DSA5101 Introduction to Big Data for Industry

Lecture 4 Brief Introduction of Regression

Li Xiaoli

National University of Singapore

Outline

- What is Regression
- Evaluation for Regression Models

Regression Analysis

- In machine learning or statistical modeling, regression analysis is a set of statistical processes for estimating the **relationships among variables**.
- It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between ***a dependent variable*** (often called the 'outcome' or 'response' variable, or label in ML) and ***one or more independent variables*** (often called 'predictors', 'covariates', 'explanatory variables' or 'features').
- Regression analysis helps one understand how the typical value of the dependent variable changes when any one of the independent variables is varied, while the other independent variables are held fixed.

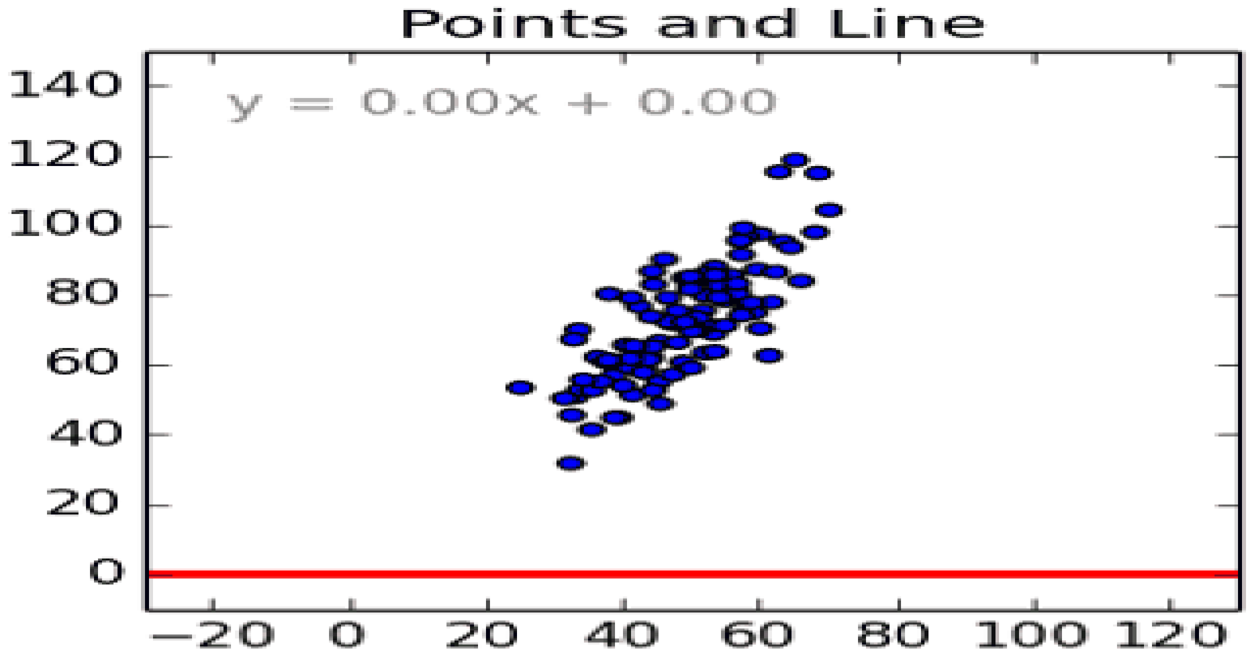
From Wikipedia….

