_8				-557682.99
θ_9				125201.43

Slide credit: CMU MLD William Cohen

Example: Linear Regression

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

i	y	x	…	x^9
1	2.0	1.2	…	$(1.2)^9$
2	1.3	1.7	…	$(1.7)^9$
…	…	…	…	…
10	1.1	1.9	…	$(1.9)^9$



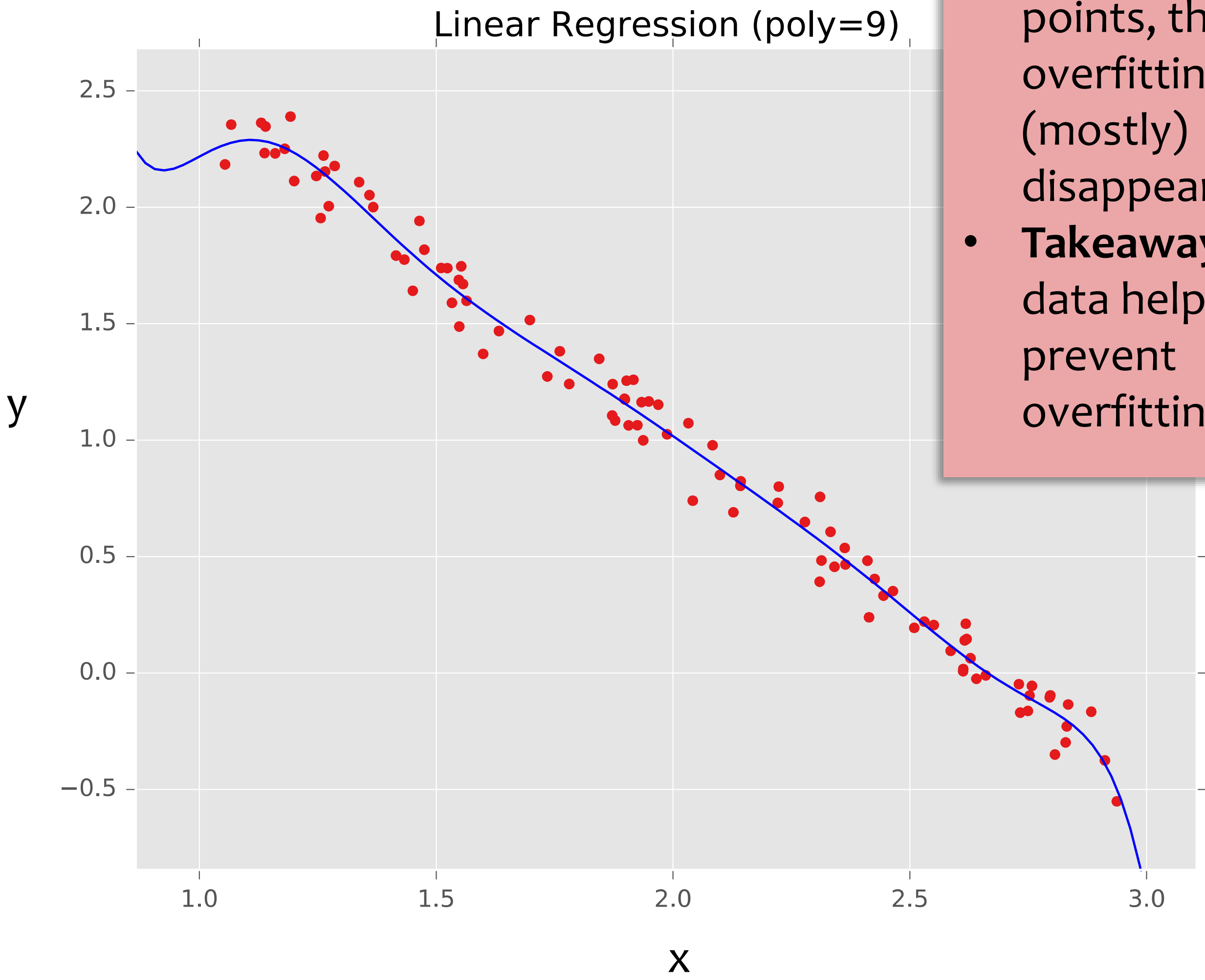
- With just N = 10 points we overfit!
- But with N = 100 points, the overfitting (mostly) disappears
- **Takeaway:** more data helps prevent overfitting

Slide credit: CMU MLD Matt Gormley

Example: Linear Regression

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

i	y	x	…	x^9
1	2.0	1.2	…	$(1.2)^9$
2	1.3	1.7	…	$(1.7)^9$
3	0.1	2.7	…	$(2.7)^9$
4	1.1	1.9	…	$(1.9)^9$
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…
98	…	…	…	…
99	…	…	…	…
100	0.9	1.5	…	$(1.5)^9$



- With just N = 10 points we overfit!
- But with N = 100 points, the overfitting (mostly) disappears
- **Takeaway:** more data helps prevent overfitting

Slide credit: CMU MLD Matt Gormley

REGULARIZATION

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:

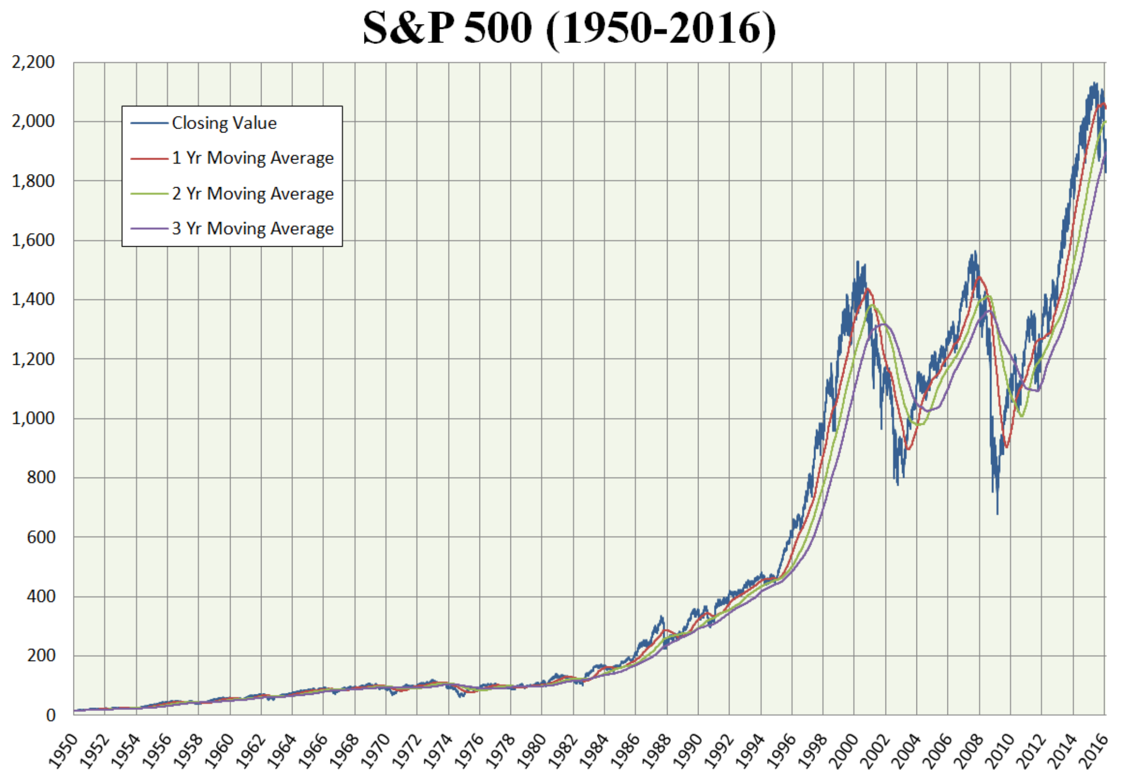
– Decision Trees (e.g. when tree is too deep)
– KNN (e.g. when k is small)
– Perceptron (e.g. when sample isn't representative)
– Linear Regression (e.g. with nonlinear features)
– Logistic Regression (e.g. with many rare features)

Slide credit: CMU MLD Matt Gormley

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?



Motivation: Regularization

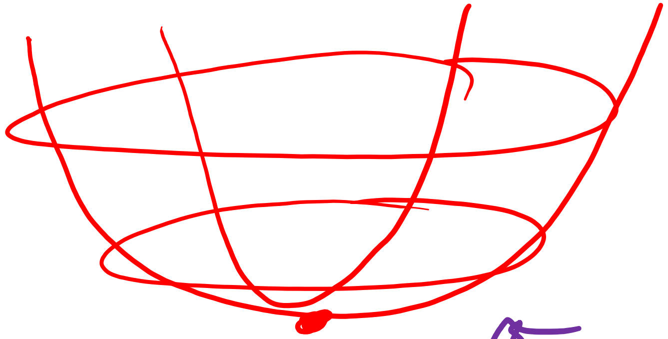
Occam's Razor: prefer the simplest hypothesis

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**)

Regularization

$\vec{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$



Given objective function: $J(\theta)$

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

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

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

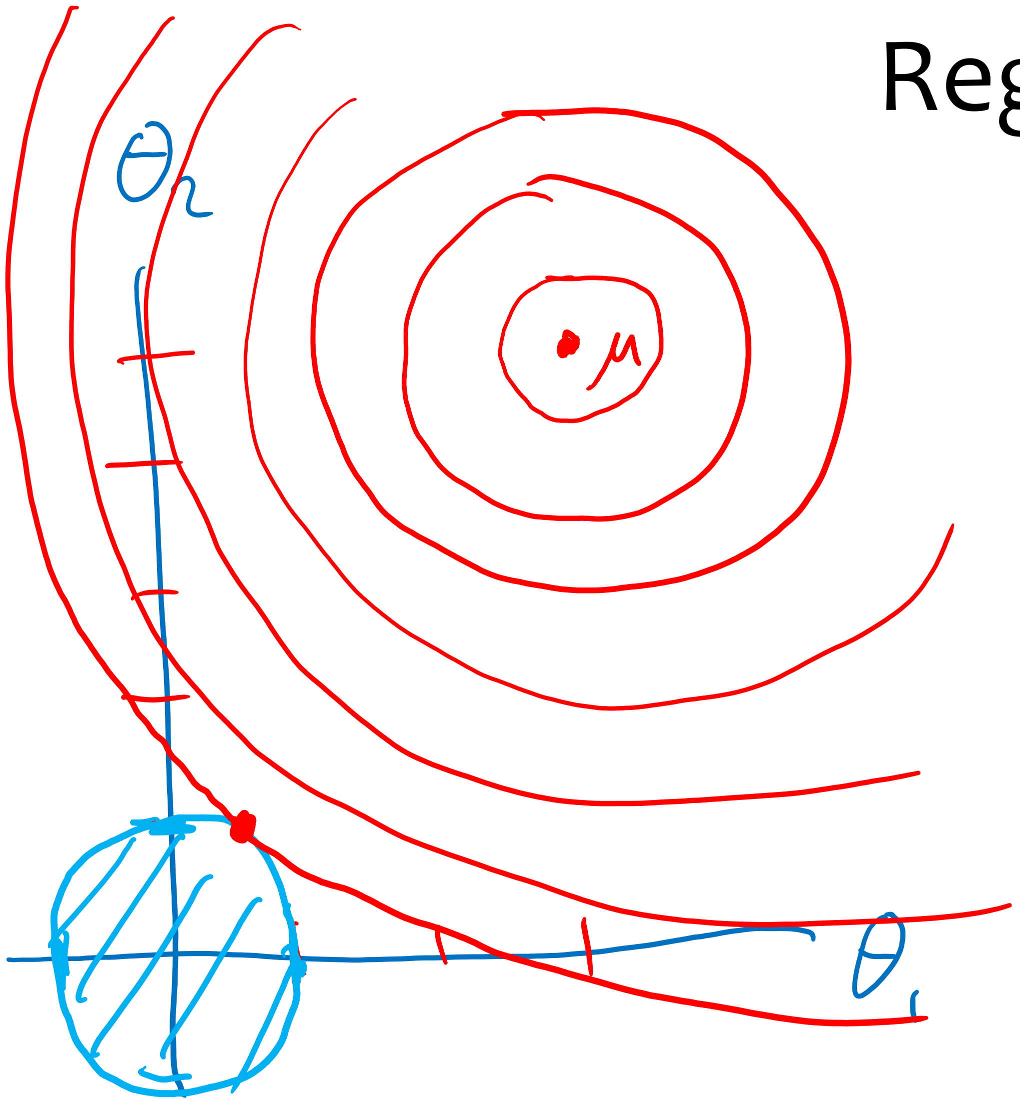
– Example: q-norm (usually p-norm) $r(\boldsymbol{\theta}) = ||\boldsymbol{\theta}||_q = \left[\sum_{m=1}^{M} ||\theta_m||^q \right]^{(\frac{1}{q})}$

