COMP4388: **MACHINE LEARNING**

Linear Regression - Part 1

- Linear Regression
- Cost function
- Gradient Descent

Dr. Radi Jarrar
Department of Computer Science

LINEAR REGRESSION MODELS

Linear Regression

- Classification problems are imposed as follows: given a new unlabelled input feature vector to a classification model, predict the class label of this input instance
- The classification output is a class label (***discrete value***)
- Regression problems, on the other hand, predicts a continuous value for an input feature vector

Linear Regression (2)

- Regression has a long history in statistics where it is thoroughly studied
- Linear regression is a parametric model
- The regression learning problem is to learn a function estimator from a set of input examples (x_i, y_i) in which y_i is now a real value
- A function estimator (regressor) is a mapping $f: x \rightarrow R$

Linear Regression (3)

- The process is to make some assumptions about data distributions
- This process is generally created by parametric models (i.e., a statistical model with a fixed number of parameters)

Linear Regression (4)

- For examples:
  - Estimate the prices of stocks given an economic indicator
  - Predict the height given a weight of a person
  - Predict the prices of houses, land, good, …, etc.
  - Setting credit limits for bank customers

Linear Regression (5)

- Given a set of historical data (x_1, y_1), (x_2, y_2), …, (x_n, y_n) where x_n is the information that will be used to build the hypothesis h using feature x to predict a continuous value y
- Input data may not be consistent (values of x), accordingly, the target will not be a deterministic function y=f(x)

Linear Regression (6)

- Indeed, it will be a noisy target formalised as a distribution of the random variable y that comes from different views (i.e., different experts, different scenarios or situations, …)
- That is, the label y_n will result from some distribution $P(y \mid x)$ instead of a deterministic function f(x)

Linear Regression (6)

- The aim is to find a a hypothesis *h* that minimises the error between h(x) and y with respect to the distribution of the data

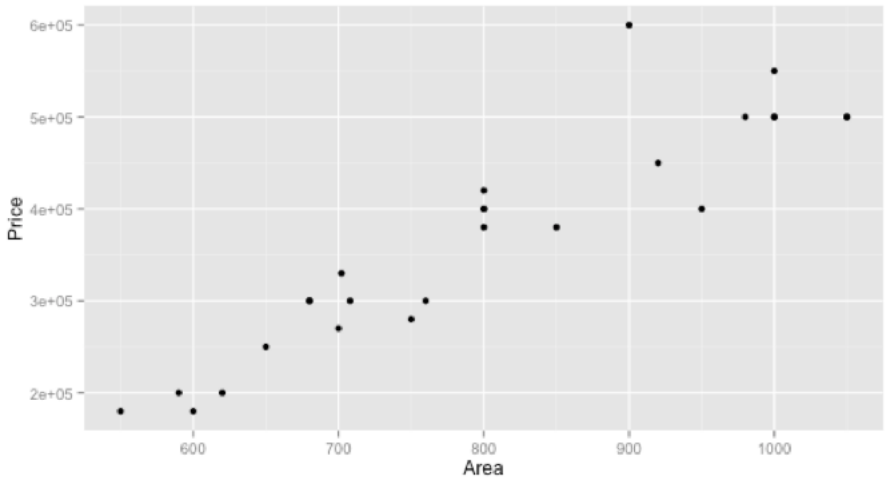
Linear Regression-Advantages

- The most common approach for modelling numeric data
- Provides estimate of the strength & size of the relationships among features & the outcome
- Can be adapted to model (almost) all data

