Standford University

Week1

Machine Learning

-Grew out of work in AI

- New capability for computers

Examples:

-Database mining

Large datasets from growth of automation/web.

e.g. Web click data, medical records, biology, engineering

Applications can’t program by hand.

E.g. Autonomous helicopter, handwriting recognition, most of Natural Language Processing(NLP), Computer Vision

Self-customizing Programs

E.g. Amazon, Netflix product recommendations

Understanding human learning (brain, real AI)

Class on machine learning

Machine learning is one of the most exciting recent technologies.

Here we

Learn state of the art and also gain practice implementing and deploying these algorithms yourself.

Unknowingly, you’re using dozens of learning algorithms in your day-to-day life.

Web search engine uses learning algorithm developed by Google or Microsoft, has learned how to rank web pages.

Facebook or Apple’s photo typing application and it recognizes your friend’s photos, that’s also machine learning

Every time you Read email and your spam filter saves your from having to wade through tons of spam emails, that’s also learning algorithm.

For me one of the reasons I’m excited is the AI dream of someday building machines as intelligent as you or me.

We’re long way away from that goal, but many AI researchers believe that the best way to towards that goal is through learning algorithms that try to mimic how the human brain learns.

I’ll tell you a little bit about that too in this content.

Learn about state-of-the-art machine learning algorithms.

Andrew Ng

What is Machine Learning?

Machine Learning Definition

* Arthur Samuel (1959). Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.

Checkers Players

Here's a definition of what is machine learning as due to Arthur Samuel.

He defined machine learning as the field of study

that gives computers the ability to learn without being explicitly learned.

Samuel's claim to fame was that back in the 1950, he wrote a checkers playing

program and the amazing thing about this checkers playing program

was that Arthur Samuel himself wasn't a very good checkers player.

But what he did was he had to programmed maybe tens of thousands of games against

himself, and by watching what sorts of board positions tended to lead to wins and

what sort of board positions tended to lead to losses,

the checkers playing program learned over time what are good board positions and

what are bad board positions.

And eventually learn to play checkers better than the Arthur Samuel

himself was able to.

This was a remarkable result.

Arthur Samuel himself turns out not to be a very good checkers player.

But because a computer has the patience to play tens of thousands of

games against itself, no human has the patience to play that many games.

By doing this, a computer was able to get so much checkers playing experience

that it eventually became a better checkers player than Arthur himself.

* Tom Mitchell (1998) Well-posed Learning Problem: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.

Suppose your email program watches which emails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

* Classifying emails as spam or not spam. --------🡪T
* Watching you label emails as spam or not spam. -----🡪E
* The number (or fraction) of emails correctly classified as spam/not spam. ---🡪P
* None of the above-this is not a machine learning problem

Absolutely, the correct ans is classifying emails as spam or not spam. --🡪T

Machine Learning Algoritihms:

* Supervised learning
* Unsupervised learning

Others: Reinforcement learning, recommender systems.

Also talk about: Practical advice for applying learning algorithms.

Review: What is Machine Learning?

Two definitions of ML are offered.

Arthur Samuel described it as: “the field of study that gives computers the ability to learn without being explicitly programmed.” This is an older, informal definition.

Tom Mitchell provides a more modern definition: “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.”

Example: playing checkers.

E = the experience of playing many games of checkers.

T = the task of playing checkers.

P = the probability that the program will win the next game.

In general, any machine learning problem can be assigned to one of the two broad classifications:

* Supervised learning
* Unsupervised learning

Supervised learning:

**Supervised Learning**

In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

Supervised learning problems are categorized into "regression" and "classification" problems. In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories.

**Example 1:**

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.