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

num nonzeros

Slide credit: CMU MLD Matt Gormley

Regularization



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

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

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

Piazza Poll 1

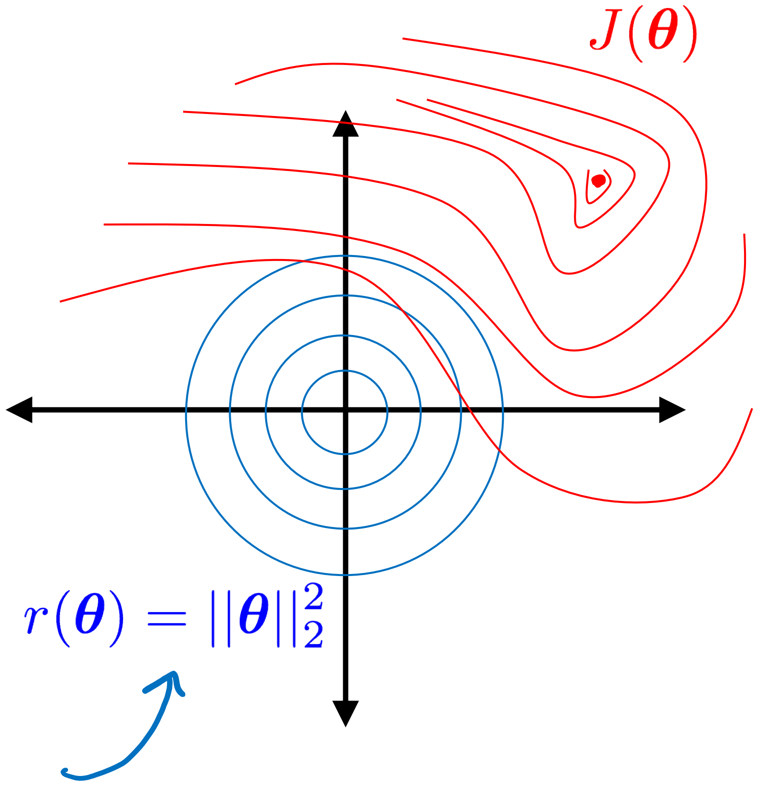
Question:

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 weight (cartoon)
2. Plot test error vs . regularization weight (cartoon)

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

Slide credit: CMU MLD Matt Gormley

Piazza Poll 2

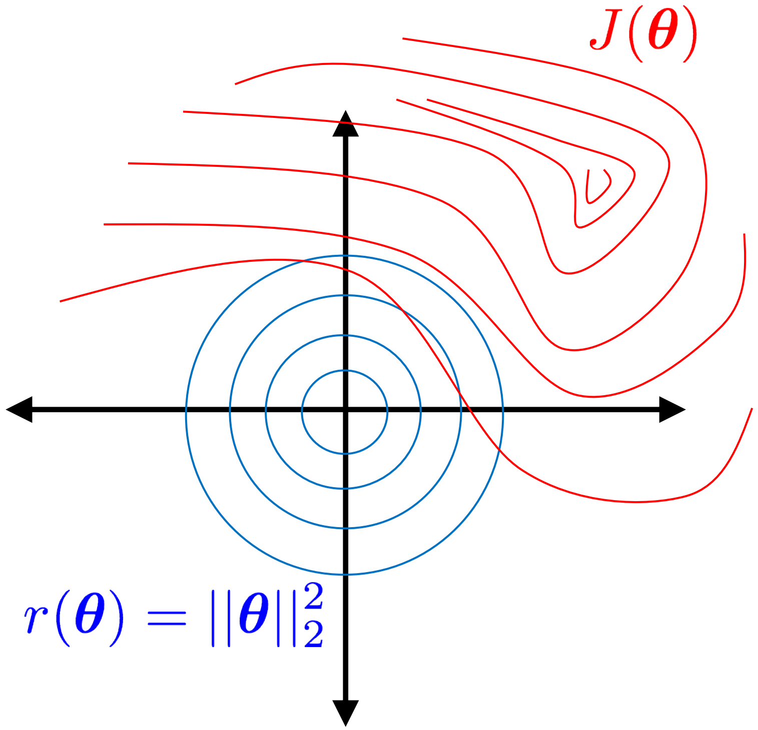
Question:

Suppose we are minimizing J'(θ) where

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

As we increase λ from 0, the 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

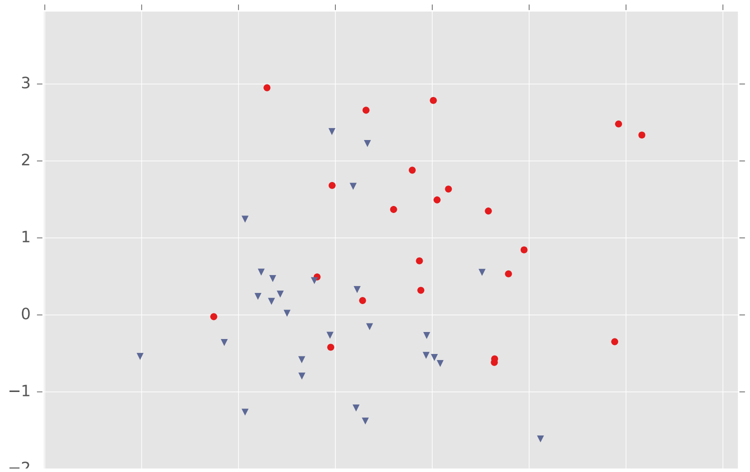
Whitening Data

- 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)

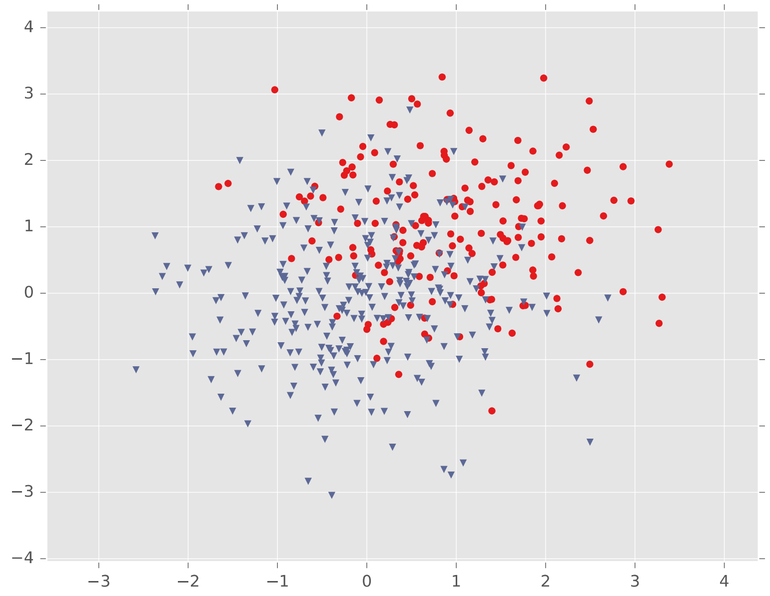
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

