Applied Machine Learning with Big Data "EE 6973"

Topic: Linear Regression

Paul Rad, Ph.D.

Chief Research Officer
UTSA Open Cloud Institute(OCI)
University of Texas at San Antonio

Logistics

Instructor: Paul Rad

- Email: paul.rad@utsa.edu
- Office: OCI - NPB 3rd floor
- Office hours: Tuesday &Thursday 10:15:11:00 (30 minutes after our class)

Course Web: https://github.com/ml6973/Course

Mailing list: TBD

Course Social Network: TBD

Outline

Prerequisites:

Calculus

Python

Linear Algebra

Probabilitiy

3 Types of Learning

- Learning from labeled data
- E.g., Spam classification

- Classification
- Regression
- Ranking

- Discover structure in unlabeled data
- E.g., Document clustering

- Clustering
- Hidden Markov Models

- Learning by "doing" with delayed reward
- E.g., Chess computer

Supervised Learning

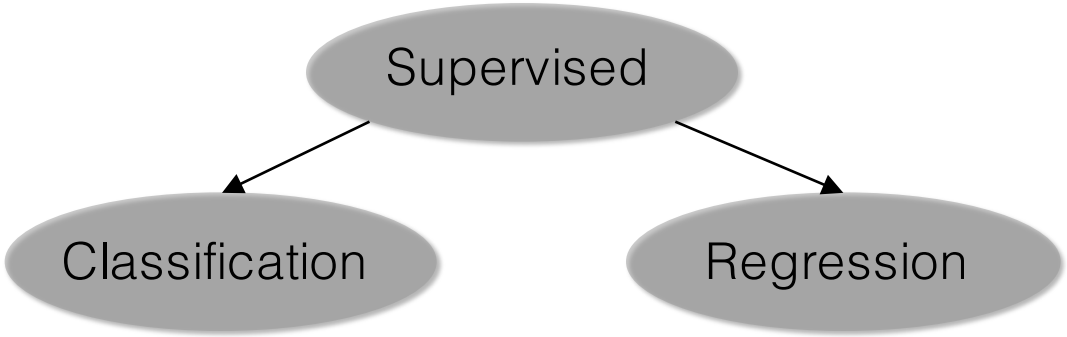
Given examples of a function *(X, F(X))*

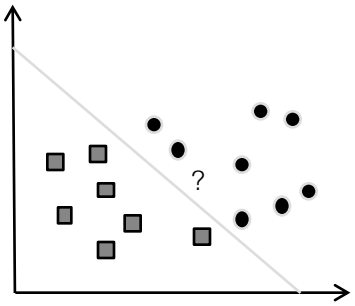
Predict function *F(X)* for new examples *X*