Regression

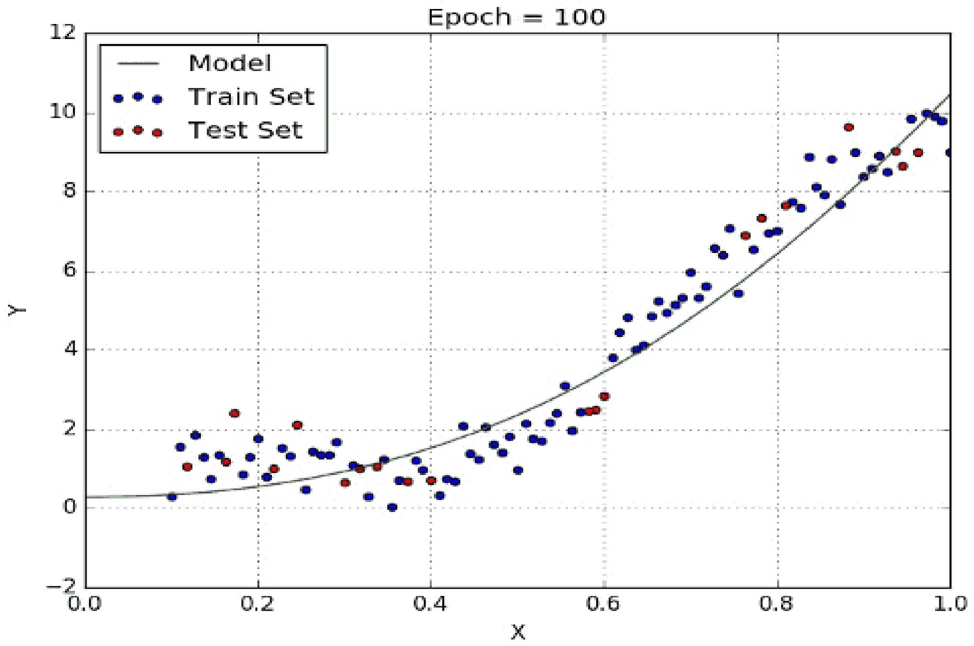
- **Goal:**
  - Predict a value of a given **continuous valued variable** based on the values of other variables, assuming a linear or nonlinear model of **dependency**.
  - Extensively studied in statistics, neural network fields.
- **Examples:**
  - Predicting when the aircraft will land into the airport.
  - Predicting sales amounts of new product based on advertising expenditure.
  - Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
  - Time series prediction of stock market indices.

Most common types of regression algorithms

- Linear Regression
- Polynomial Regression (relationship between *x* and *y* is modelled as an *n*-th degree polynomial in *x;* can model fairly complex relationships; need data knowledge to select the best exponents)
- Ridge Regression and Lasso Regression (alleviate collinearity amongst regression *independent* variables)



Linear Regression



Polynomial Regression

Regression Example 1

Relationship between **systolic blood pressure** (y), **birthweight** (x_1), and **age** (in days) (x_2)

i	Birthweight in oz (x_1)	Age in days (x_2)	Systolic BP mm HG (y)
1	135	3	89
2	120	4	90
3	100	3	83
4	105	2	77
5	130	4	92
6	125	5	98
7	125	2	82
8	105	3	85
9	120	5	96
10	90	4	95
11	120	2	80
12	95	3	79
13	120	3	86
14	150	4	97
15	160	3	92
16	125	3	88

Training regression model:

Use linear regression method to determine the regression eqn:

$$y=53.45 + 0.126 * x_1 + 5.89 * x_2$$

Prediction using model:

To predict the systolic BP of a baby with birthweight 8 lb (**128 oz**) measured at **3** days of life

