Regression and Regularization

Lesson 7

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Module Overview

Review HW 6

Finding the best model

- Model Fit
- Measuring the model

Regularization

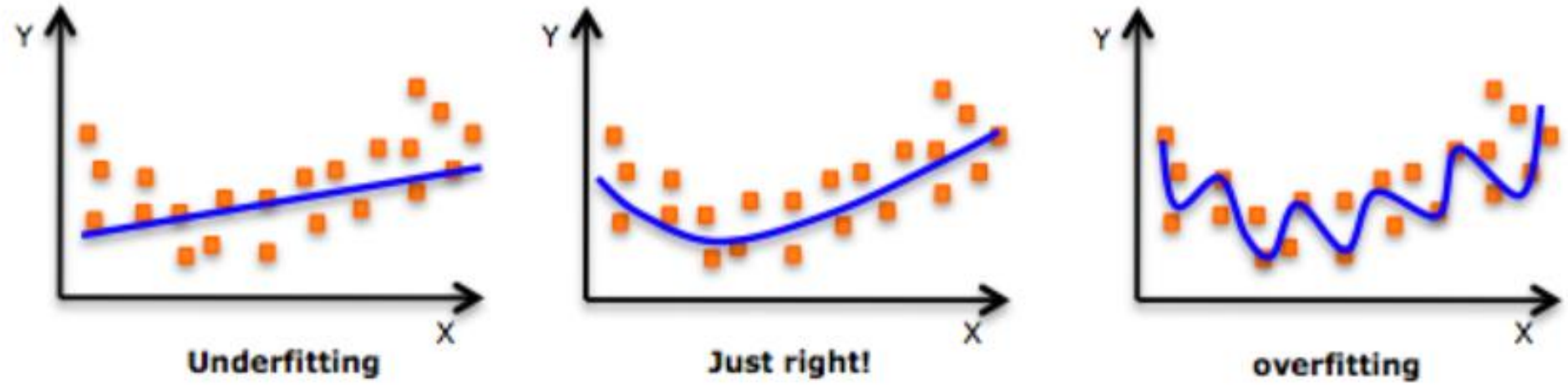
- Stepwise
- Ridge
- Lasso

Logistic Regression

Model Fitness

Model Fitness

- Goodness of fit- How closely a model's predicted values match the observed (true) values
- Overfit – The model works great for your particular dataset but isn't generalizable to the real world
- Underfit- model is too simple, generalizable but not meaningful
  - Less variance but more bias towards wrong answers



Overfitting

- Over-parameterization:
  - Number of parameters > dimensionality of data
  - Linear dependency of input features
  - Works great on your training set, but not generalizable to the real world

Tools for preventing overfit

- Stepwise Regression
  - Eliminate features in step-wise fashion
- Regularization
  - Size of coefficients increase exponentially with model complexity
  - Ridge and lasso methods to stabilize the size of the coefficients

Measuring Fitness

How to measure a model?

Akaike information criterion (AIC)

$$AIC=-2*\ln L+2*k$$

- k: number of estimated parameters
- L: likelihood
- N: number of recorded measurements

Bayesian information criterion (BIC)

$$-2*\ln L+2*\ln N*k$$

Choose the model that has the minimal AIC or BIC

- Balances the level of fit with the model complexity
- Penalize using the sample data to estimate the model parameters

Regularization

What is feature selection?

Process of selecting a subset of features that are good predictors of the target

- Useful for
  - Controlling complexity of model
  - Speed up model learning without reducing accuracy
  - Improve generalization capability

Stepwise Regression

- Forward:
  - Start with a model with only inception
  - Add one feature in the model at each step
  - At each step, the variable that can maximally reduce the residual sum of squares (RSS) is chosen as the feature to add in the model.
- Backward:
  - Start with a model with all features
  - Remove one feature from the model at each step
  - At each step, the variable that can minimally increase the residual sum of squares (RSS) is chosen as the feature to remove from the model.
- Both:
  - At each step, will check whether add a feature, or remove a feature

Stepwise Regression

- Use when you don't have a lot of prior knowledge about your dataset
- Not the best option
  - High bias towards false positives
  - Underestimates standard error
  - So basically not as accurate as you'd like
- Better option?
  - Ridge and lasso

Cost

Cost functions

A cost function determines how well your model can make predictions.