Training Data



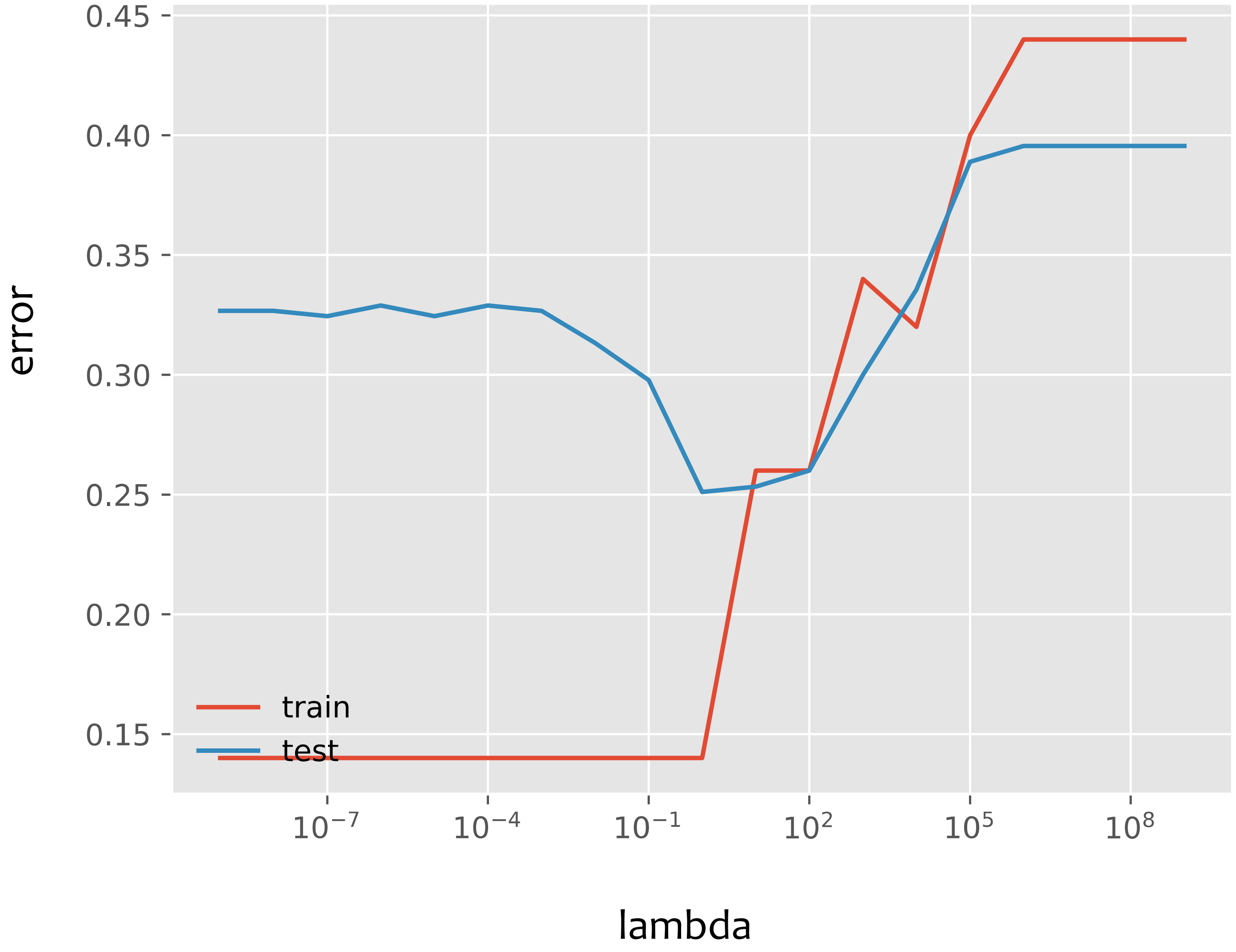
Test Data



- 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)

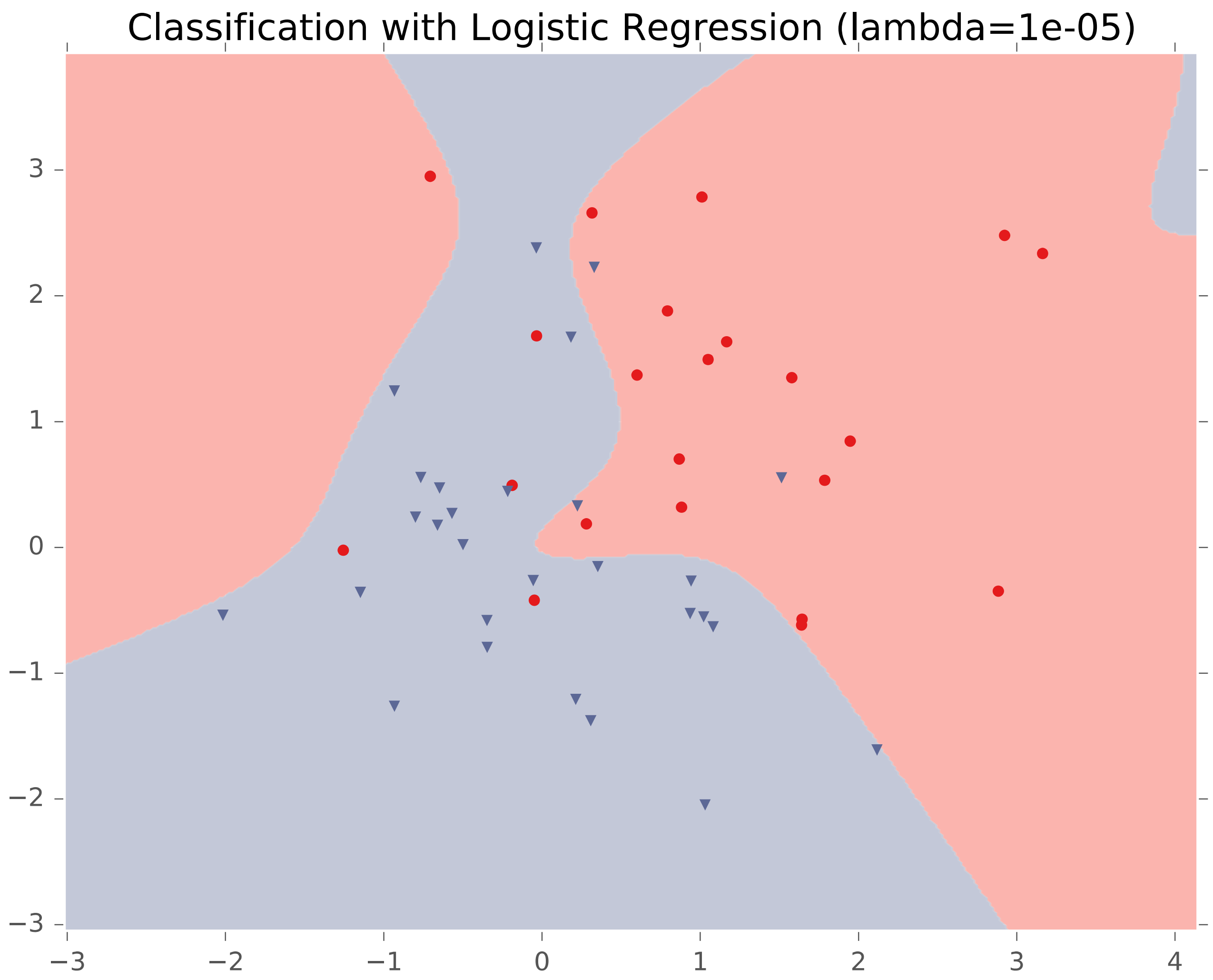
Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression