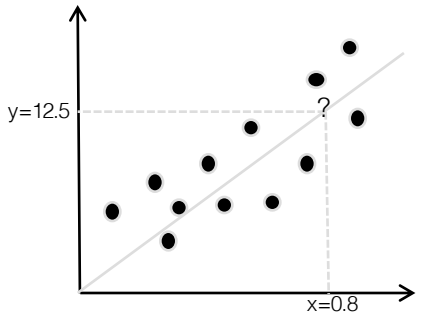
Discrete *F(X)*: Classification

Continuous *F(X)*: Regression

F(X) = Probability(*X*): Probability estimation







Regression and Classification Examples

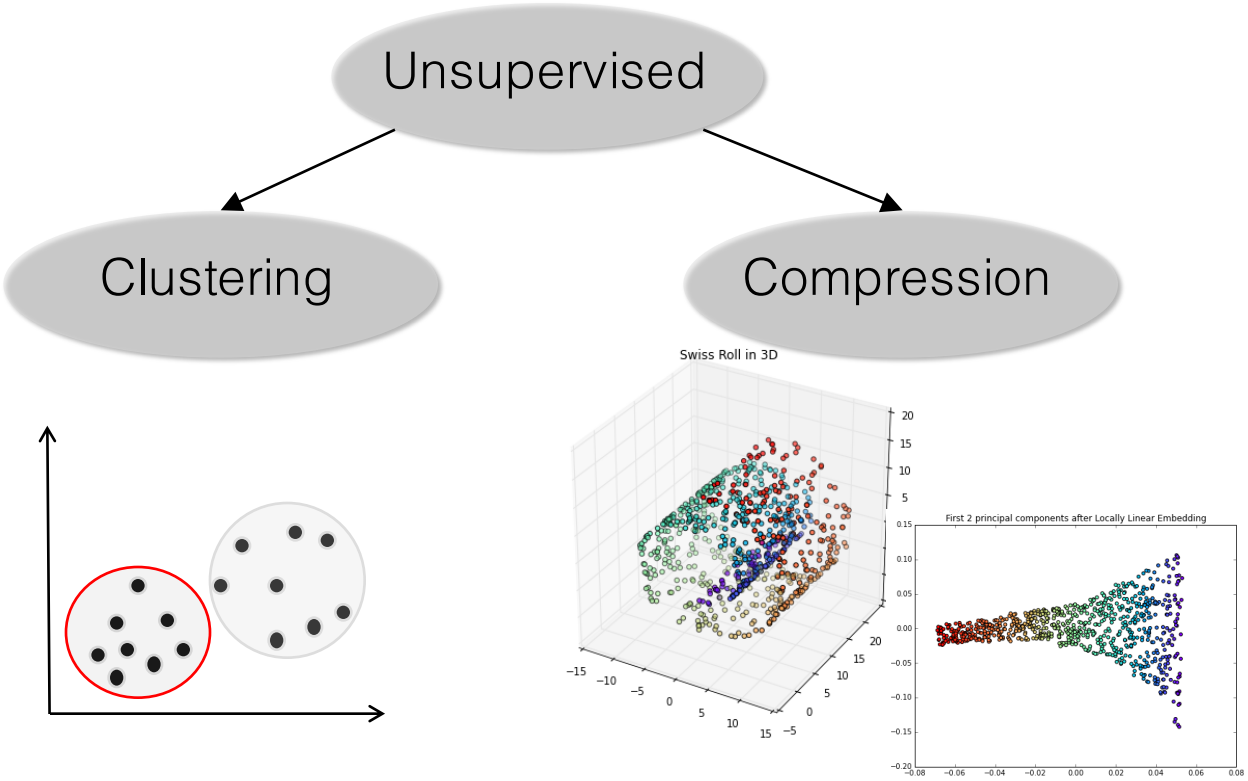
Stock prediction

* Predict the price of a stock (y)
  * Depends on x =
    - Recent history of stock price
    - News events
    - Related commodities