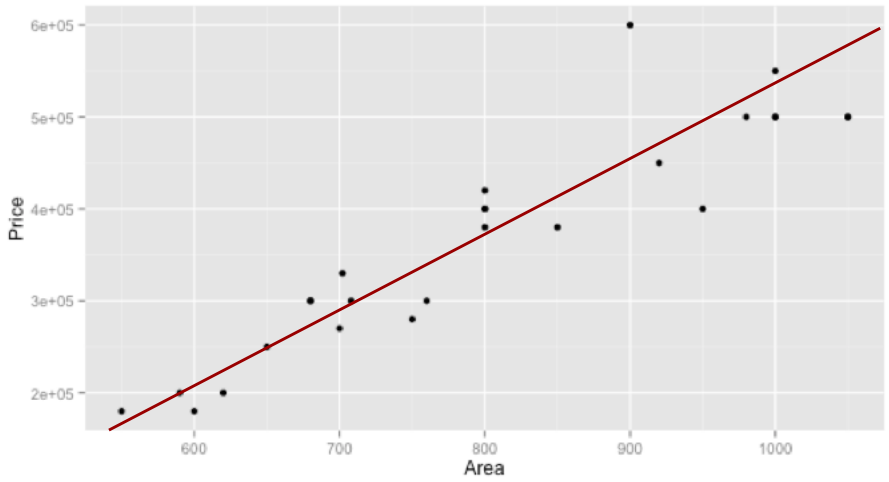
Linear Regression-Disadvantages

- Makes strong assumption about the data
- The form of the model has to be specified by the user in advance
- Does not do well with missing data
- Works with numeric features. Categorical data needs extra processing
- Statistical knowledge (to understand the model) is mainly required

Example-predicting the price of lands



Example-predicting the price of lands



Model representation

- The data of the previous model can be seen as:
  (x, y) = (area, price)
- Given a training set containing the area of lands and their prices, the learning algorithm will use the **areas** of the lands to build a **hypothesis** which will estimate the **price** of a new land given its area
- Which is also seen as: estimate the value of y given x
- The hypothesis *h* maps from *x* to *y*

Model representation (2)

- There is a relation between the area of a land and the prices
- Simple linear regression defines the relationship between a dependent variable and a single independent predictor variable using a line
- The line is denoted as the following equation

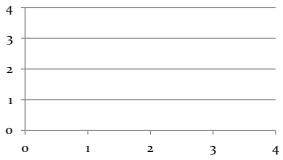
Model representation (3)

$$h(x) = \alpha + \beta x$$

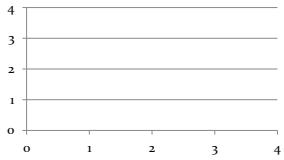
- where α represents where the line crosses the y axis; and the slope β represents the change in y given the increase in x
- Performing the regression analysis involves finding parameter estimates for α and β
- This means that we are predicting that y is a linear function of x

Model representation (4)

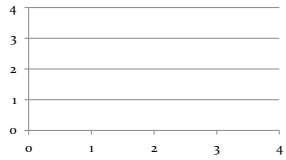
- This is a representation of a univariate linear function (i.e., linear function with one variable)
- How to select the parameters α and β?
- Selecting α and β values will result in different hypothesis of the model



α=0, β=0.5



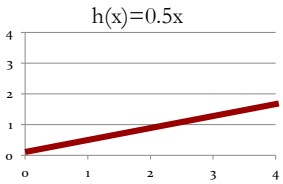
α=1, β=1



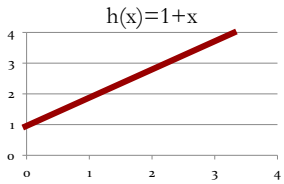
α=2, β=0

Model representation (5)

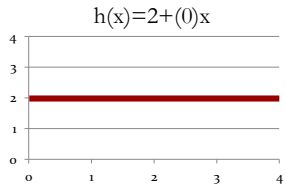
- This is a representation of a univariate linear function (i.e., linear function with one variable)
- How to select the parameters α and β?
- Selecting α and β values will result in different hypothesis of the model