For regression

- Sum of squared error
- Difference between estimated and real values

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Sum of Squared Error

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

m = number of samples

$$\frac{1}{2m}$$

Constant ½ * number of samples

$$\sum_{i=1}^{m}$$

For 1 to x samples, do the following and sum:

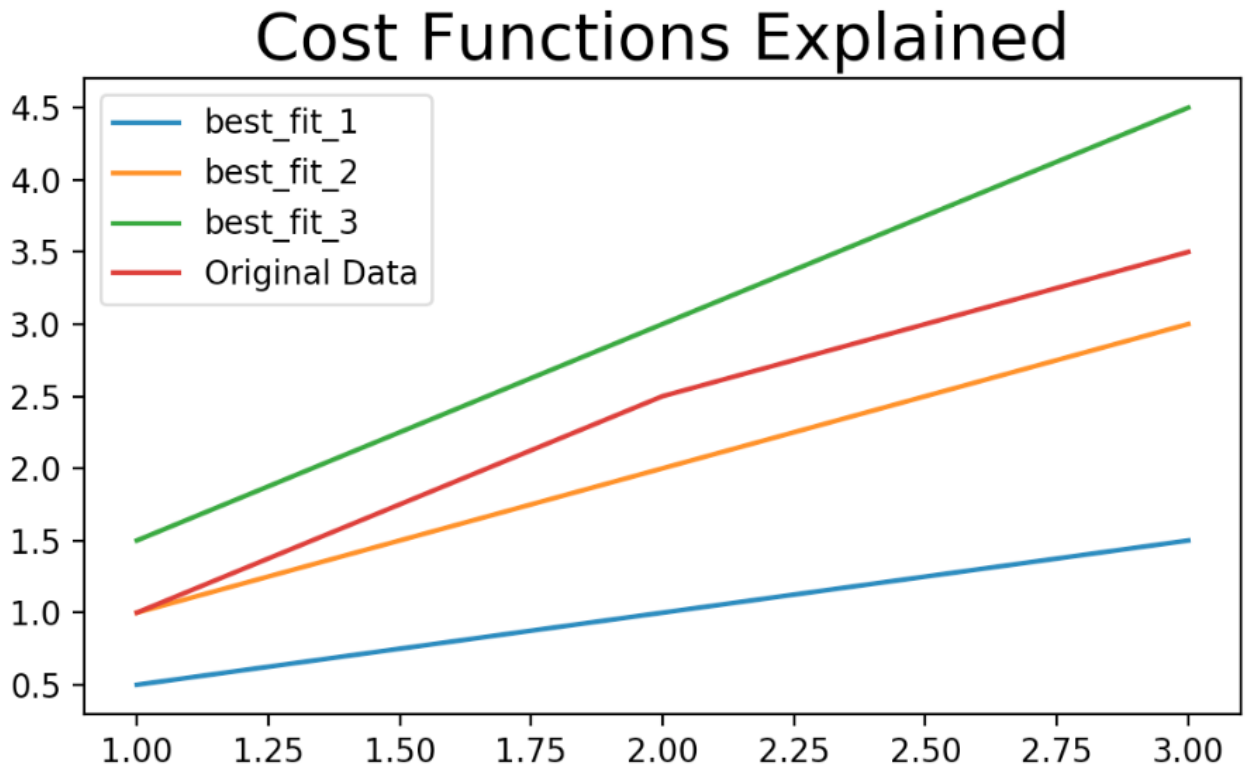
$$(h_\theta(x^{(i)}) - y^{(i)})^2$$

h(0) is the hypothese value of x - Actual value of y, square outcome

Minimizing Cost

Let's say we have three hypothesis to determine what might be happening in our data.

We've used linear regression to model the line of best fit for each scenario. Which one best represents our data?