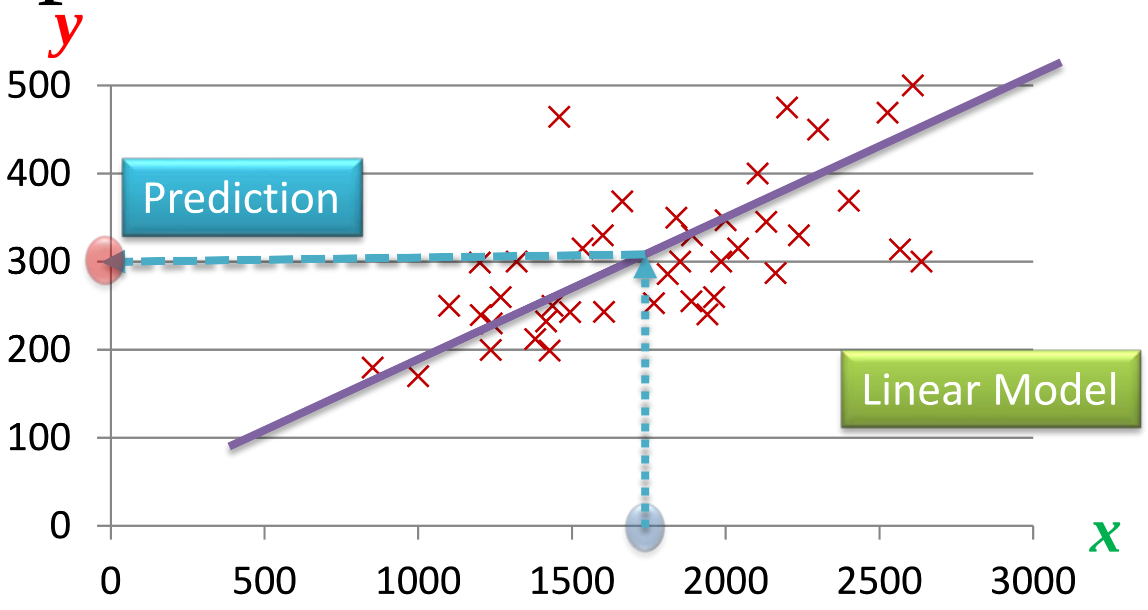
$$y=53.45+0.126*(128)+5.89*(3)$$

$$=87.2\text{mm Hg}$$

Regression Example 2

Price (in 1000s of $)
Dependent/response Variable

Problem: Housing Price Prediction based on the size of living areas



Size (feet^2)
Independent variable

We can first learn a model and then use it for prediction

- Regression is *Supervised Learning*
  - Learn a model based on a training data, where each training example contain a answer, i.e. real value for dependent variable
- Difference between regression and classification
  - Regression: predict real-valued output (e.g. actual price of a property)
  - Classification: predict discrete valued output (e.g. property up +1 or down -1)

Training Set and Hypothesis

- Training Set T, size |T|=*m*=100

Index	Size in feet² (x)	Price ($) in 1000's (y)
1	2094	446
2	1675	245
3	1452	326
4	837	183
…	…	…
100	3378	718

Input/ Independent variable

Output/ Dependent/Target/ Response variable/Label

Training set representation

$(x^{(i)}\ y^{(i)})$: *i-th* training data

$x^{(1)}=2094,\quad y^{(1)}=446$

$x^{(2)}=1675,\quad y^{(2)}=245$

$x^{(3)}=1452,\quad y^{(3)}=326$

$x^{(4)}=837,\quad y^{(4)}=183$

…..

$x^{(100)}=3378,\quad y^{(100)}=718$

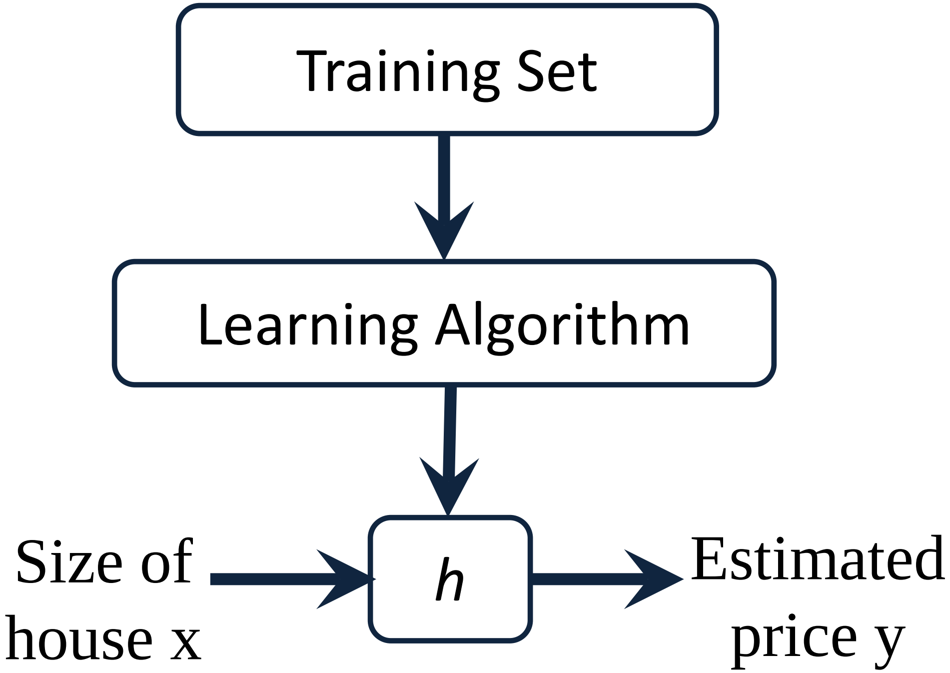
- Hypothesis *h*: we will build a linear model with equation