α=0, β=0.5



α=1, β=1



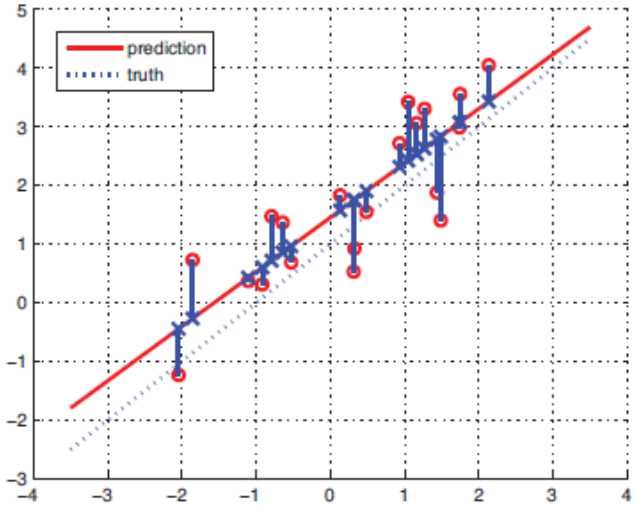
α=2, β=0

Model representation (6)

- We need to find a straight line that fits the data
- This is done by finding values for the parameters α and β
- Idea: Choose α and β so that h(x) is close to y for the input training examples (x, y)
- This can be achieved through a minimisation problem
- This is also known as the Least Squares Approach

Model representation (7)

- Each of these errors is known as a residual



Model representation (8)

$$\min_{\alpha,\beta}(h(x)-y)^2$$

- and for the entire dataset, we take the sum difference of the squared error (i.e., the squared difference between the output of the hypothesis given an input x with the difference to the actual value of y

$$\min_{\alpha,\beta}\sum_{n=1}^{N}(h(x_n)-y_n)^2$$

Model representation (9)

$$\min_{\alpha,\beta}\frac{1}{N}\sum_{n=1}^{N}(h(x_n)-y_n)^2$$

- Min: find the values of α and β that cause the expression to be minimised
- Find the values of α and β so that the average of the sum of the squared error between the predictions of the prices of lands minus the actual data of the training set is minimised
- **This is the objective function for Linear Regression**

Model representation (10)

$$\min_{\alpha,\beta} \frac{1}{N} \sum_{n=1}^{N} (h(x_n) - y_n)^2$$

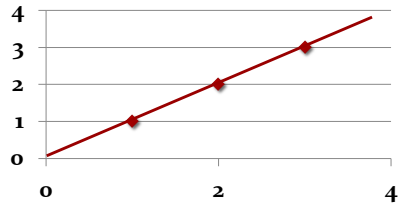
- This is also called the Cost function (Squared Error function)
- Squared because we are considering the square error between the new predictions and the actual data
- Works well for most of the regression problems (most commonly used)

Model representation (11)

- The optimisation objective is to minimise the cost function for the parameters α and β
- Different options of those parameters will end-up in different straight lines
- Plotting the hypothesis and the cost function
  - The hypothesis is a function of x
  - The cost function is a function of α

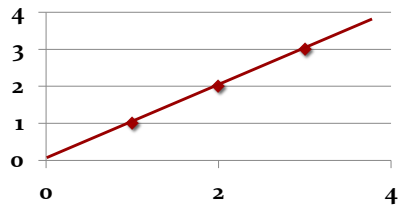
Model representation (12)

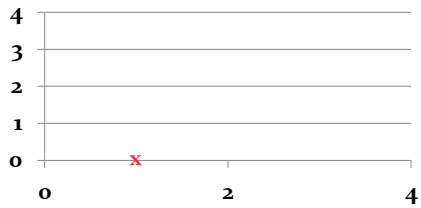
- h(x) = α.x
  Training set (1,1), (2,2), (3,3)
  α = 1 then the hypothesis will be a straight line crossing all data
  - We want to find the value of J (the cost function) when α = 1 (i.e., find J(1))



Model representation (12)

- h(x) = α.x
  Training set (1,1), (2,2), (3,3)
  α = 1 then the hypothesis will be a straight line crossing all data
  - We want to find the value of J (the cost function) when α = 1 (i.e., find J(1))

