Week 1

Introduction

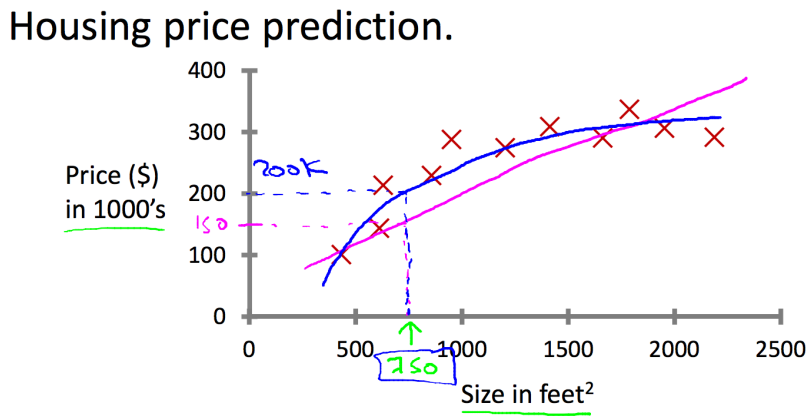
Definition of Machine Learning: A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

-Supervised Learning: Right answer given

Example 1: housing price prediction

Given data about the size of houses on the real estate market, try to predict their price. Price as a function of size is a continuous output, so this is a regression problem.

We could turn this example into a classification problem by instead making our output about whether the house "sells for more or less than the asking price." Here we are classifying the houses based on price into two discrete categories.



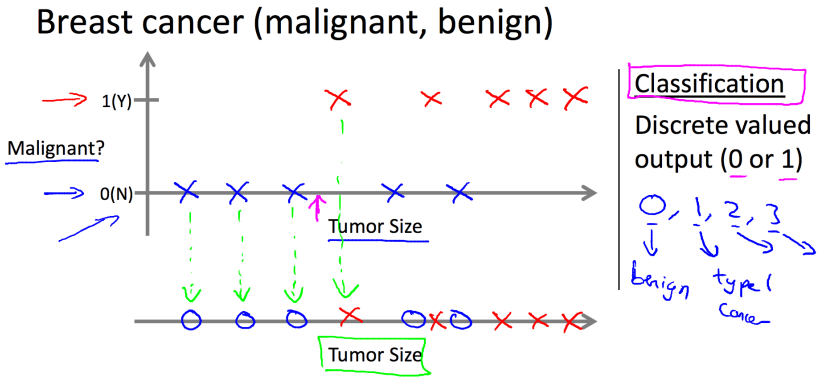
This is a **regression problem**, try to predict **continuous** valued output.

(a) Regression - Given a picture of a person, we have to predict their age on the basis of the given picture

(b) Classification - Given a patient with a tumor, we have to predict whether the tumor is malignant or benign.

Example 2: Breast cancer

Predict if it's a malignant cancer or benign cancer.



This is a **Classification problem**, try to predict **discrete** valued output (0 or 1, but can also be extended to 0,1,2…)

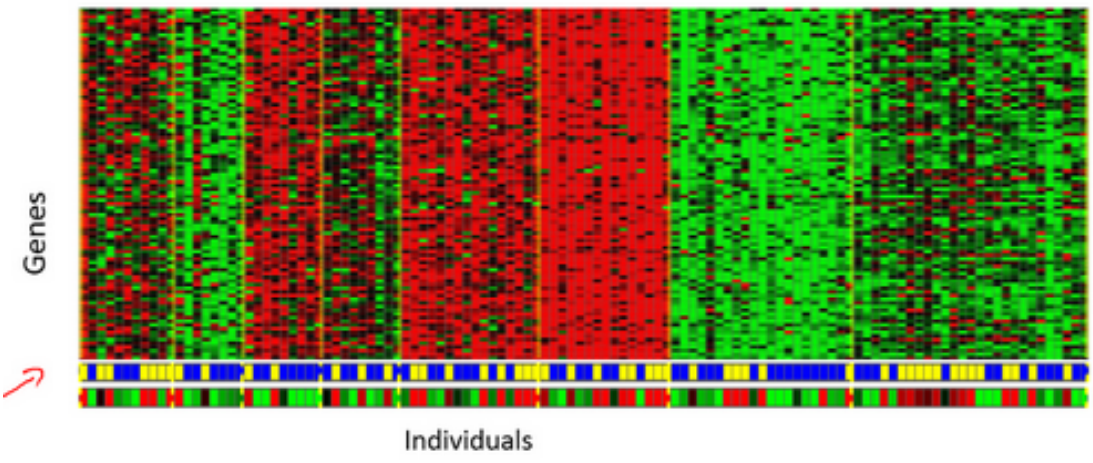
-Unsupervised learning: structure data

Unsupervised learning allows us to approach problems with little or no idea what our results should look like. We can derive structure from data where we don't necessarily know the effect of the variables.