$$y = h_\theta(x) = \theta_0 + \theta_1 x,$$

θ_i (*i*=0, 1): model parameters

- Learning question: How to choose θ_i

Supervised learning for regression



h maps from x to y. Hope it is a very good function/hypothesis

$$y = h_\theta(x) = \theta_0 + \theta_1 x$$

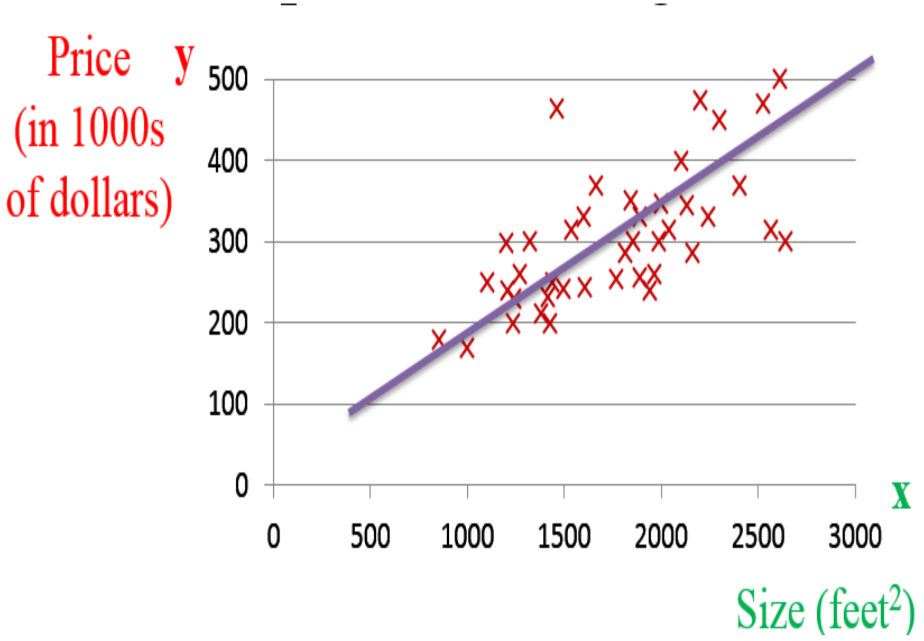
Linear regression with one variable.
Univariate linear regression.

How do we represent *h* ?