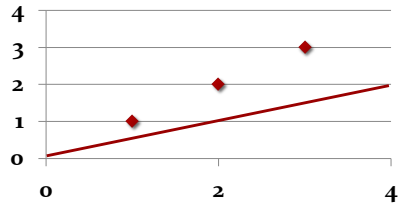




Model representation (13)

- h(x) = α.x
  Training set (1,1), (2,2), (3,3)
  α = 0.5

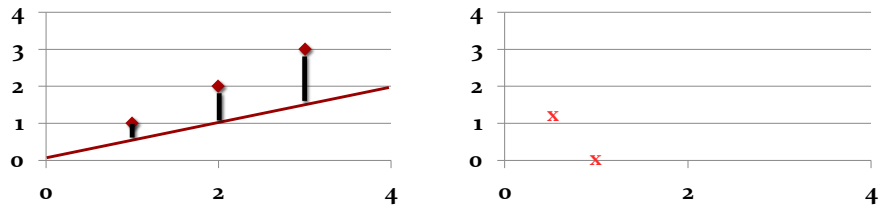
- We want to find the value of J (the cost function) when α = 0.5 (i.e., find J(0.5))



Model representation (13)

- h(x) = α.x
  Training set (1,1), (2,2), (3,3)
  α = 0.5

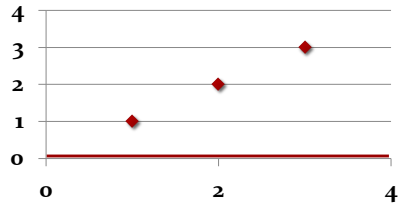
- We want to find the value of J (the cost function) when α = 0.5 (i.e., find J(0.5))



Model representation (14)

- h(x) = α.x
  Training set (1,1), (2,2), (3,3)
  α = 0

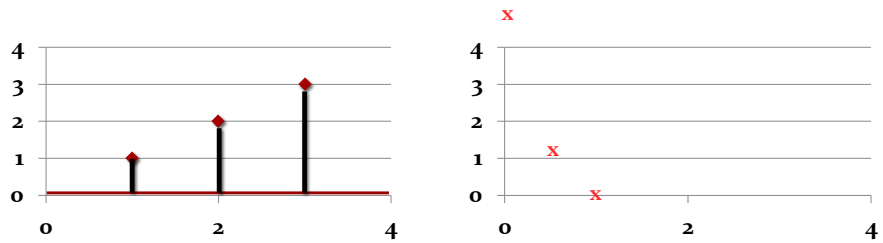
- We want to find the value of J (the cost function) when α = 0 (i.e., find J(0))



Model representation (14)

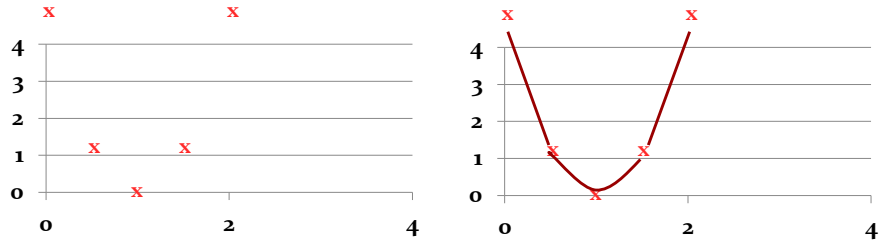
- h(x) = α.x
  Training set (1,1), (2,2), (3,3)
  α = 0

- We want to find the value of J (the cost function) when α = 0 (i.e., find J(0))



Model representation (15)

- Keep computing the errors (i.e., the cost function) with difference values of a it will end up in a concave shape
- Each value of a corresponds to a different hypothesis (i.e., different straight line fit)