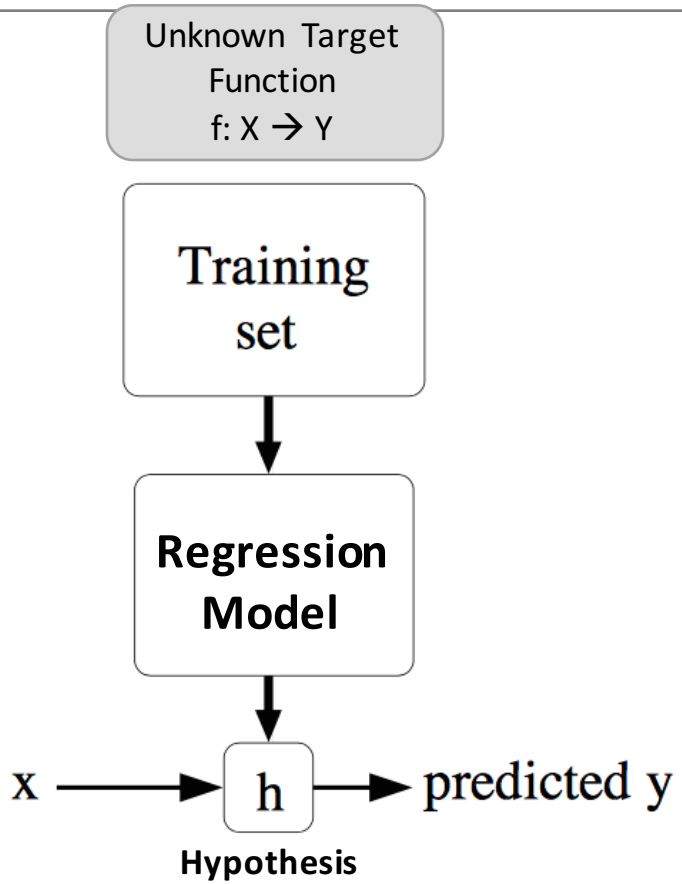
Spam or Not spam emails

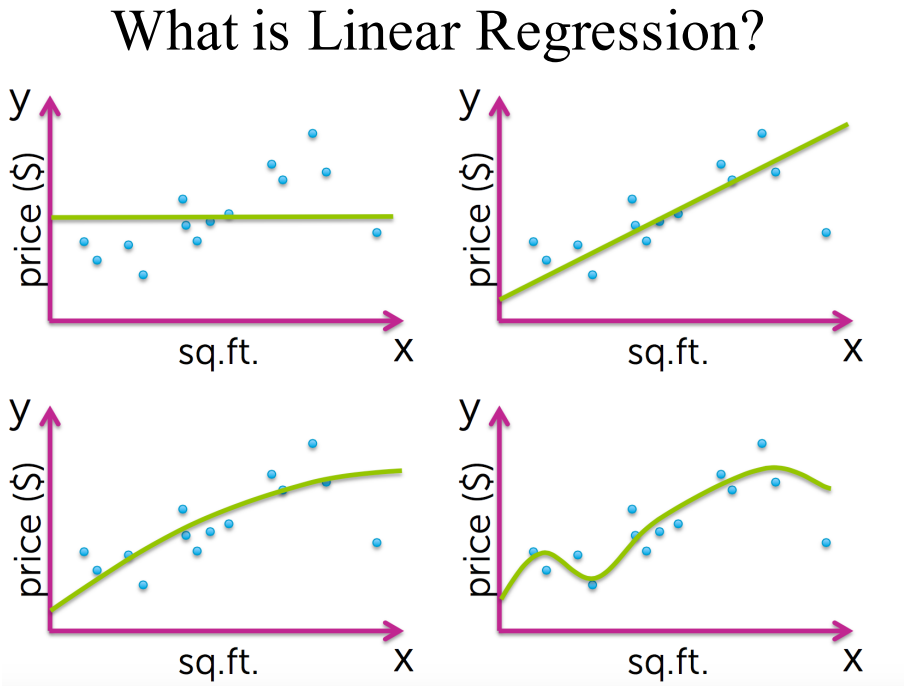
Music or Tweeter Sentiment Analysis

Unsupervised Learning



Linear Regression Learning Model

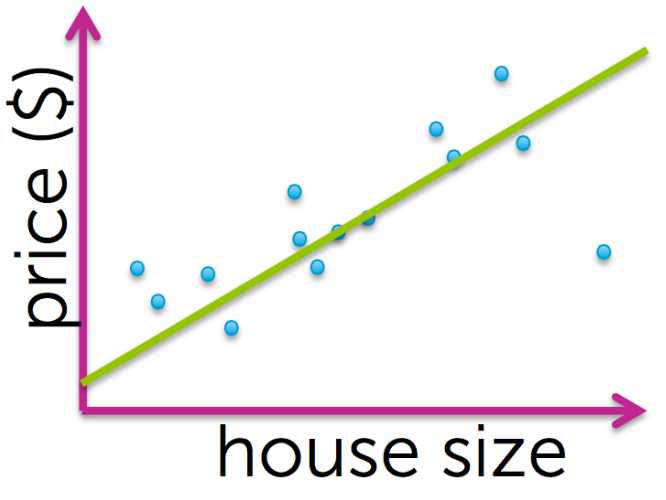




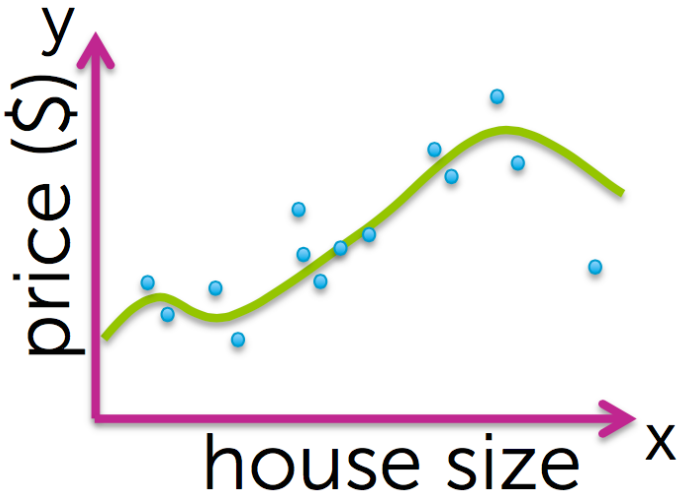
Simple Regression vs. Multiple Regression

What makes it simple?