Clustering the data based on relationships among the variables in the data.

Example1: Gene clustering (Cluster)

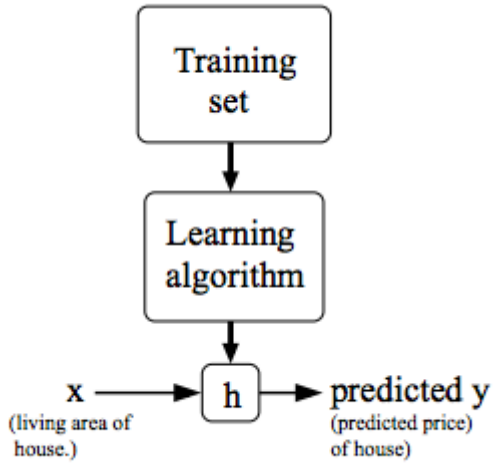
Automatically group genes into groups that are similar or related by different values



Example2: Cocktail Party Algorithm (Non-cluster)

Model and Cost Function

-Model Representation



-Cost Function (squared error function)

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

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\hat{y}_i - y_i \right)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x_i) - y_i \right)^2$$

Cost Function:

Goal: minimize cost function

- Gradient Descent

Gradient descent algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

(for $j = 1$ and $j = 0$)

}

Linear Regression Model

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

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

Simultaneous update:

Correct: Simultaneous update

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := \text{temp0}$

$\theta_1 := \text{temp1}$

Incorrect:

$\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$\theta_0 := \text{temp0}$

$\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

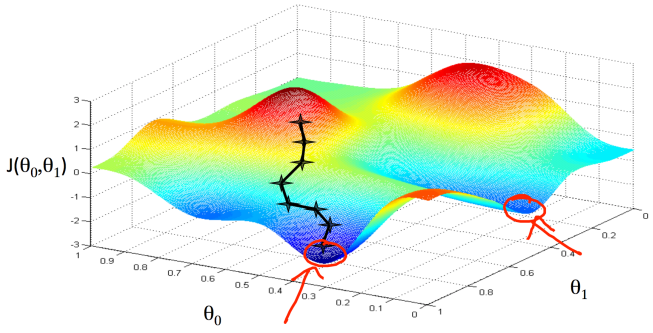
$\theta_1 := \text{temp1}$

Notes: if write temp0 to θ0 first, temp1 will use the new θ0 to calculate θ1.

Selection of α:

If α is too small, gradient descent can be slow.
If α is too large, gradient descent can overshoot the minimum. It can diverge.

Local minimum:

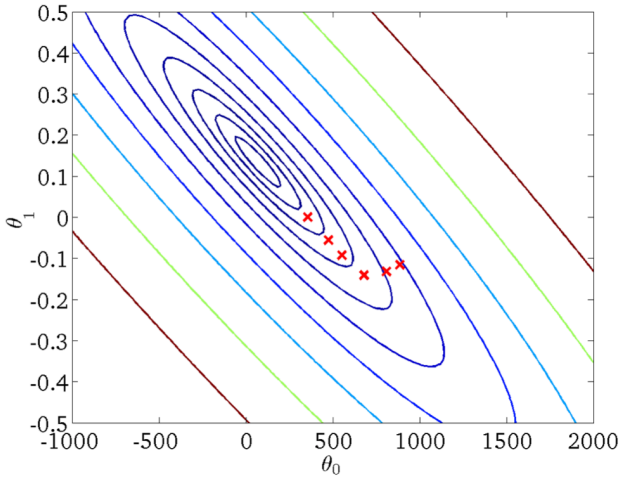


Gradient descent can converge to a local minimum, even with the learning rate α fixed.
As we approach a local minimum, gradient descent will automa6cally take smaller

steps. So, no need to decrease α over time.

Notes: two close start point can lead to different local optima.

Contour plot:



Notes: from start point, approaching the inner most oval.

batch gradient descent:
looks at every example in the entire training set on every step