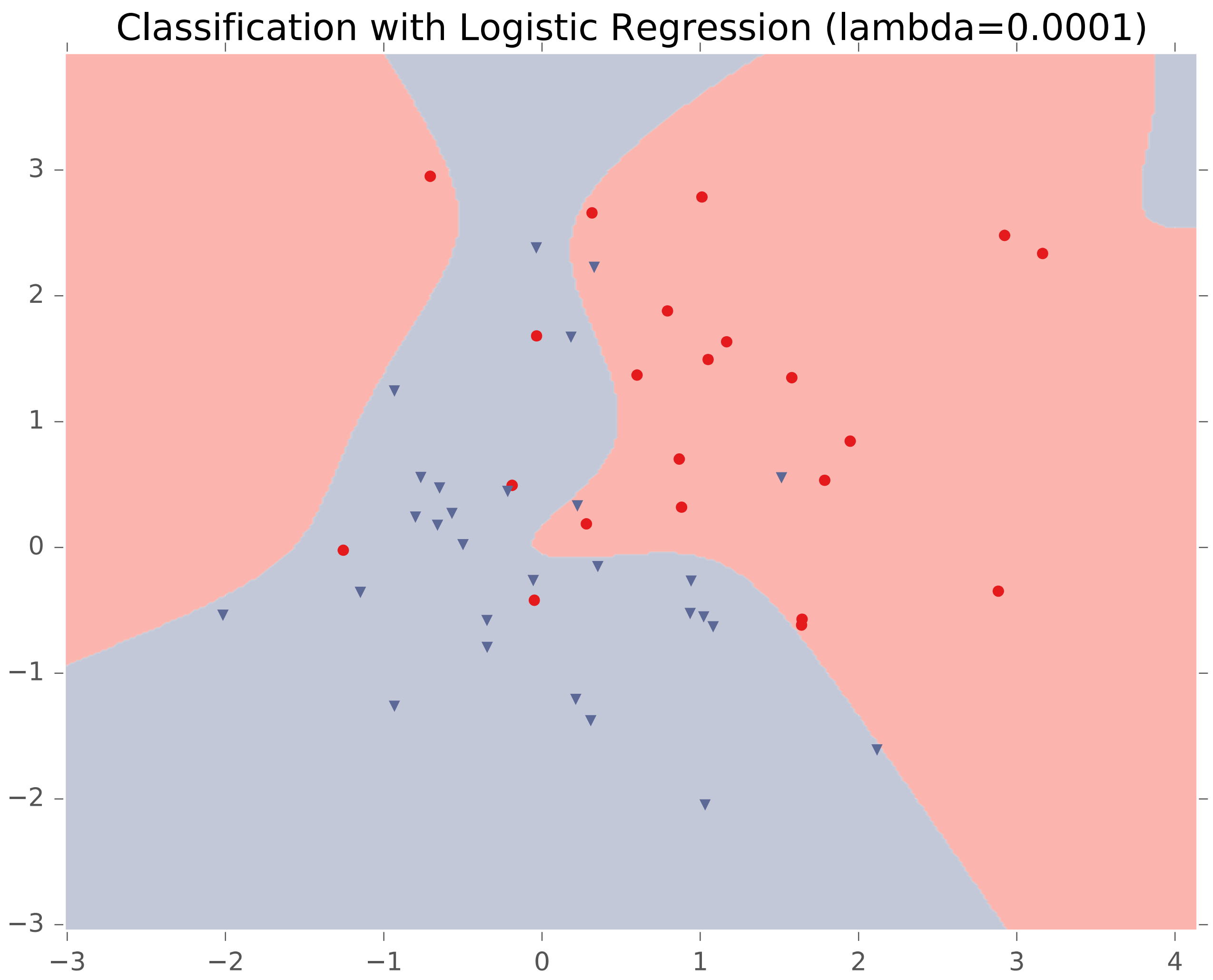


Slide credit: CMU MLD Matt Gormley

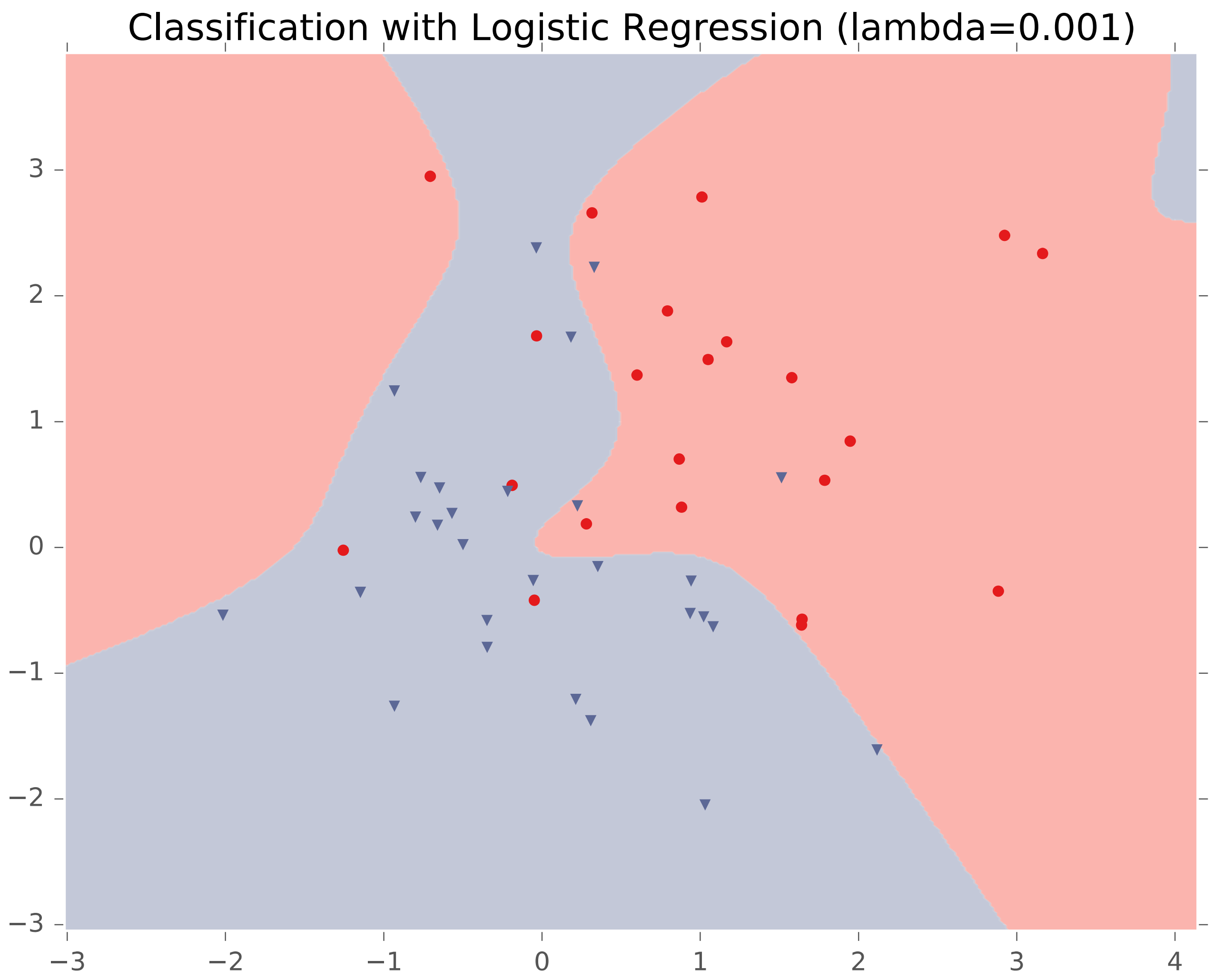
Example: Logistic Regression



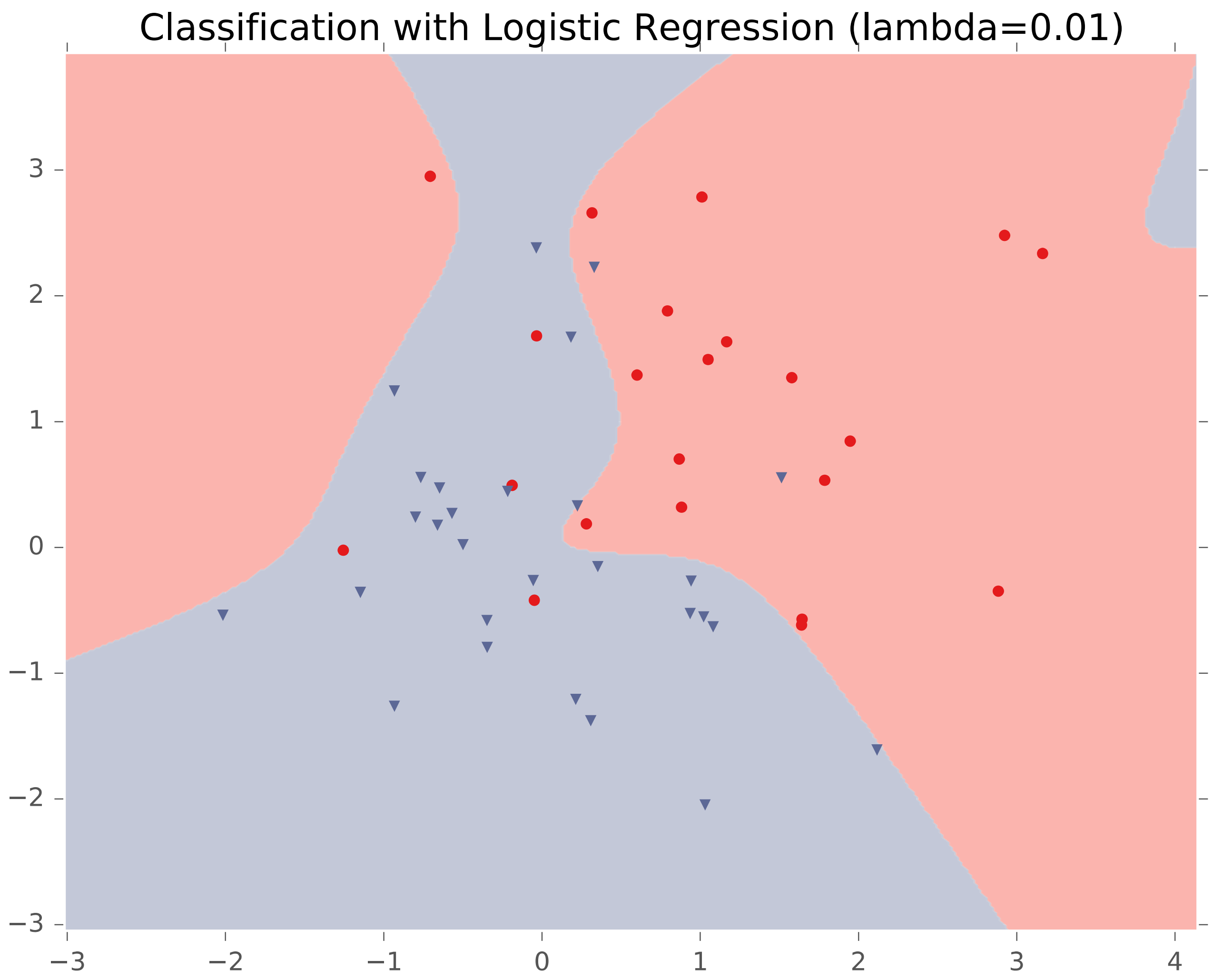
Example: Logistic Regression



Example: Logistic Regression

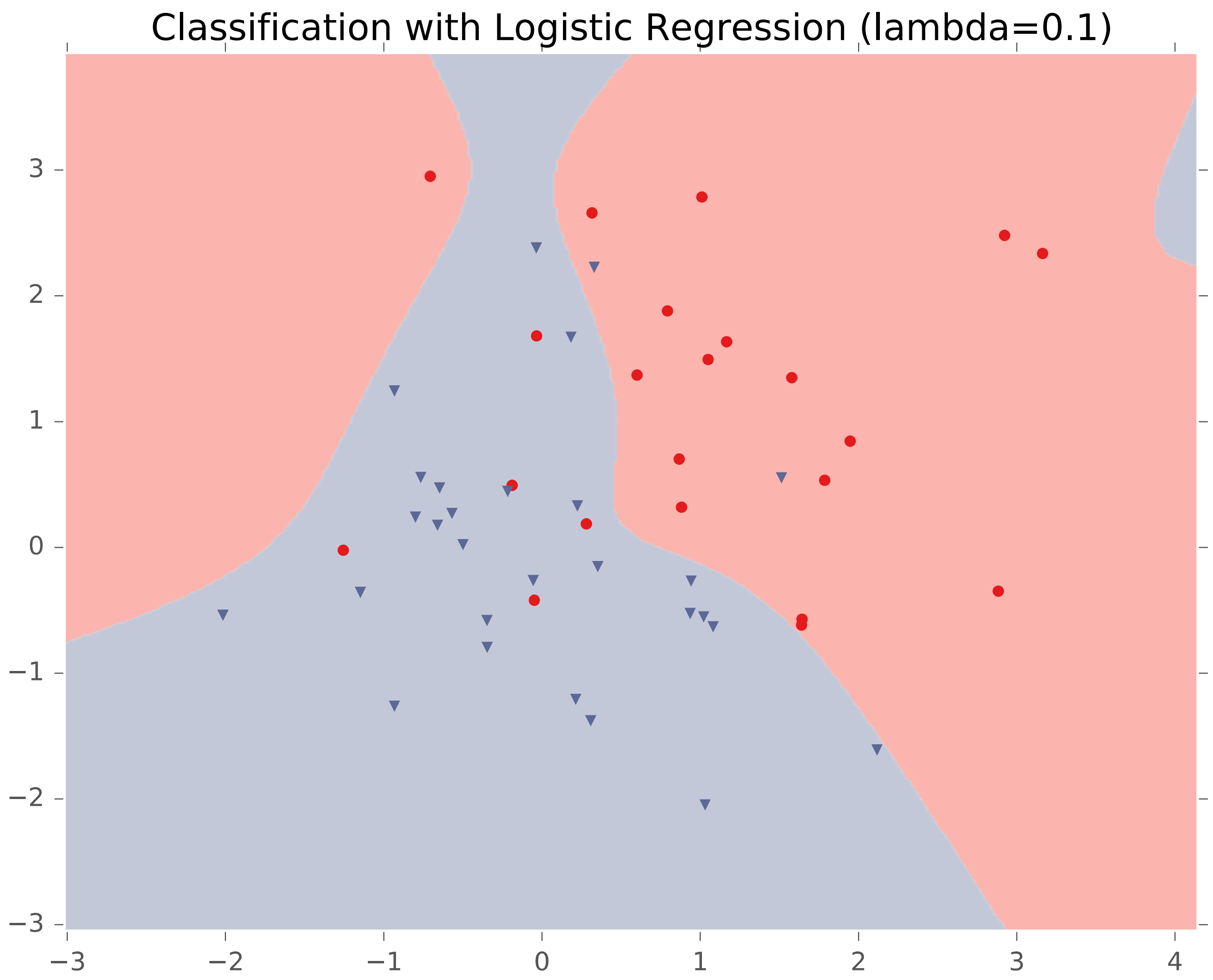


Example: Logistic Regression

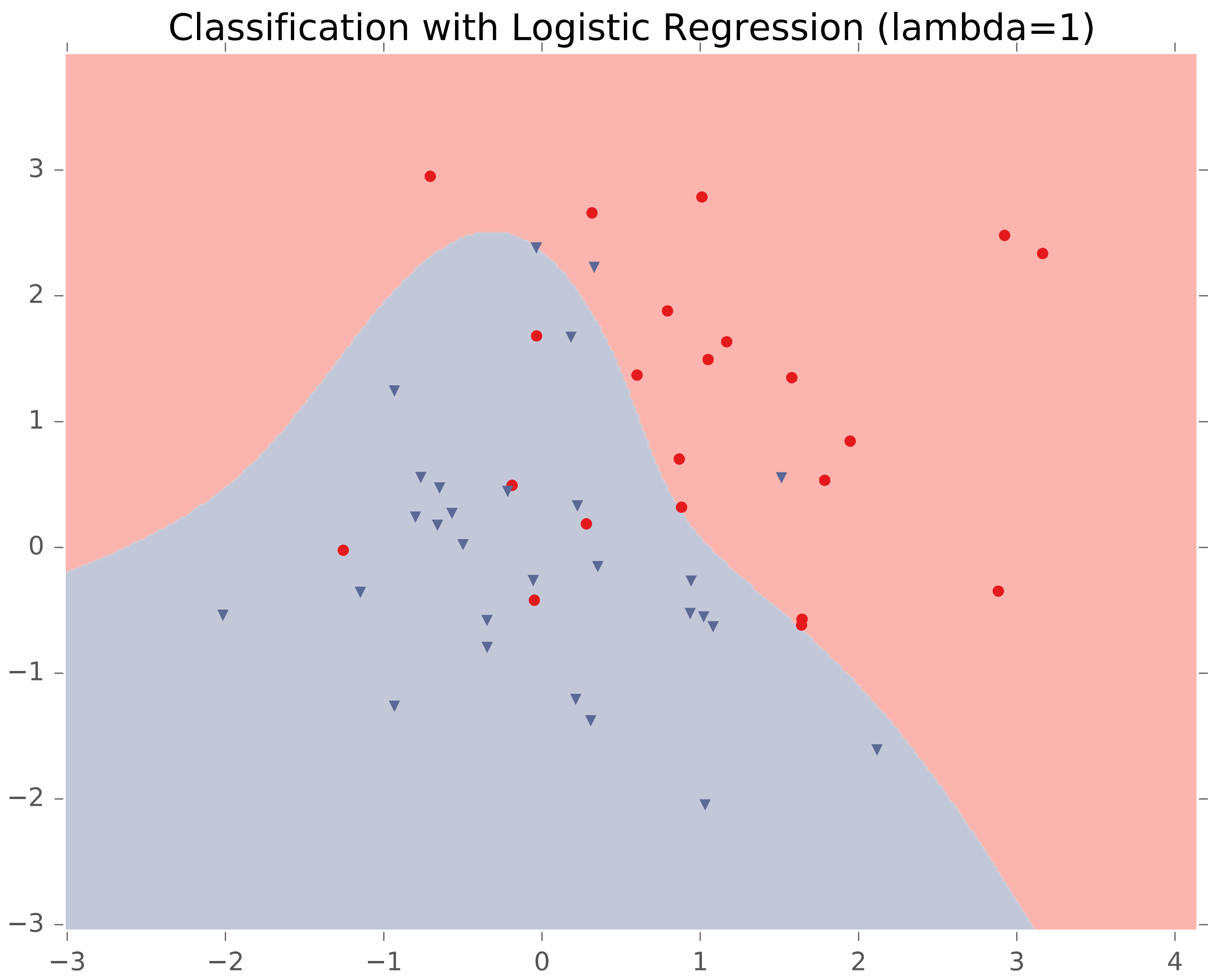


Slide credit: CMU MLD Matt Gormley

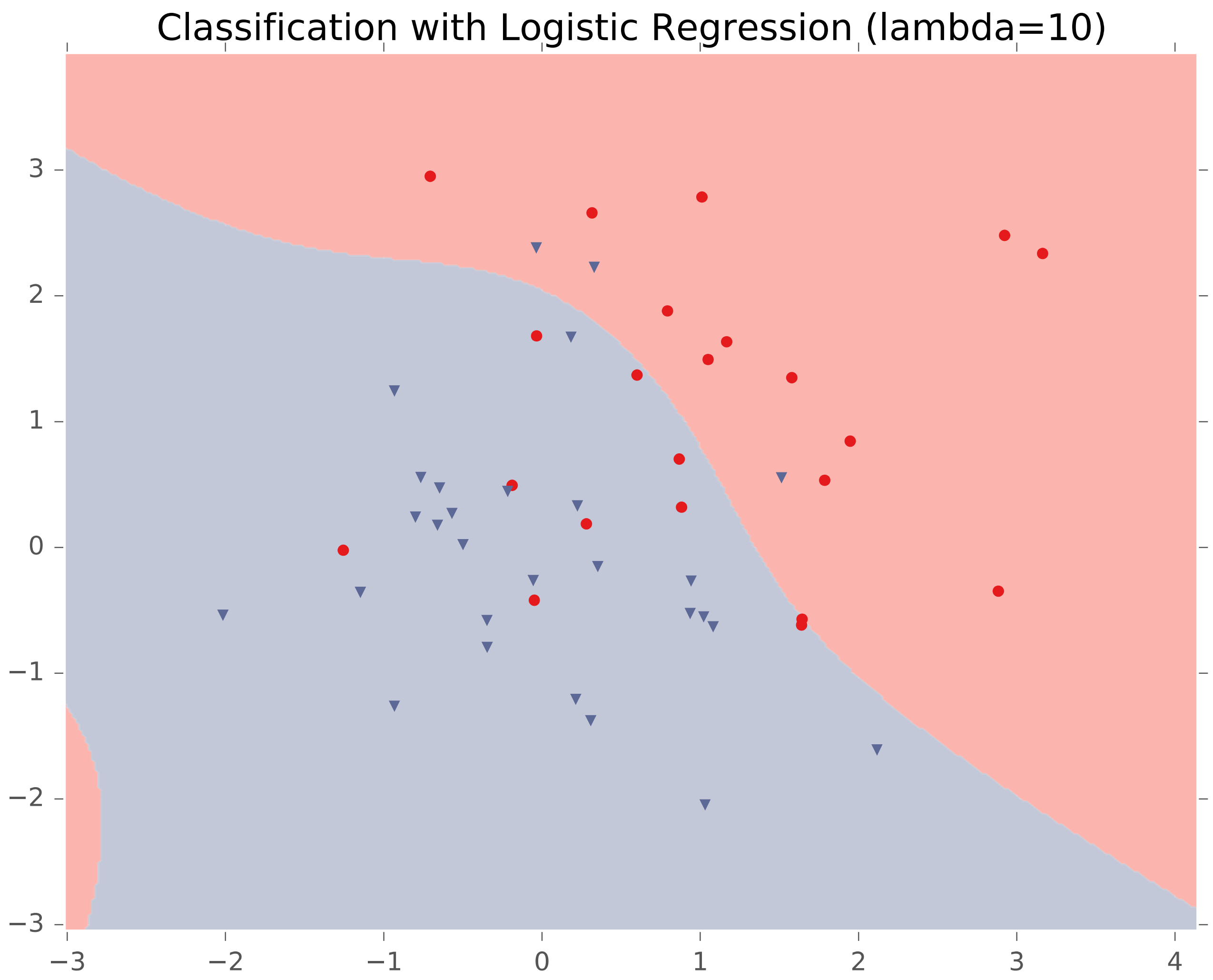
Example: Logistic Regression



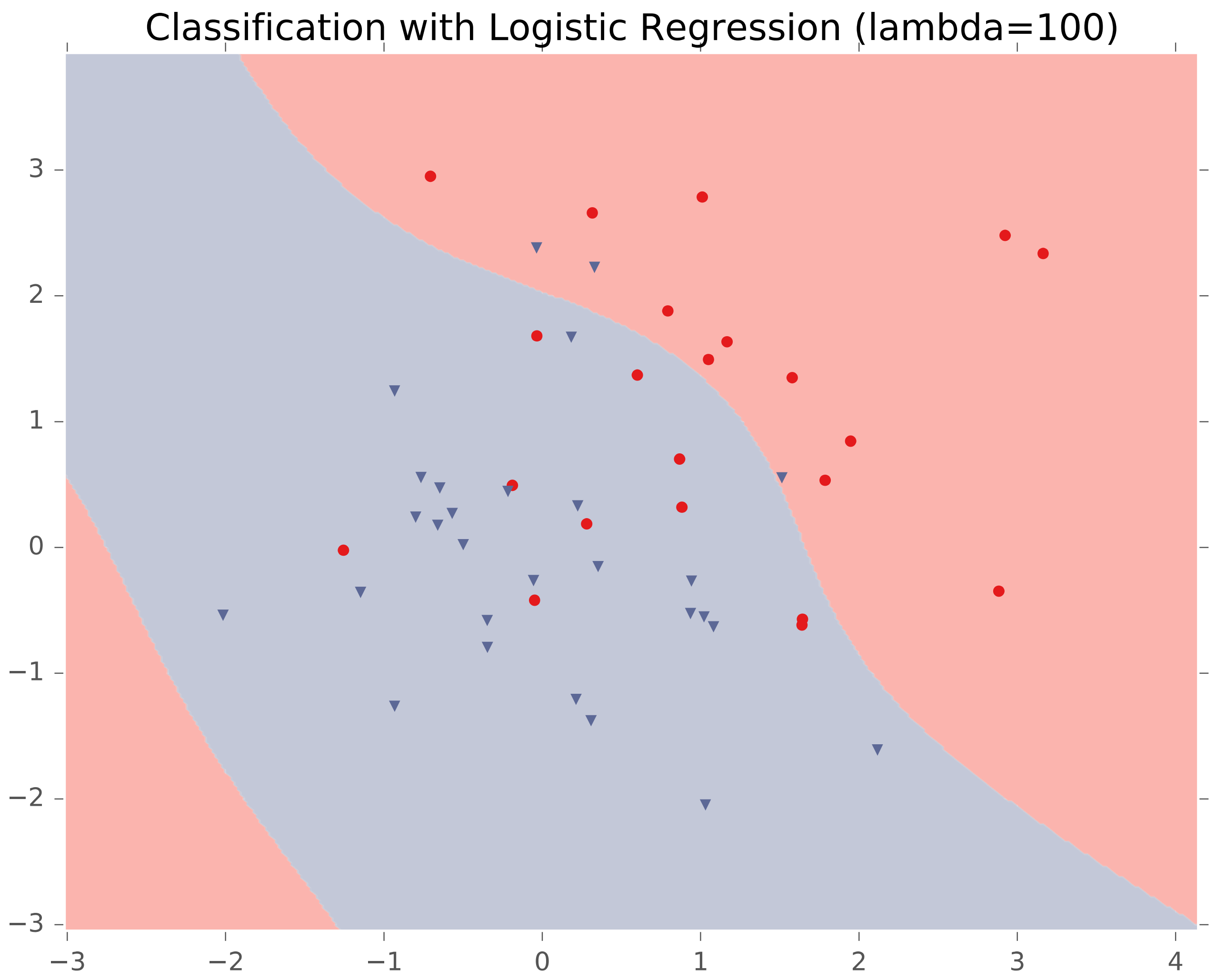