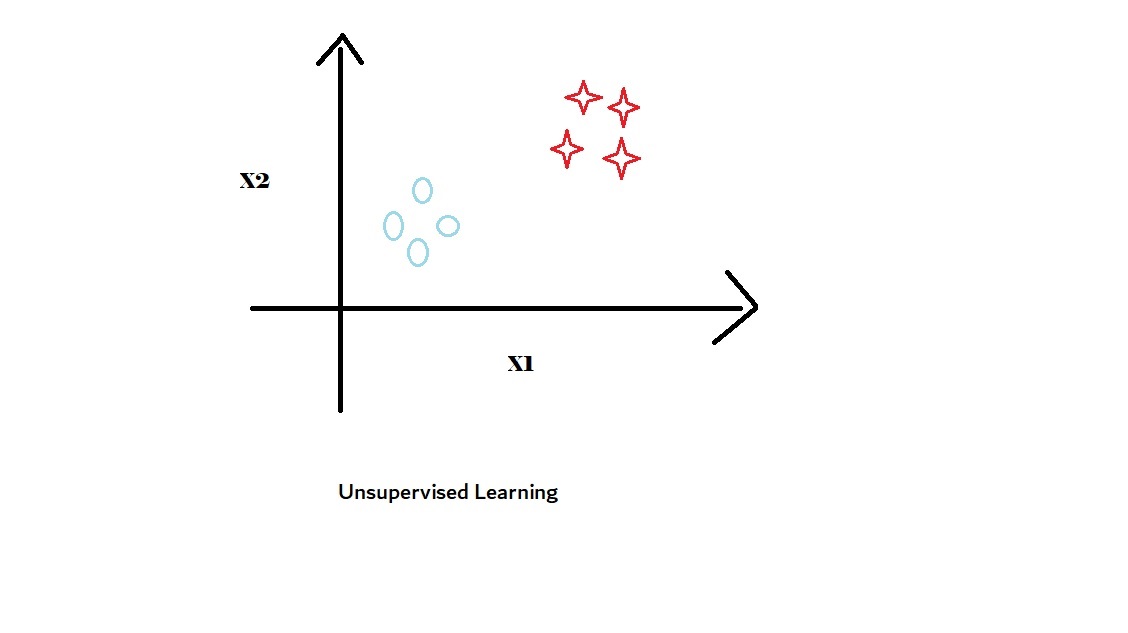
**Example 2**:

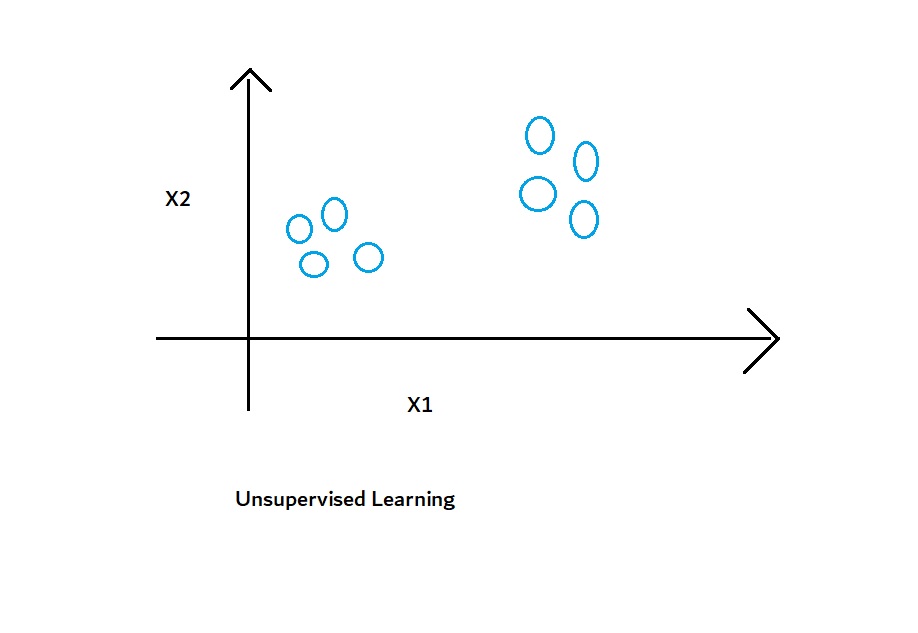
(a) Regression - Given a picture of a person, we have to predict their age on the basis of the given picture

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

Unsupervised Learning

Recall





Cocktail party problem algorithm

[W,s,v] = svd((repmat(sum(x.\*x,1),size(x,1),1).\*x)\*x');

Of the following examples, which would you address using an unsupervised learning algorithm? (Check all that apply.)

1. Given email labeled as spam/not spam, learn a spam filter.
2. Given a set of news articles found on the web, group them into set of articles about the same story.
3. Given a database of customer data, automatically discover market segments and group customers into different market segments.
4. Given a dataset of patients diagnosed as either having diabetes or not, learn to classify new patients as having diabetes or not.

Correct Ans: b and c

## Unsupervised Learning

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.

We can derive this structure by clustering the data based on relationships among the variables in the data.

With unsupervised learning there is no feedback based on the prediction results.

**Example:**

Clustering: Take a collection of 1,000,000 different genes, and find a way to automatically group these genes into groups that are somehow similar or related by different variables, such as lifespan, location, roles, and so on.

Non-clustering: The "Cocktail Party Algorithm", allows you to find structure in a chaotic environment. (i.e. identifying individual voices and music from a mesh of sounds at a [cocktail party](https://en.wikipedia.org/wiki/Cocktail_party_effect)).

# **Who are Mentors?**

**Mentor Program Overview:**

Community Mentors are successful, dedicated Coursera learners who volunteer to assist with support and discussion forum moderation in courses that they have already completed. They have been recruited by Coursera to encourage newer learners, answer questions, set an example by posting thoughtful and timely content, and report platform bugs and inappropriate content to Coursera.