X	y	best_fit_1	best_fit_2	best_fit_3
1.00	1.00	0.50	1.00	1.50
2.00	2.50	1.00	2.00	3.00
3.00	3.50	1.50	3.00	4.00

Best_fit = slope of the line

Example: Calculating Cost

X	y	best_fit_1
1.00	1.00	0.50
2.00	2.50	1.00
3.00	3.50	1.50

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

For x = 1:

1/2m = 1 / (2 * 3) = 1/6

Observed = hθ(x(i)) = 0.50

Acutal = y = 1.00

So

(0.50—1.00)^2

= 0.25

Example: Calculating Cost

X	y	best_fit_1
1.00	1.00	0.50
2.00	2.50	1.00
3.00	3.50	1.50

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

For x = 2:

1/2m = 1 / (2 * 3) = 1/6

Observed = hθ(x(i)) = 1.00

Acutal = y = 2.50

So

(1.0—2.50)^2

= 2.25

Example: Calculating Cost

X	y	best_fit_1
1.00	1.00	0.50
2.00	2.50	1.00
3.00	3.50	1.50

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

For x = 3:

1/2m = 1 / (2 * 3) = 1/6

Observed = hθ(x(i)) = 1.50

Acutal = y = 2.50

So

(1.50—3.50)^2

= 4.00

Example: Calculating Cost

X	y	best_fit_1
1.00	1.00	0.50
2.00	2.50	1.00
3.00	3.50	1.50

$$\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

Sum Results:

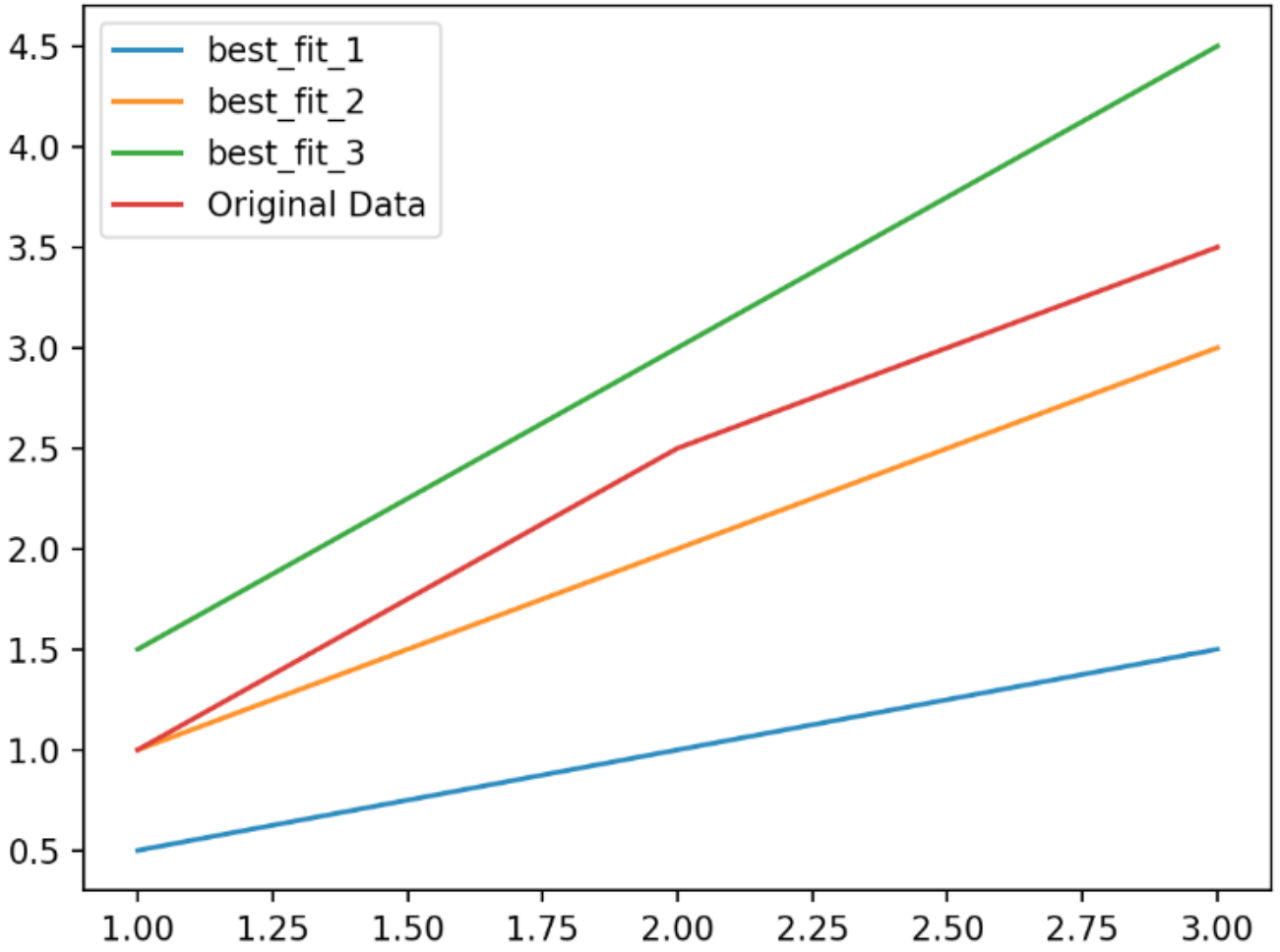
Sum([0.25, 2.25, 4.00])