Example: Logistic Regression



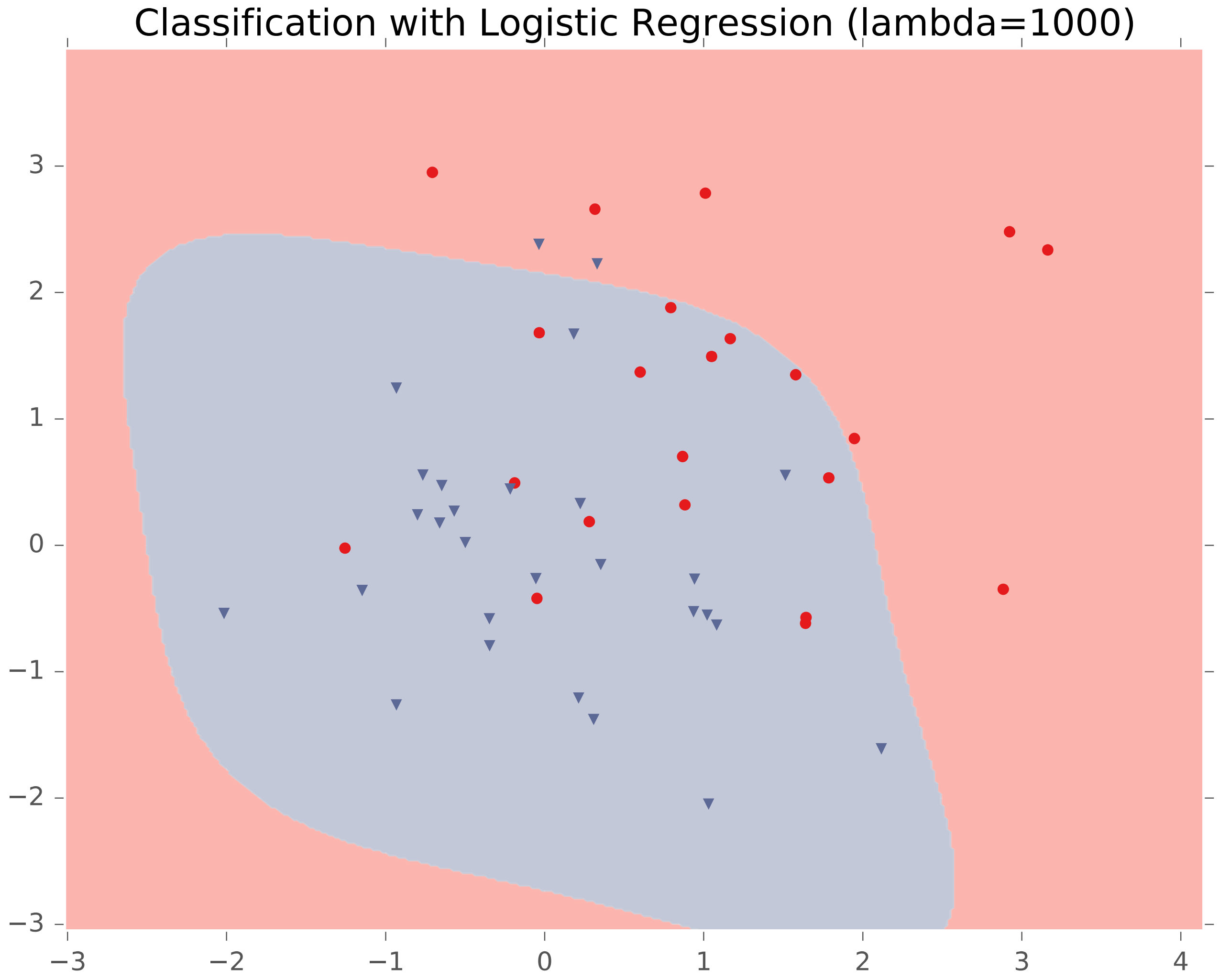
Example: Logistic Regression



Example: Logistic Regression

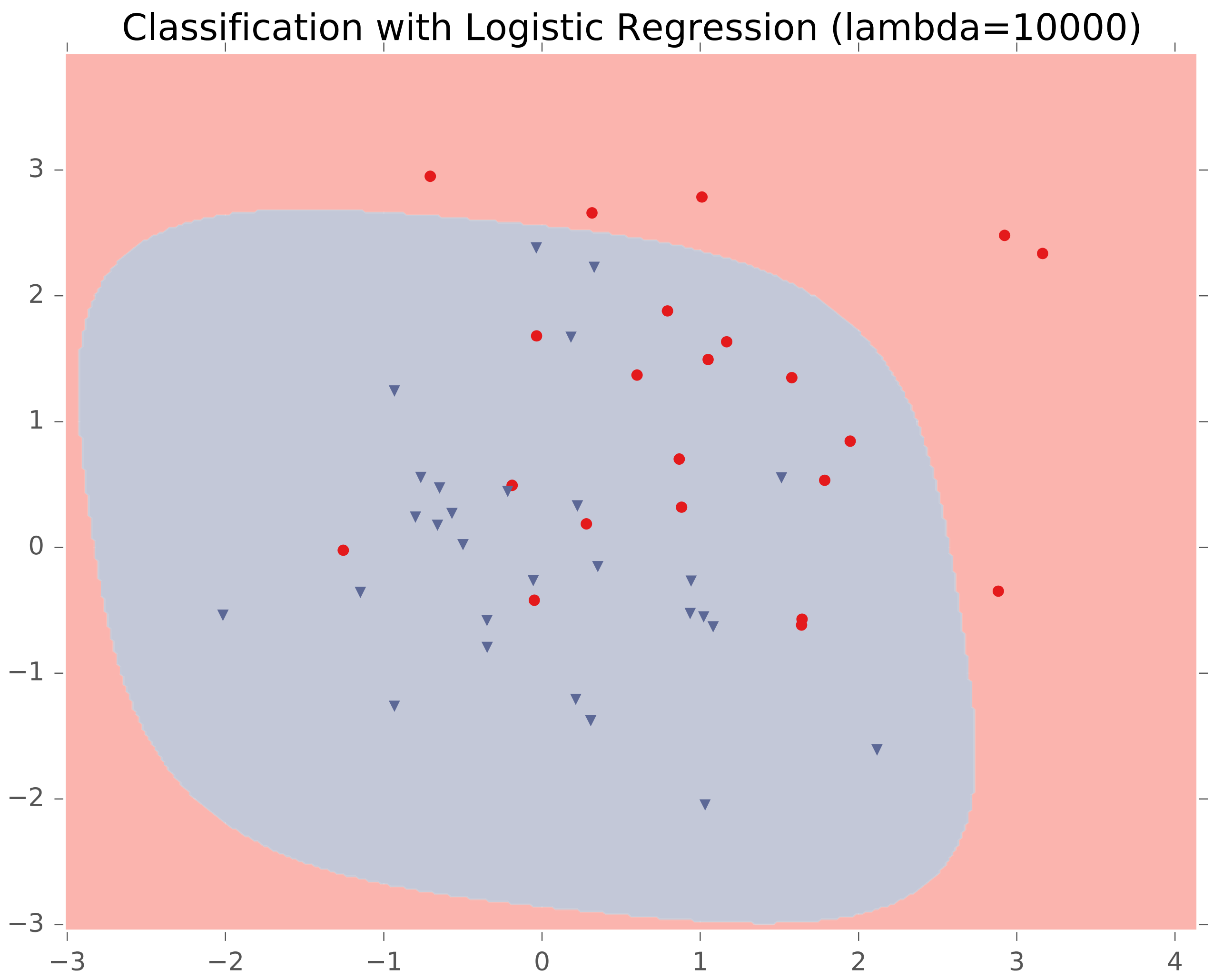


Example: Logistic Regression

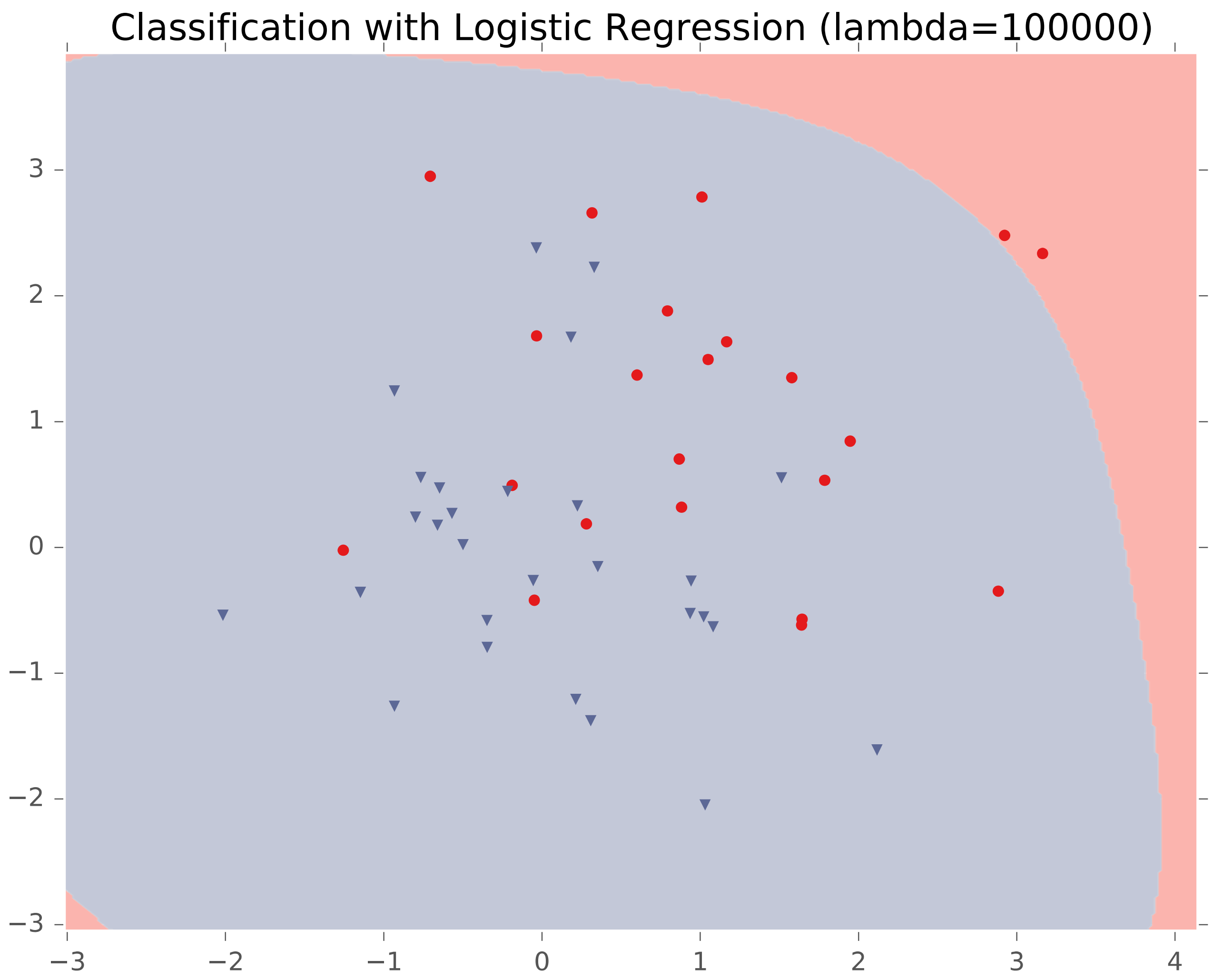


Slide credit: CMU MLD Matt Gormley

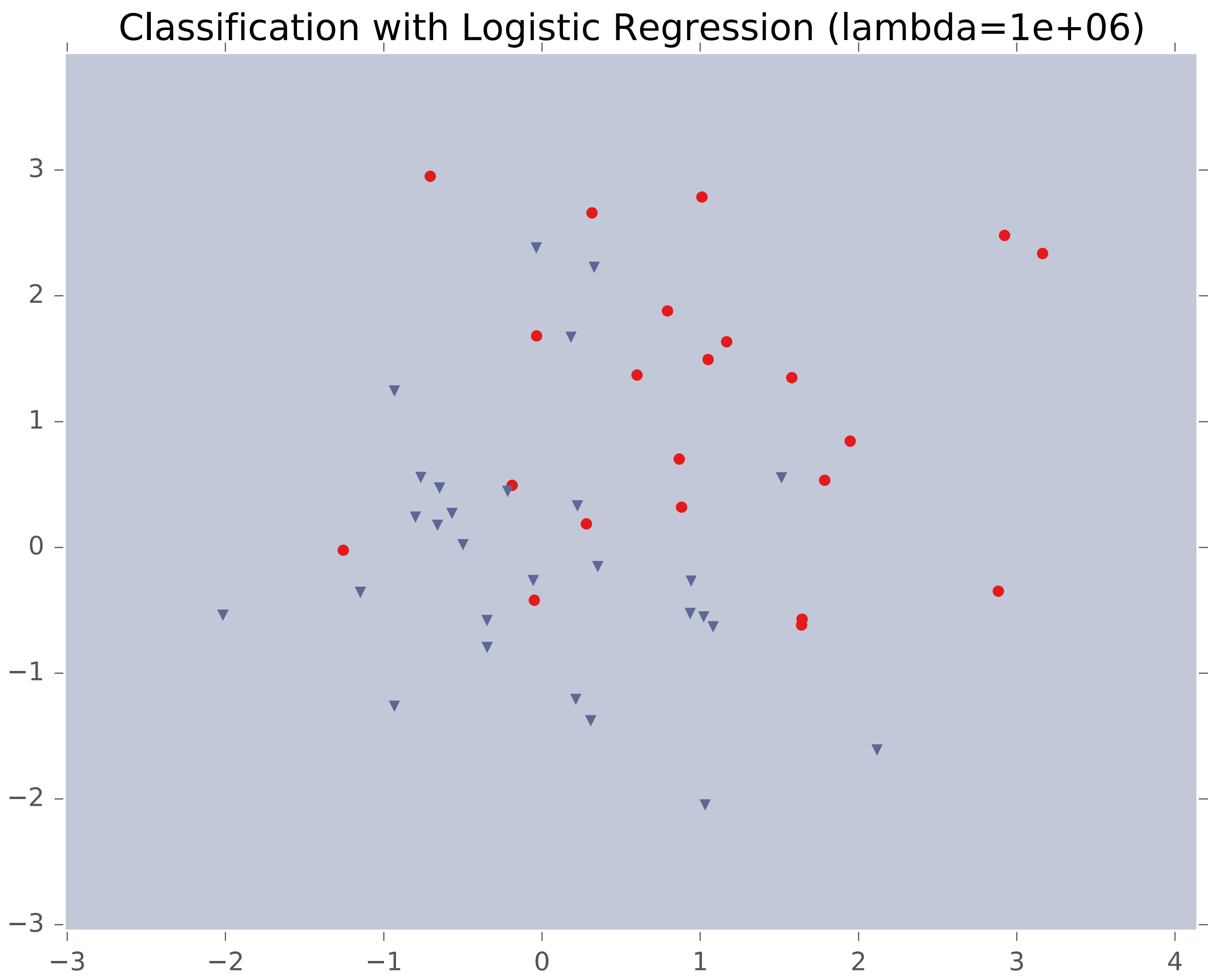
Example: Logistic Regression



Example: Logistic Regression

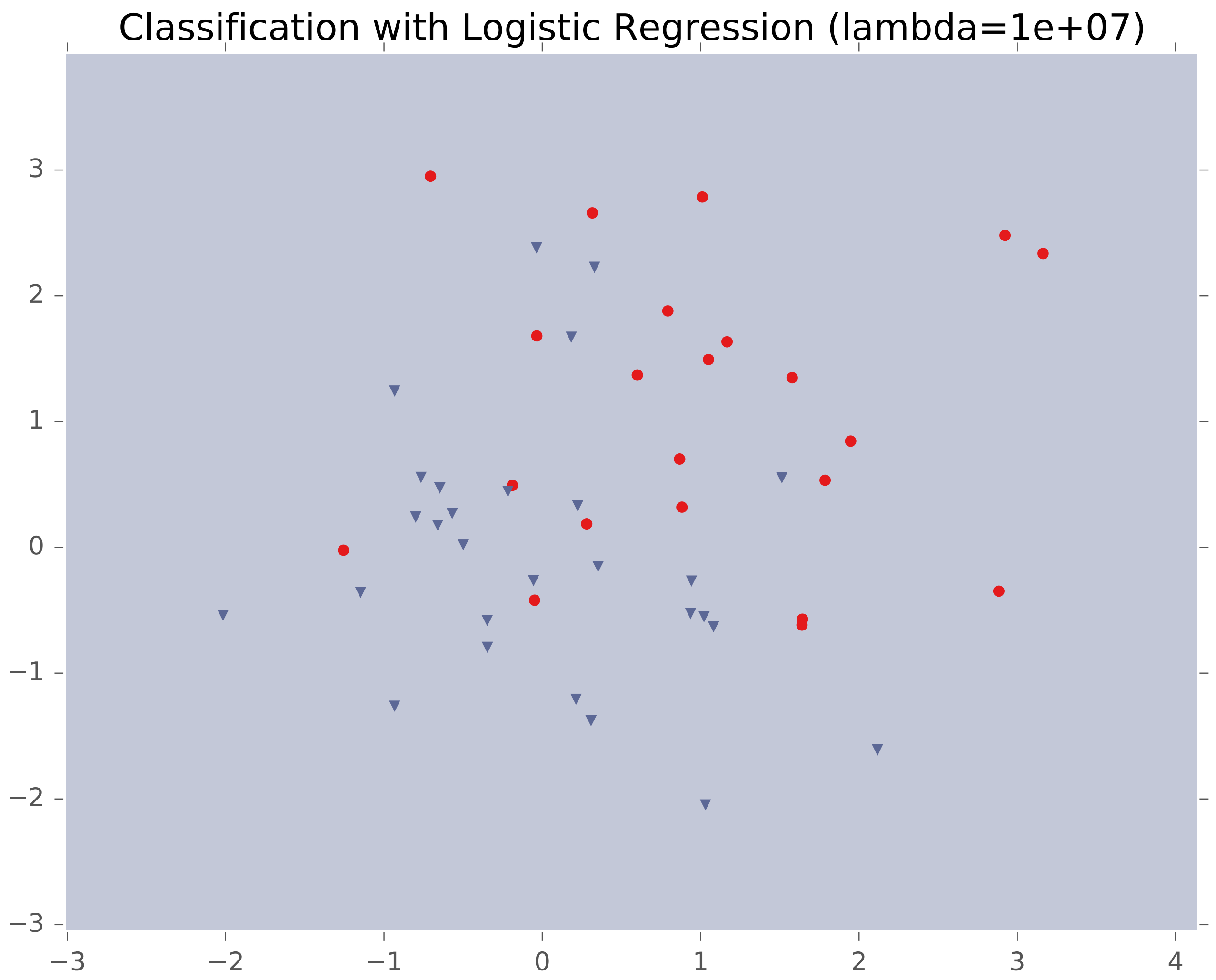


Example: Logistic Regression

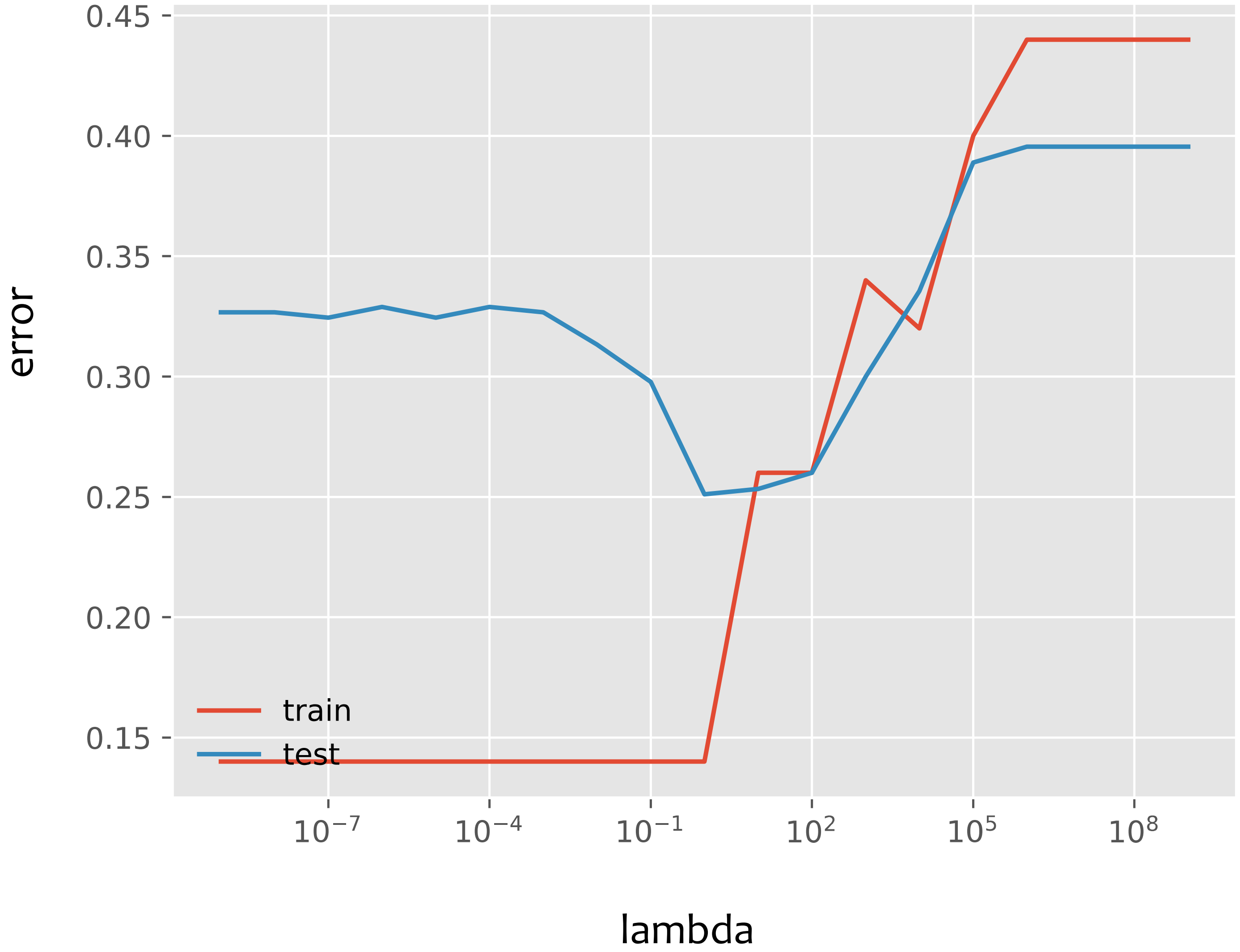


Slide credit: CMU MLD Matt Gormley