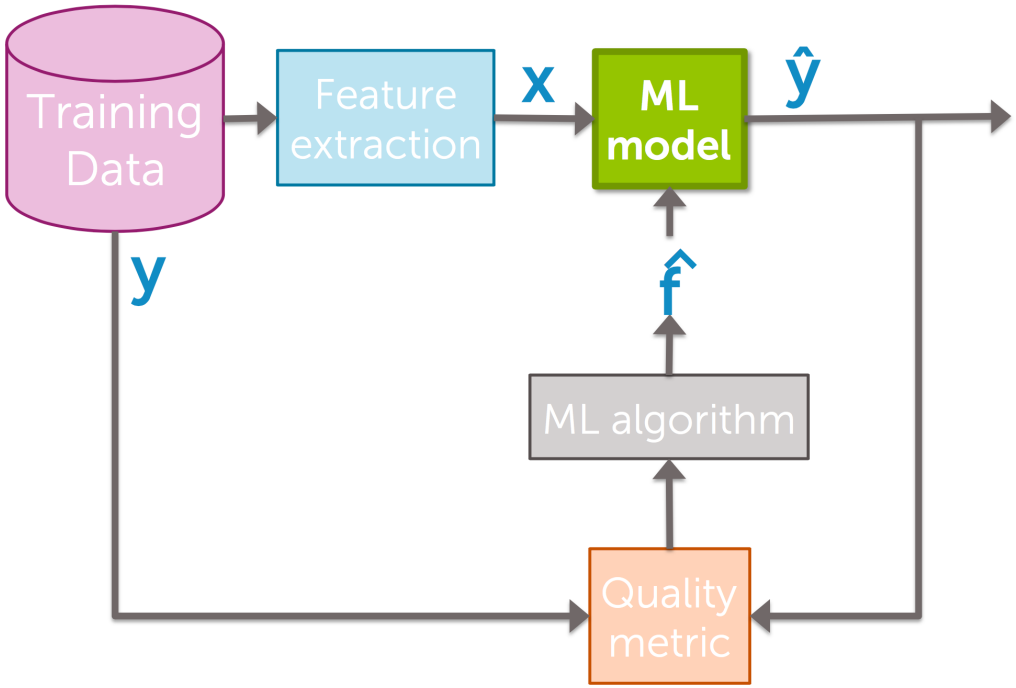
1 input and just fit a line to data

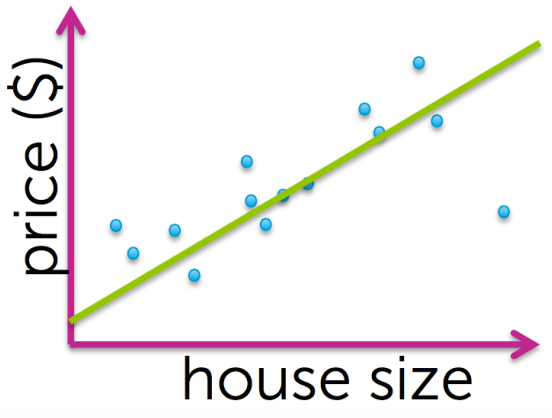


Fit more complex relationships than just a line