Multiply by constant: 1/6

= 6.5 * (1/6) = 1.083

Repeat.

Example: Calculating Cost



Results for each line:

```
best_fit_1: 1.083
best_fit_2: 0.083
best_fit_3: 0.25
```

Lowest cost =

best_fit2

Cost in Regressoin

Regression Cost

- When measure of fit (RSS) is small = good fit
- When magnitude of coefficients (sum of squared error) is small = good fit

Residual Sum of Squares

RSS- measures the variability left unexplained after performing the regression:

$$RSS = \sum_{i=1}^{n} (\varepsilon_i)^2 = \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2$$

- Where $y_i = a + bx_i + \varepsilon_i$
  - Y is predicted, X is predictor
  - And ε is the error term
- Where alpha and beta are coefficients (slope, intercept)
- RSS = measure distance from each data point, sum, square

Regularization

Regularizatoin seeks to control variance by adding a tuning parameter λ (lambda) or alpha to balance the coefficients:

- Ridge Regression= L2 Regularization
  - Penalizes the size (square of magnitude) of the regression coefficients
  - Enforces the β (slope) coefficients to be lower, but not 0
  - So it will not remove irrelevant features, but minimizes their impact
- LASSO
  - Regularization term penalizes absolute value of the coefficients
  - Will set irrelevant values to 0
  - But you might end up with less features in your model

Ridge Regression

Ridge Regression

Ridge Objective = RSS + λ * (sum of square of coefficients)

- Uses λ to balance minimizing RSS versus minimizing the coefficients
- When λ = 0, results are same as regular linear regression
- As λ increases to infinity, coefficients become closer to 0
- When 0 < λ < ∞ the magnitude of α determines the balancing of the objective
- **Take home** = any non-zero value will shrink the coefficients to be smaller than with non-regularized regression