Example: Logistic Regression



Example: Logistic Regression



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

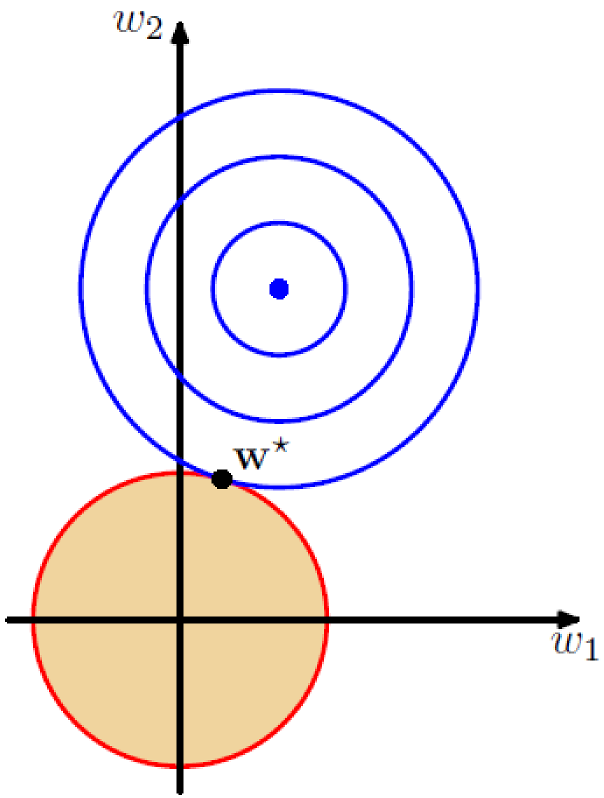
q	$r(\boldsymbol{\theta})$	**yields parameters that are...**	**name**	**optimization notes**
0	$\lVert\boldsymbol{\theta}\rVert_0 = \sum \mathbb{1}(\theta_m \neq 0)$	zero values	L0 reg.	no good computational solutions
1	$\lVert\boldsymbol{\theta}\rVert_1 = \sum \lvert\theta_m\rvert$	zero values	L1 reg.	subdifferentiable