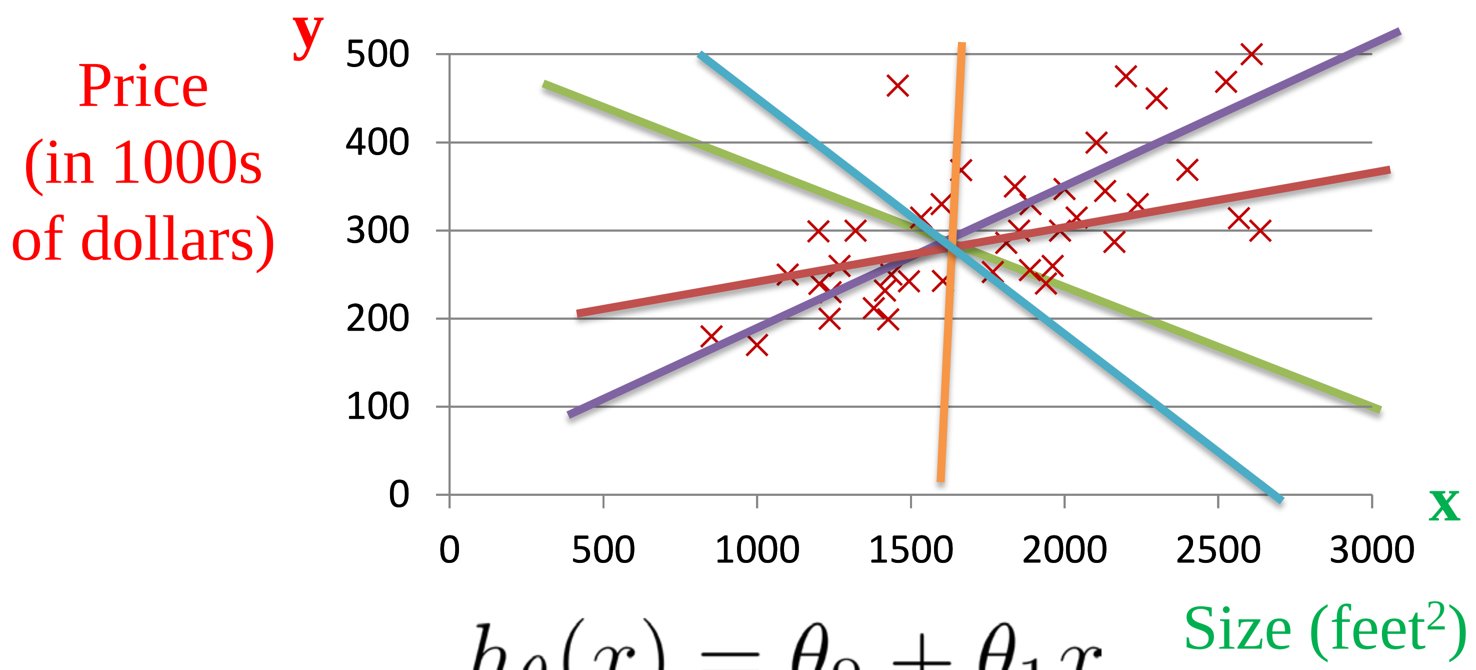
For a hypothesis or linear function,

$$h_\theta(x) = \theta_0 + \theta_1 x$$

we have infinite possibilities to model/fit the given training data. However, which line is the *best* line/hypothesis?



Which line is the *best* line/hypothesis?



$$h_\theta(x) = \theta_0 + \theta_1 x$$

Idea: Choose θ_0, θ_1 so that $h_\theta(x)$ *is close to* y for all the given training examples (x, y)

$h_\theta(x)$: predict value for a given data *x*

y : actual value in training data (*x*, *y*)

Best model is the model that can fit all training data well

Standard Hypothesis Function:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

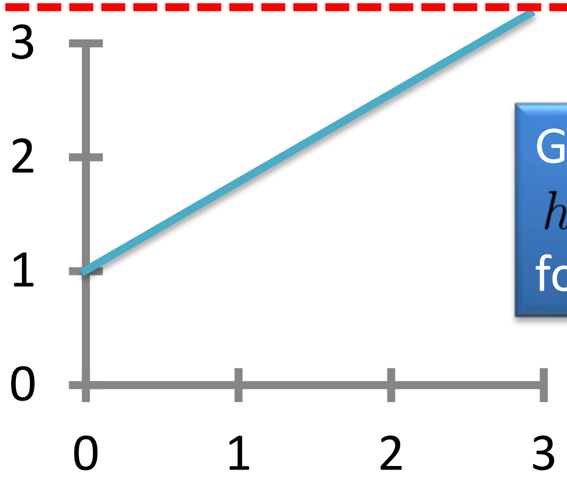
$$\theta_0, \theta_1$$

Cost Function:

$$\theta_0 + \theta_1 x^{(i)}$$

Least-square $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}}\ J(\theta_0, \theta_1)$



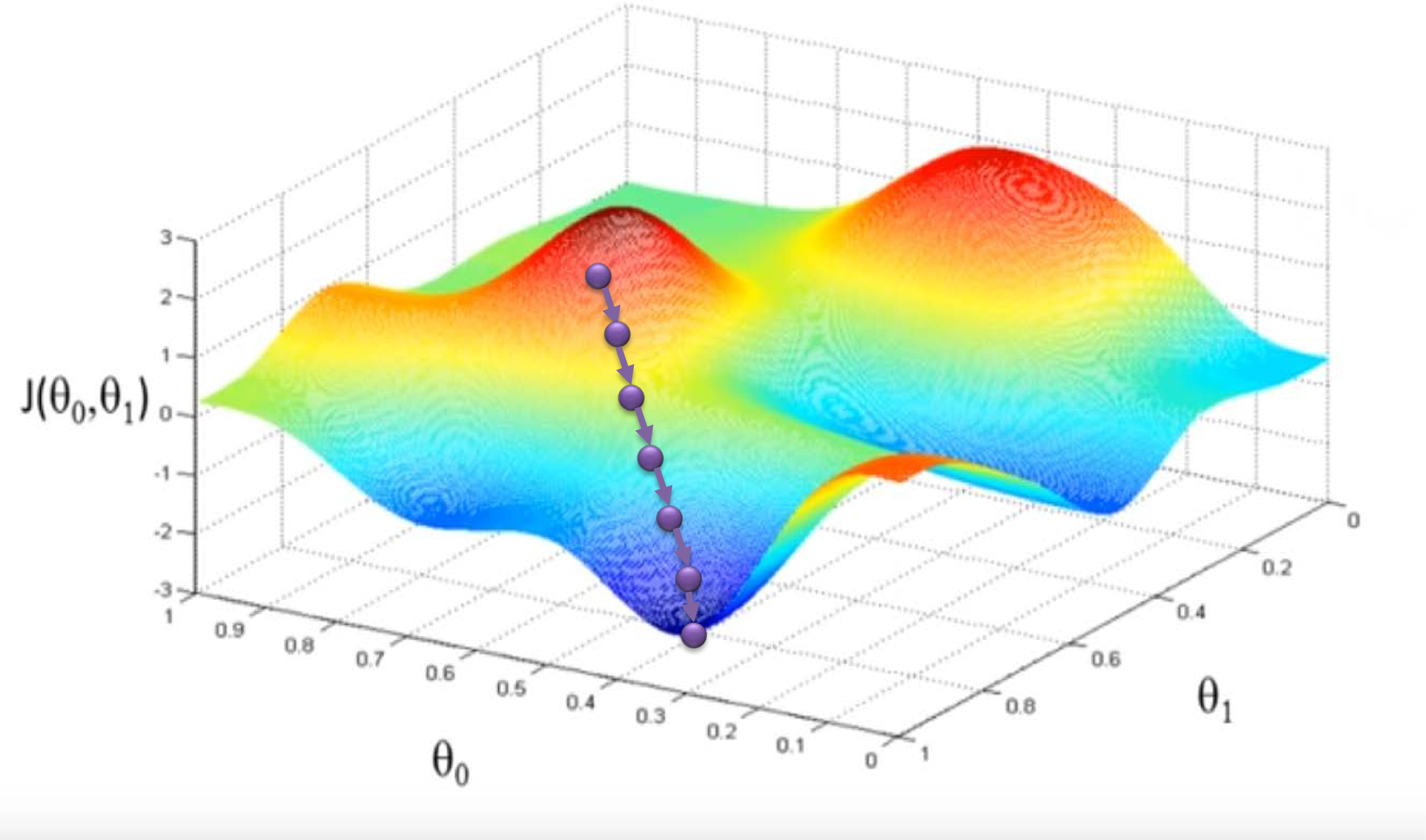
Goal:
$h_\theta(x)$ is close to y
for all training data