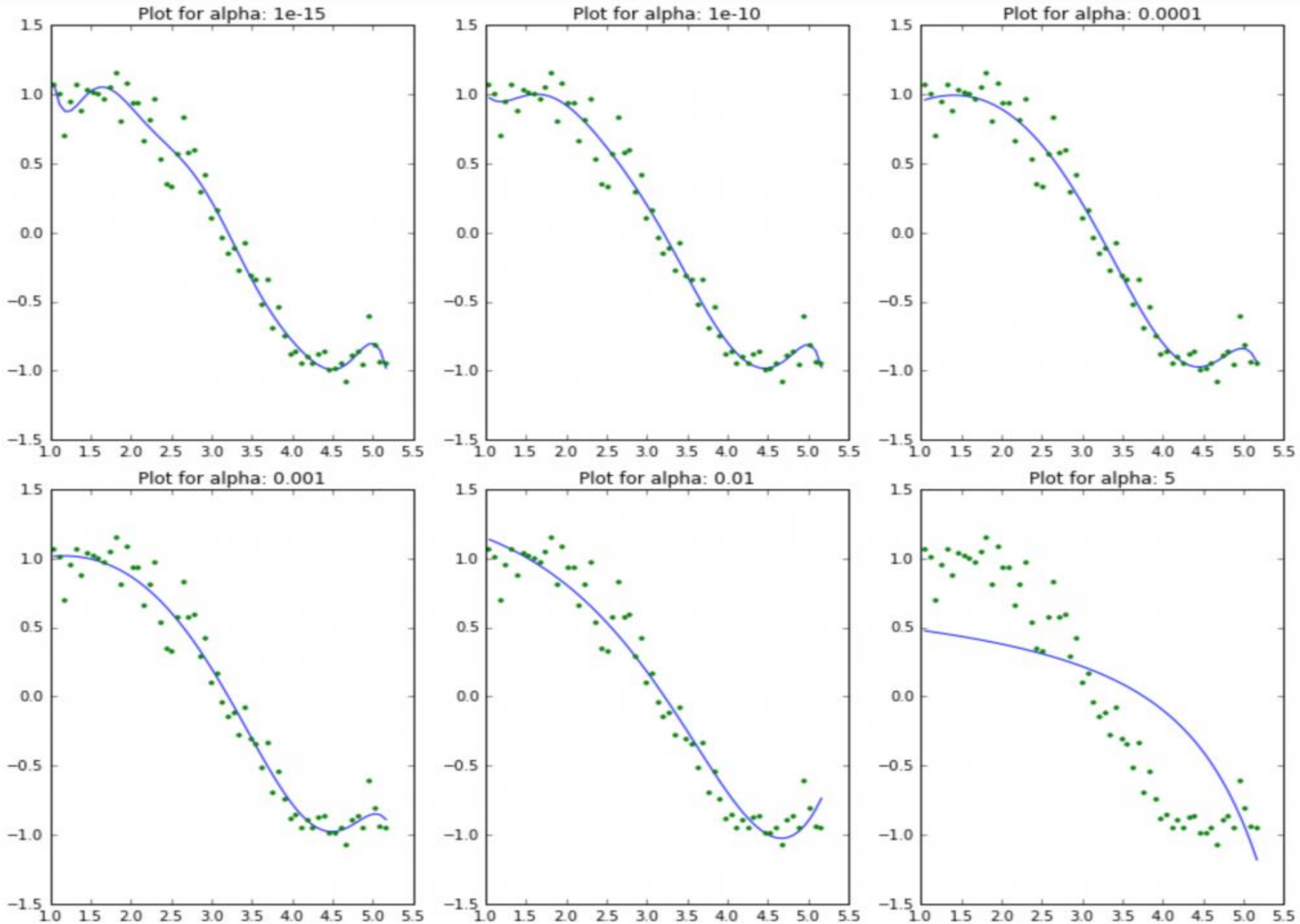
$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum \beta^2$$

1. Sum distances between prediction and truth
2. Sum squared beta values
3. Multiply by lambda

Lambda is a scalar that should be learned from your data using cross-validation

Ridge Regression

As α increases, model complexity decreases



LASSO:

Least Absolute Shrinkage and Selection Operator

Lasso Regression

Lasso Objective = RSS + λ * (sum of absolute value of coefficients)

- Similar to Ridge
  - Λ is a trade-off between balancing coefficients and RSS
  - Difference is that regularization term is of the absolute value of coefficients
- Punishes high values of Beta, and sets irrelevant features to 0