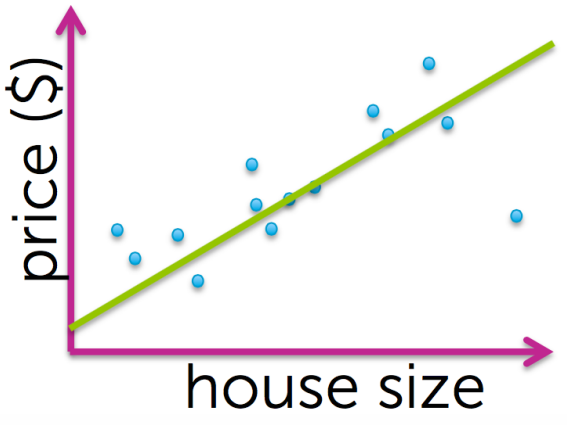
Regression Model





Predicting house prices

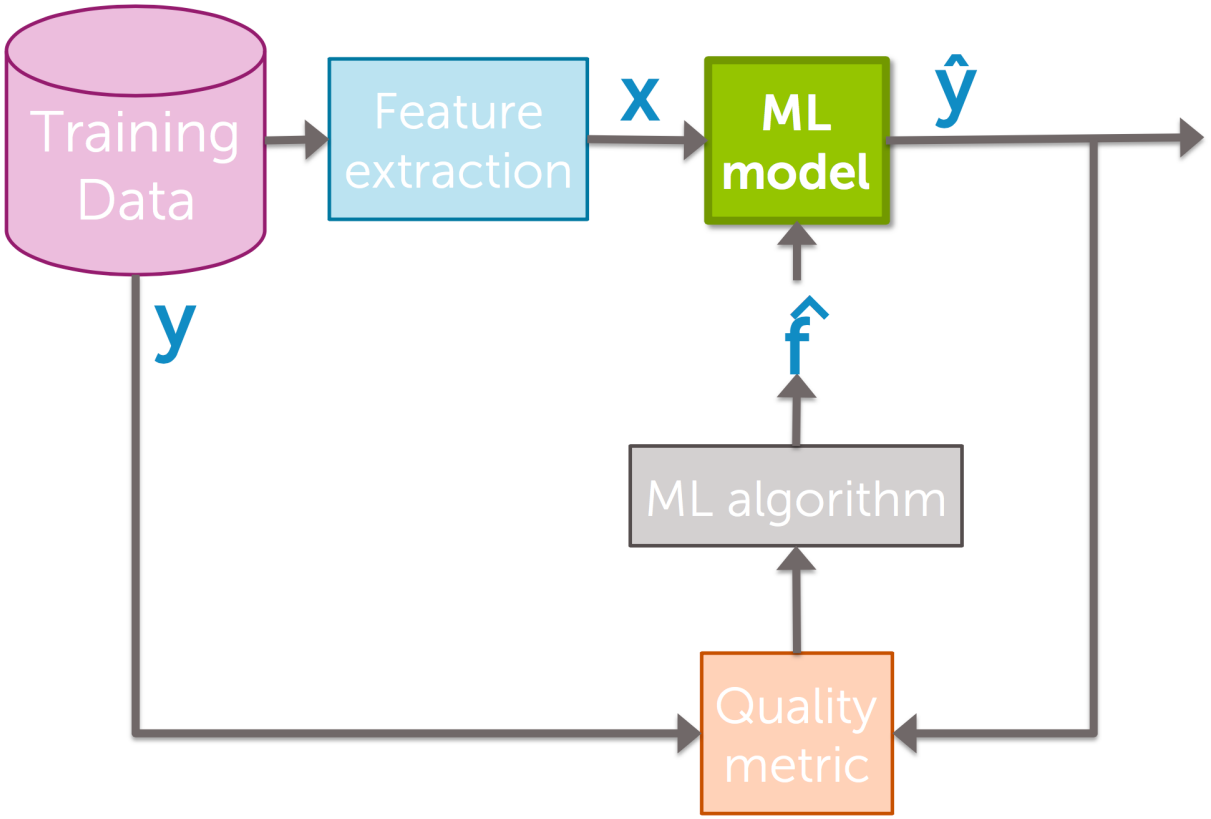
How much is my house worth?