Gradient Decent

- **Gradient descent** is an optimization algorithm used to minimize some functions by iteratively moving in the direction of **steepest descent** as defined by the negative of the **gradient**. In machine learning, we use **gradient descent** to update the parameters of our model.
- It is a generic algorithm that is widely used to optimize the objective functions – not only for linear regression problem.

To understand more about gradient decent, pls. watch Andrew Ng's video (11 mins): https://www.youtube.com/watch?v=YovTqTY-PYY

Minimizing the cost function likes a downhill



Outline

- What is Regression
- Evaluation for Regression Models

Evaluating Regression Model (n-test examples)

- Actual target values: $a_1 \quad a_2 \quad \ldots \quad a_n$
- Predicted target values: $p_1 \quad p_2 \quad \ldots \quad p_n$

For dependent variable, we have *n* test examples.

- The *mean-squared error* (**MSE**)

$$\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}$$

$p_i - a_i$ is the error in dimension i

- The *root mean-squared error* (**RMSE**)

$$\sqrt{\frac{(p_1 - a_1)^2 + \ldots + (p_n - a_n)^2}{n}}$$

- The *mean absolute error* (**MAE**) is less sensitive to outliers than the mean-squared error:

$$\frac{|\, p_1 - a_1 \,| + \ldots + |\, p_n - a_n \,|}{n}$$

Evaluating Regression Model (Cont.)

- How much does the scheme improve on *simply predicting the average* (a=(a_1+a_2+… +a_n)/n)?
- The relative absolute error (**RAE**) is:

$$\frac{|p_1 - a_1| + \cdots + |p_n - a_n|}{|a - a_1| + \cdots + |a - a_n|}$$

- The relative squared error (**RSE**) is:

$$\frac{(p_1 - a_1)^2 + \cdots + (p_n - a_n)^2}{(a - a_1)^2 + \cdots + (a - a_n)^2}$$

- R-Squared $\boldsymbol{R^2}$ (the bigger, the better)

$$R^2 = 1 - \frac{(p_1 - a_1)^2 + \cdots + (p_n - a_n)^2}{(a_1 - a)^2 + \cdots + (a_n - a)^2}$$

R-Squared R^2

R-Squared R^2 **:** a measure of how well the data fits the model. It is the ratio of the *variance of the model's error* (or *unexplained variance*), to the total variance of the data:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

The numerator is the sum of squares of residuals (residual sum of squares)
The denominator is the total sum of squares (proportional to the variance of the data; variance will be divided by *n*)

An R^2 of 1 indicates that the regression predictions perfectly fit the data.
Values of R^2 outside the range 0 to 1 can occur when the model fits the data worse than a horizontal line (or hyperplane)

https://en.wikipedia.org/wiki/Coefficient_of_determination

Thank You

Contact: xlli@i2r.a-star.edu.sg if you have questions