$$L = \sum(\hat{Y}_i - Y_i)^2 + \lambda \sum |\beta|$$

1. Sum distances between prediction and truth
2. Sum absolute values of beta
3. Multiply by lambda

Logistic Regression

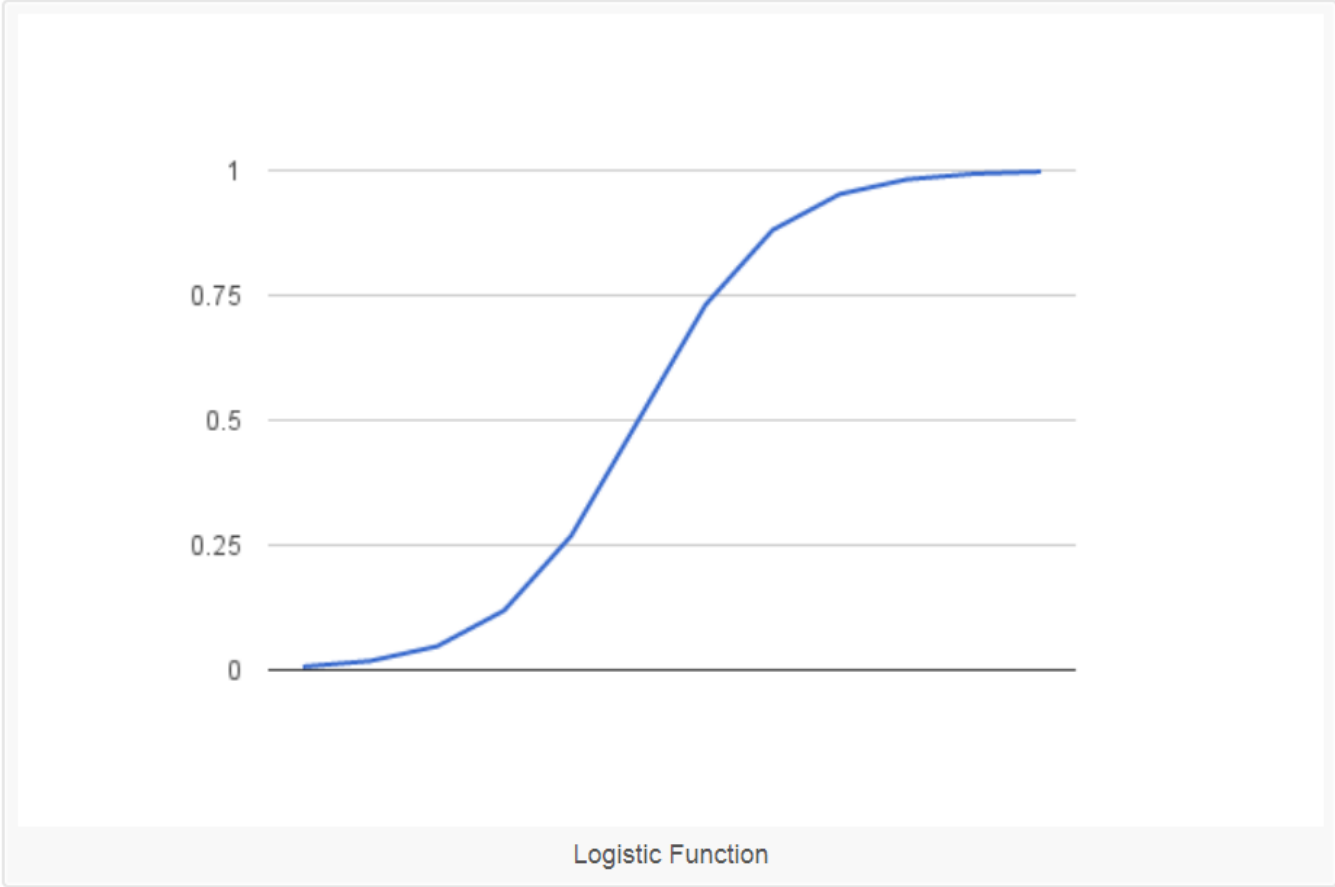
Logistic Regression- Predicting Categorical Variables

- Predict the probability of a categorical variable
  - Ex: Is this email spam or not?
- Dependent variable is binary
  - Encoded as 0 or 1
  - You are predicting P(Y=1), or the probability of success
  - Results range from 0 to 1
- Assumptions
  - Binary dependent value
  - Independent variables are not correlated, errors are not correlated
  - There is a linear relationship
  - Larger sample sizes

Logistic Function

- Designed by ecologists to model carrying capacity
- 1 / (1 + e^-x)
- e= Euler's constant aka base of natural log
- Takes any real number and maps between 0 and 1

Log transform of [-5,-4,-3,-2,-1,0,1,2,3,4,5]



Logistic Regression

Logistic regression is also called the sigmoid function or 'logit' model:

- **When y is categorical, we use the logit of y as our response instead of y itself.**
- logit(p) = log(odds) = log(p/q)
- **Original model:** $y_i = \beta_0 + \beta_1 x_1 + \varepsilon_0$

Beta 0 = y-intercept
Beta 1 = first coefficient
X1 = first predictor
E = euler's constant

- **Logit (log odds) model:** $\ln\left[\frac{p_i}{1-p_i}\right] = \beta_0 + \beta_1 x_1 + \varepsilon_0$