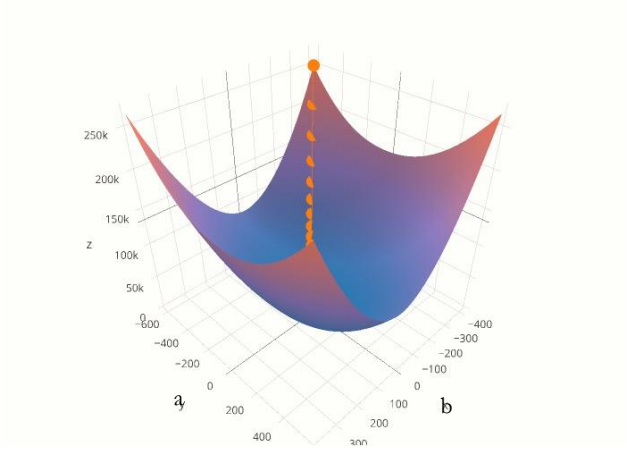


Model representation (16)

- The cost function is considered to be the sum of squared values of the heights (which is the difference between the height of the lines between straight line and the predicted values h(x))
- Notice that the cost function (after computing a range of values) looks like a concave

Model representation (17)

- Note as the number of parameters increase, the dimensionality of the space increases as well



Gradient Descent

- Algorithm to minimise the cost function
- Used in many learning algorithms
- How it works?
  - Start with some values of α and β (or as many parameters as required)
  - Keep on changing the values of the parameters to reduce the cost function (i.e., $J(\alpha,\beta)$) until we end up at the minimum
  - Repeatedly update the parameters **simultaneously** until **convergence**

Gradient Descent (2)

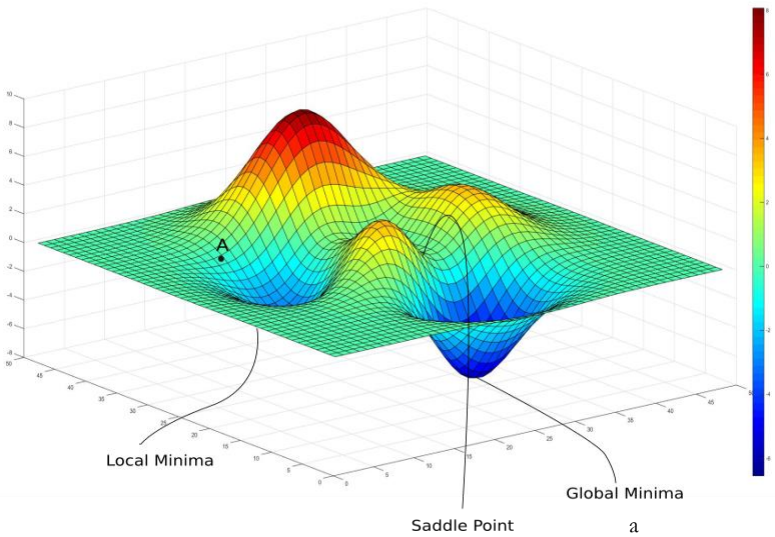
- We need to simultaneously update α and β

$$\alpha_j = \alpha_j - \Phi \frac{\partial}{\partial \alpha_j} J(\alpha_j, \beta_j)$$

$$\beta_j = \beta_j - \Phi \frac{\partial}{\partial \beta_j} J(\alpha_j, \beta_j)$$

- Φ is the learning rate; slow value for Φ will result in slow GD; a large value of Φ may fail to converge to a local minimum

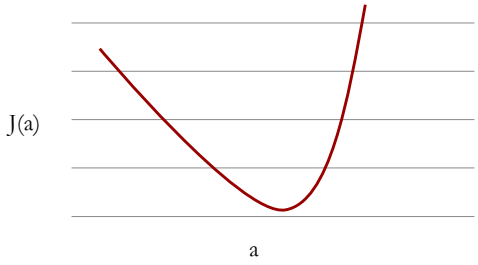
Gradient Descent (3)



From Intro to optimization in deep learning gradient descent/Ayoosh Kathuria

Gradient Descent (4)

- Gradient descent can converge to a local minimum
- As a local minimum is approached, GD will automatically take smaller steps



Example - Bivariate

Height	Weight
160	56
168	95
174	77
177	80
179	67
170	65
174	70
170	107
191	100