Look at recent sales in my neighborhood

- How much did they sell for?

Regression Model



Regression Algorithm

We will need a starting value for the slope and intercept, a step_size and a tolerance

initial_intercept = 0, nitial_slope = 0, step_size = 0.05, tolerance = 0.01

The algorithm

In each step of the gradient descent we will do the following:

1. Compute the predicted values given the current slope and intercept

2. Compute the prediction errors (prediction - Y)

3. Update the intercept:

compute the derivative: sum(errors)

compute the adjustment as step size times the derivative

decrease the intercept by the adjustment

4. Update the slope:

compute the derivative: sum(errors*input)

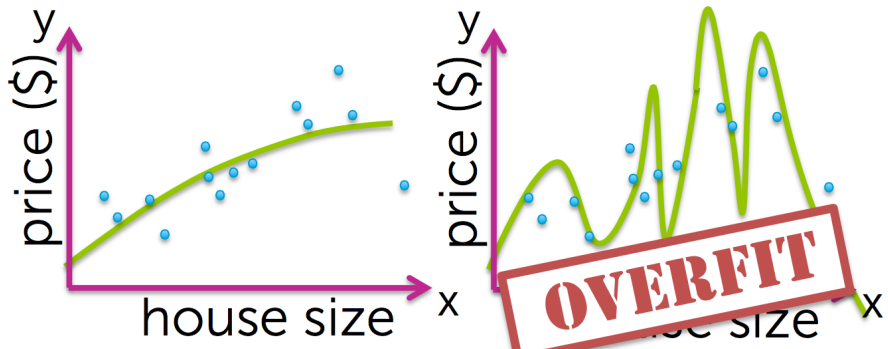
compute the adjustment as step_size times the derivative

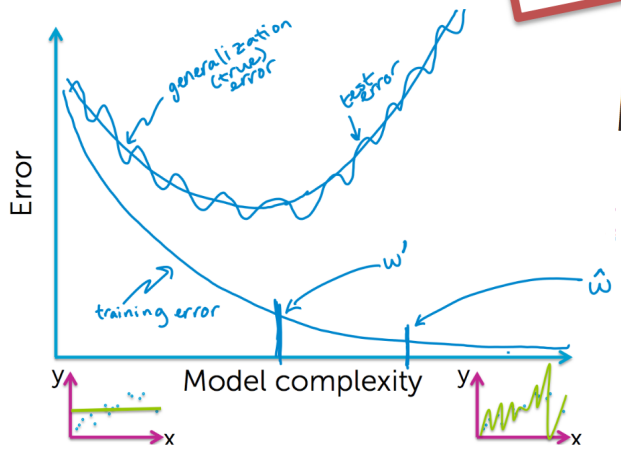
decrease the slope by the adjustment

5. Compute the magnitude of the gradient

6. Check for convergence

Assessing Performance





Measures of error:

- Training
- Test
- True (generalization)

Real world examples

X = # hours of exercise in a week → Y = body mass index

Y = # hours of studying in a week → Y = grade for the course

X

Y

Objective/Error/Cost Function

Thank you

Question #1

What is an appropriate application of linear regression?

1) Predicting a person's height, given the amounts of certain substances in their diet

2) Predicting whether or not a person likes cake, given the amounts of certain substances in their diet

Question #2

What would lead to a negative r-squared?

1) We predict the opposite of the target value

2) We predict the negative of the target value

3) We predict worse than the mean of the target values

Assignment #1 - closed form example

Computing regression parameters The data

Consider the following 5 point synthetic data set:

X: 0,1,2,3,4

Y: 1,3,7,13,21

We want the line that “best fits” this data set as measured by residual sum of squares

In summary, we have:

slope = 5, intercept = -1

Finally we can add the line to our plot from above