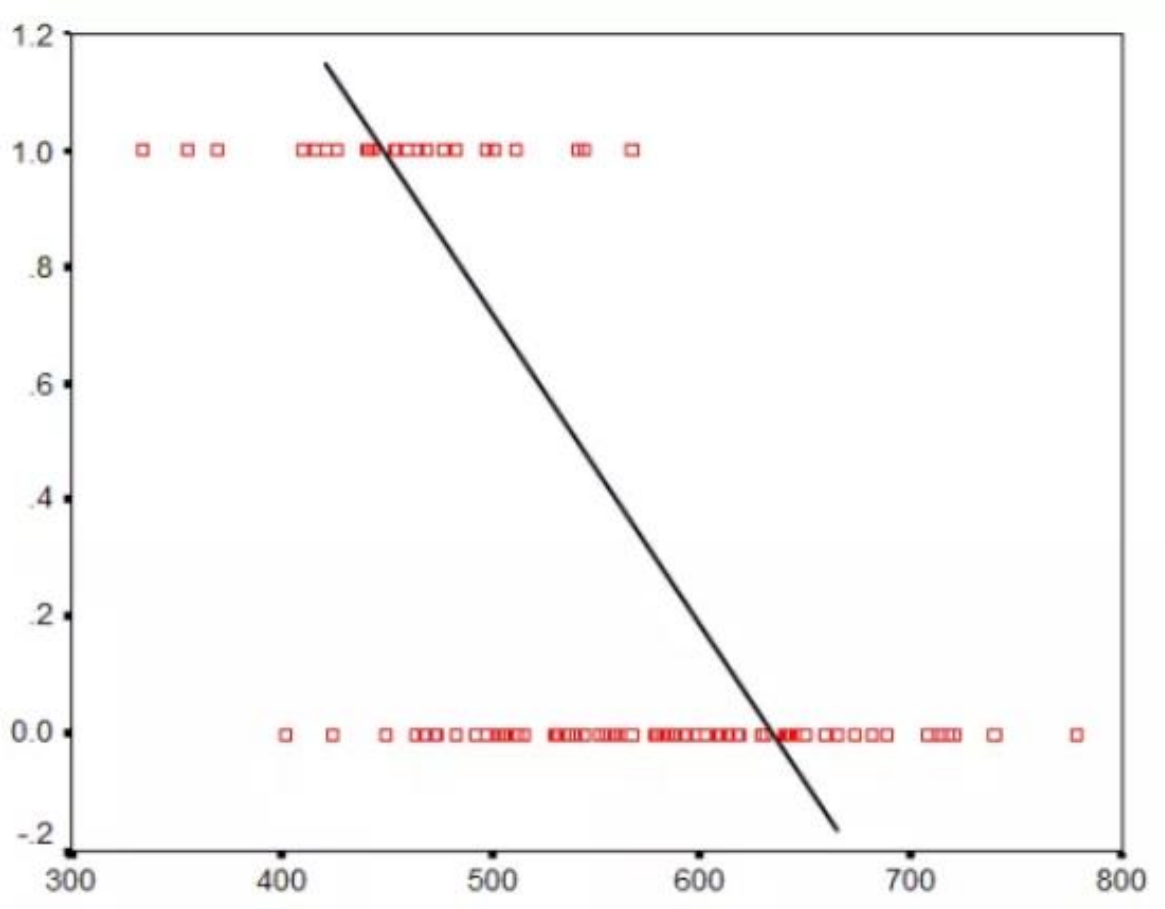
Where p = probability that y is in a category

Log-odds-ratio

- **So estimated probabilities follow: (solving for p)** $p_i = \frac{1}{1+\exp(-(\beta_0 + \beta_1 x_1))}$