2	$(\lVert\boldsymbol{\theta}\rVert_2)^2 = \sum \theta_m^2$	small values	L2 reg.	differentiable

Slide credit: CMU MLD Matt Gormley

Regularization

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

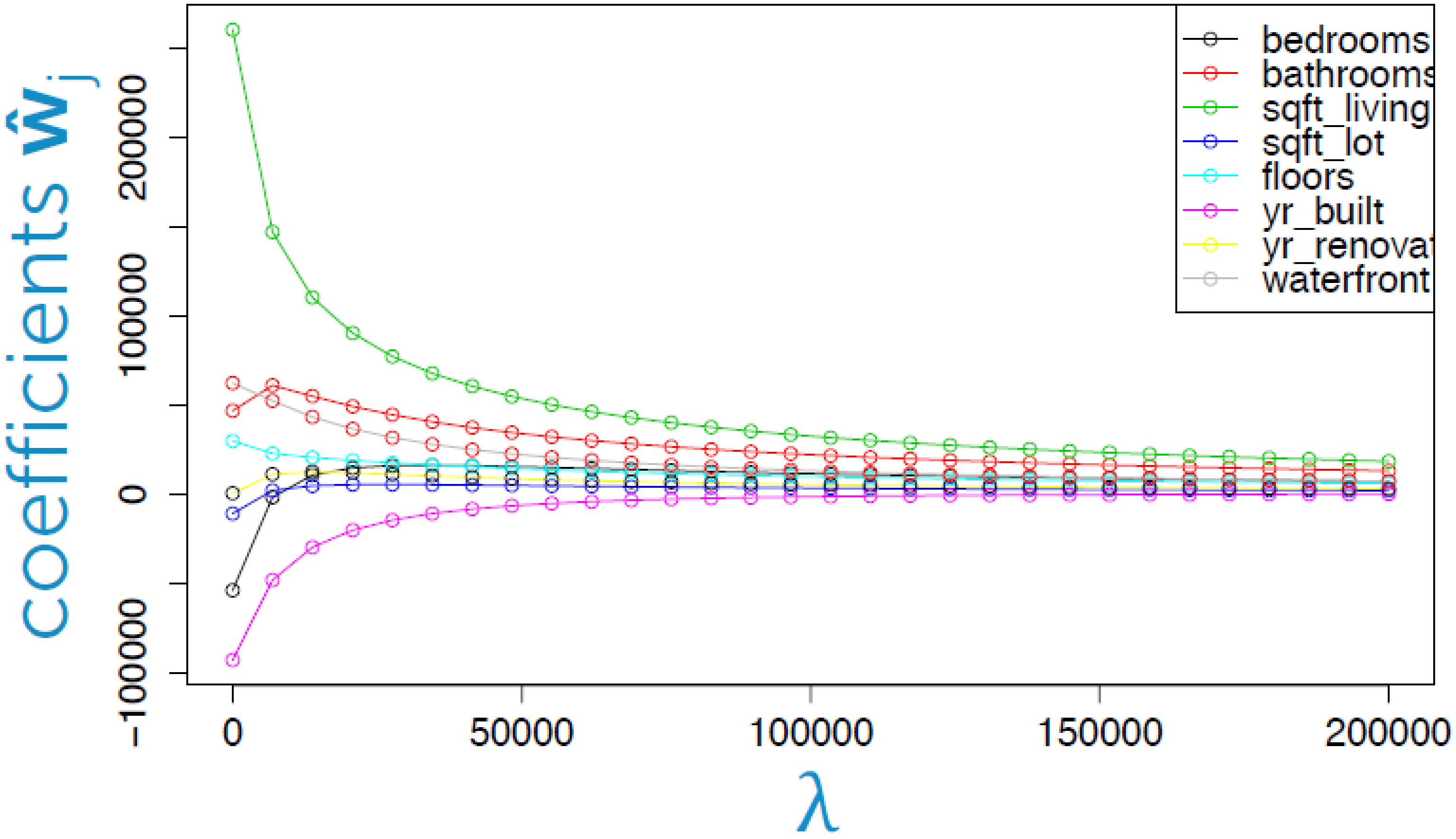


Figure: Emily Fox, University of Washington

L2 vs L1: Housing Price Example

Predict housing price from several features

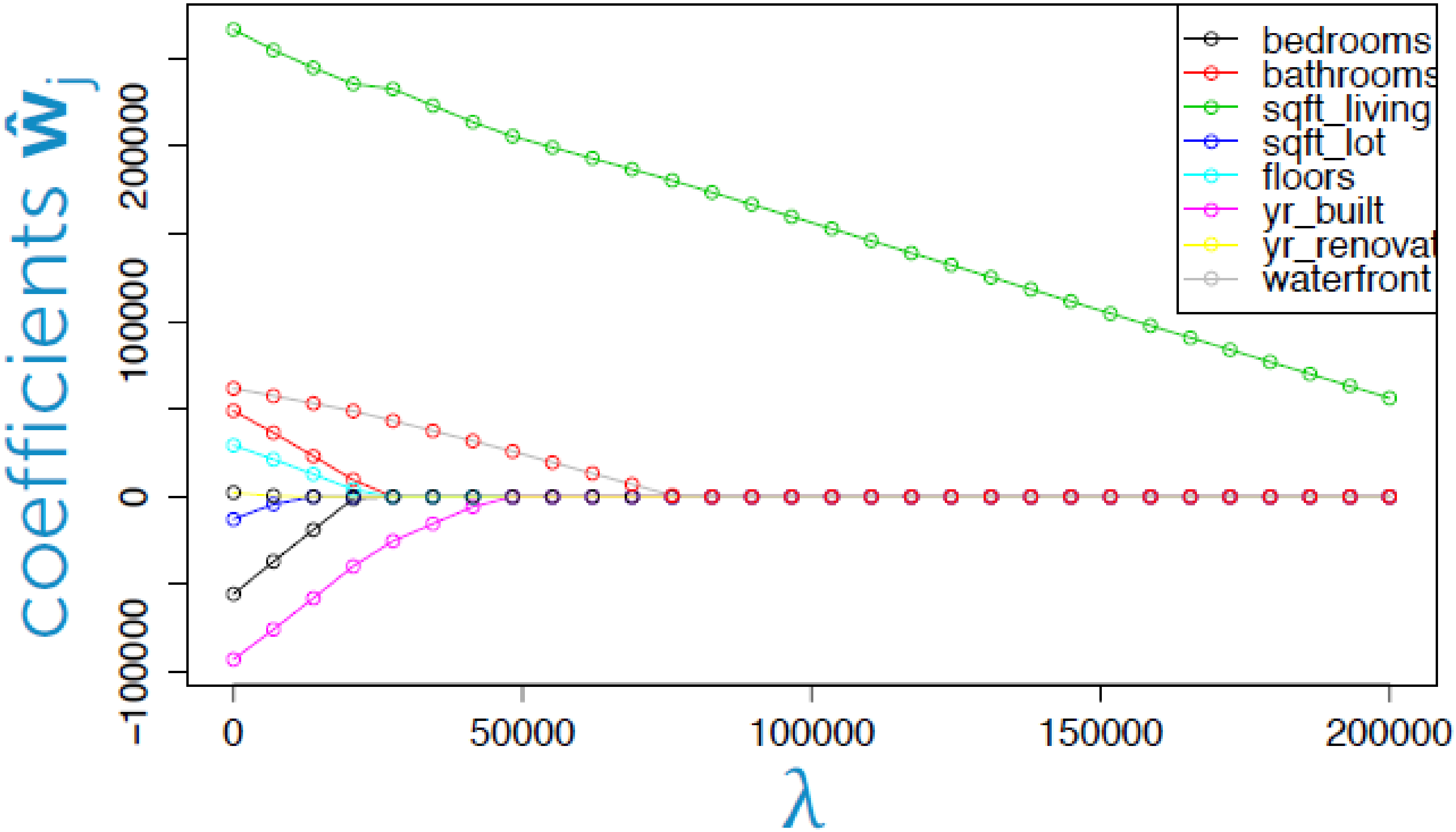


Figure: Emily Fox, University of Washington

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…

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 are **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

Feature Engineering / Regularization Objectives

You should be able to…

- Engineer appropriate features for a new task
- Use feature selection techniques to identify and remove irrelevant features
- Identify when a model is overfitting
- Add a regularizer to an existing objective in order to combat overfitting
- Convert linearly inseparable dataset to a linearly separable dataset in higher dimensions
- Describe feature engineering in common application areas