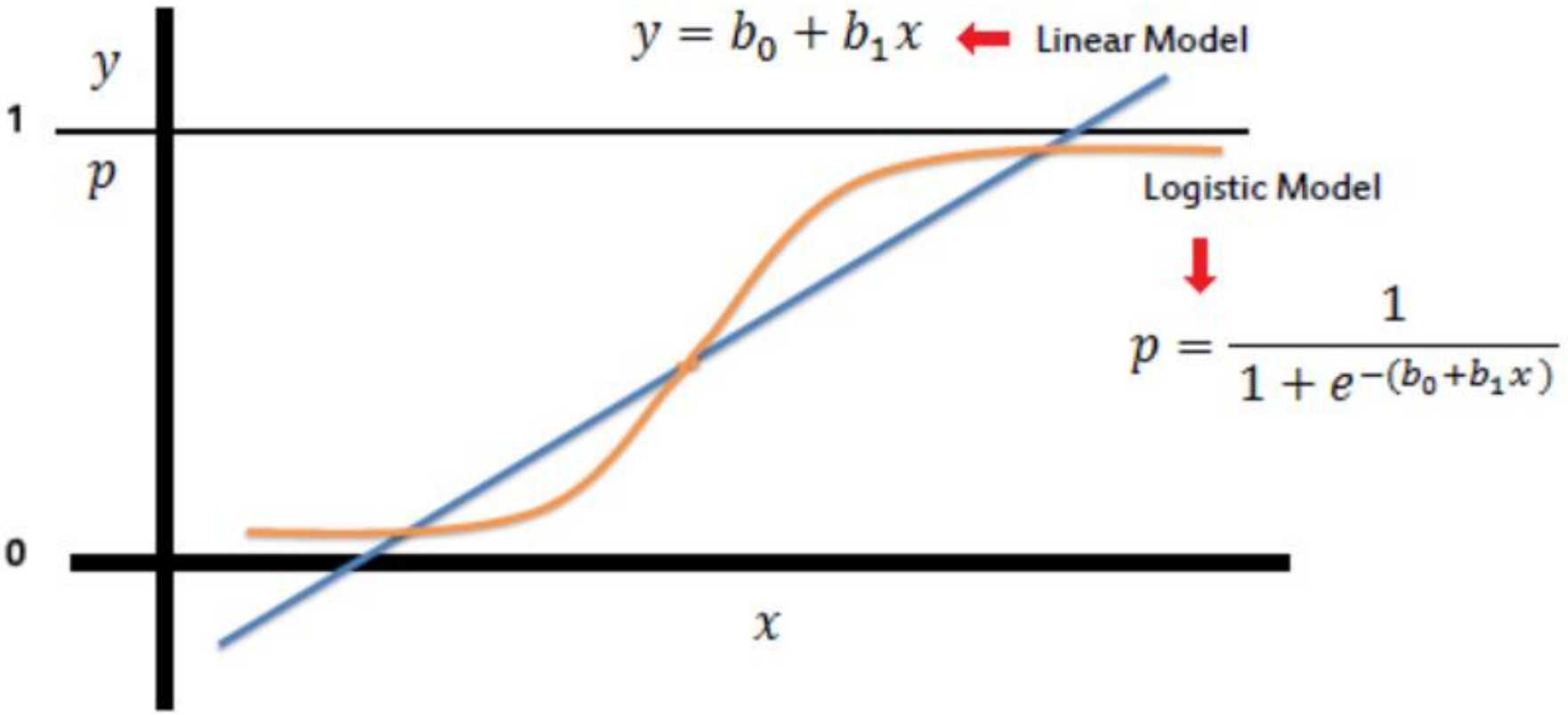
Logistic Regression

Why can't we just use normal regression?



Logistic Regression

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1))}$$



- Constant (b_0) moves the curve left and right
- slope (b_1) defines the steepness of the curve

Logistic Regression

Differences between linear and logistic regression.

Predictions

- Linear regression outcomes are unbounded.
- Logistic regression outcomes are bounded between 0 and 1.

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1))}$$

Error distribution

- Linear regression errors are normally distributed.
- Logistic regression errors are Bernoulli distributed.

Multinomial Logistic Regression

"Multinomial"- when the dependent variable is nominal with more than two levels

Logistic regression is built for only two classes.

With predicting N classes, we can work-around this with two options.

- A) Creating N 1-vs-all logistic regressions. Predictions would then be which ever class prediction has the highest probability.
- B) Creating (N Choose 2) pairwise logistic regressions on all pairs of classes. Predictions are then the class with the most "votes".

Option A is the most common, especially with more classes because option B is much more computationally intense.

Regression and Regularization

Lesson 7