As you use the [discussion areas](https://www.coursera.org/learn/machine-learning/discussions), please be aware that the ideas expressed by participants in this course, including the Mentors, do not represent the views of Stanford University. The mentors are not employed by Stanford University and they have not been vetted by Stanford University as experts on course content or course facilitation.

# Model Representation

To establish notation for future use, we’ll use x^{(i)}*x*(*i*) to denote the “input” variables (living area in this example), also called input features, and y^{(i)}*y*(*i*) to denote the “output” or target variable that we are trying to predict (price). A pair (x^{(i)} , y^{(i)} )(*x*(*i*),*y*(*i*)) is called a training example, and the dataset that we’ll be using to learn—a list of m training examples {(x^{(i)} , y^{(i)} ); i = 1, . . . , m}(*x*(*i*),*y*(*i*));*i*=1,...,*m*—is called a training set. Note that the superscript “(i)” in the notation is simply an index into the training set, and has nothing to do with exponentiation. We will also use X to denote the space of input values, and Y to denote the space of output values. In this example, X = Y = ℝ.

To describe the supervised learning problem slightly more formally, our goal is, given a training set, to learn a function h : X → Y so that h(x) is a “good” predictor for the corresponding value of y. For historical reasons, this function h is called a hypothesis. Seen pictorially, the process is therefore like this:



When the target variable that we’re trying to predict is continuous, such as in our housing example, we call the learning problem a regression problem. When y can take on only a small number of discrete values (such as if, given the living area, we wanted to predict if a dwelling is a house or an apartment, say), we call it a classification problem.

# Cost Function

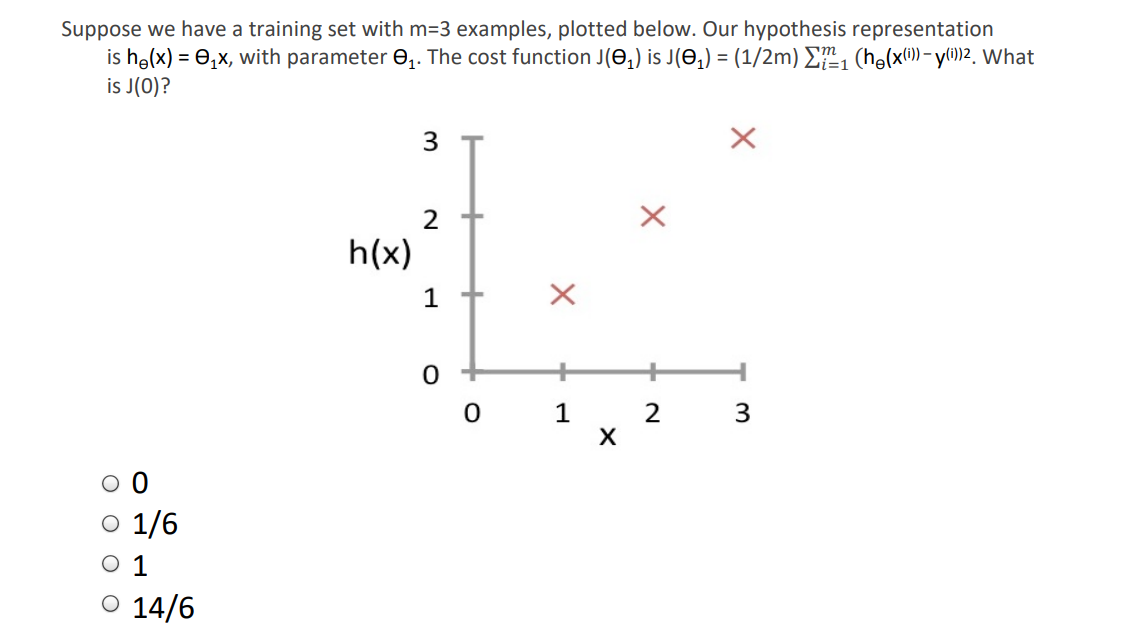
We can measure the accuracy of our hypothesis function by using a **cost function**. This takes an average difference (actually a fancier version of an average) of all the results of the hypothesis with inputs from x's and the actual output y's.

J(\theta\_0, \theta\_1) = \dfrac {1}{2m} \displaystyle \sum \_{i=1}^m \left ( \hat{y}\_{i}- y\_{i} \right)^2 = \dfrac {1}{2m} \displaystyle \sum \_{i=1}^m \left (h\_\theta (x\_{i}) - y\_{i} \right)^2*J*(*θ*0​,*θ*1​)=2*m*1​*i*=1∑*m*​(*y*^​*i*​−*yi*​)2=2*m*1​*i*=1∑*m*​(*hθ*​(*xi*​)−*yi*​)2

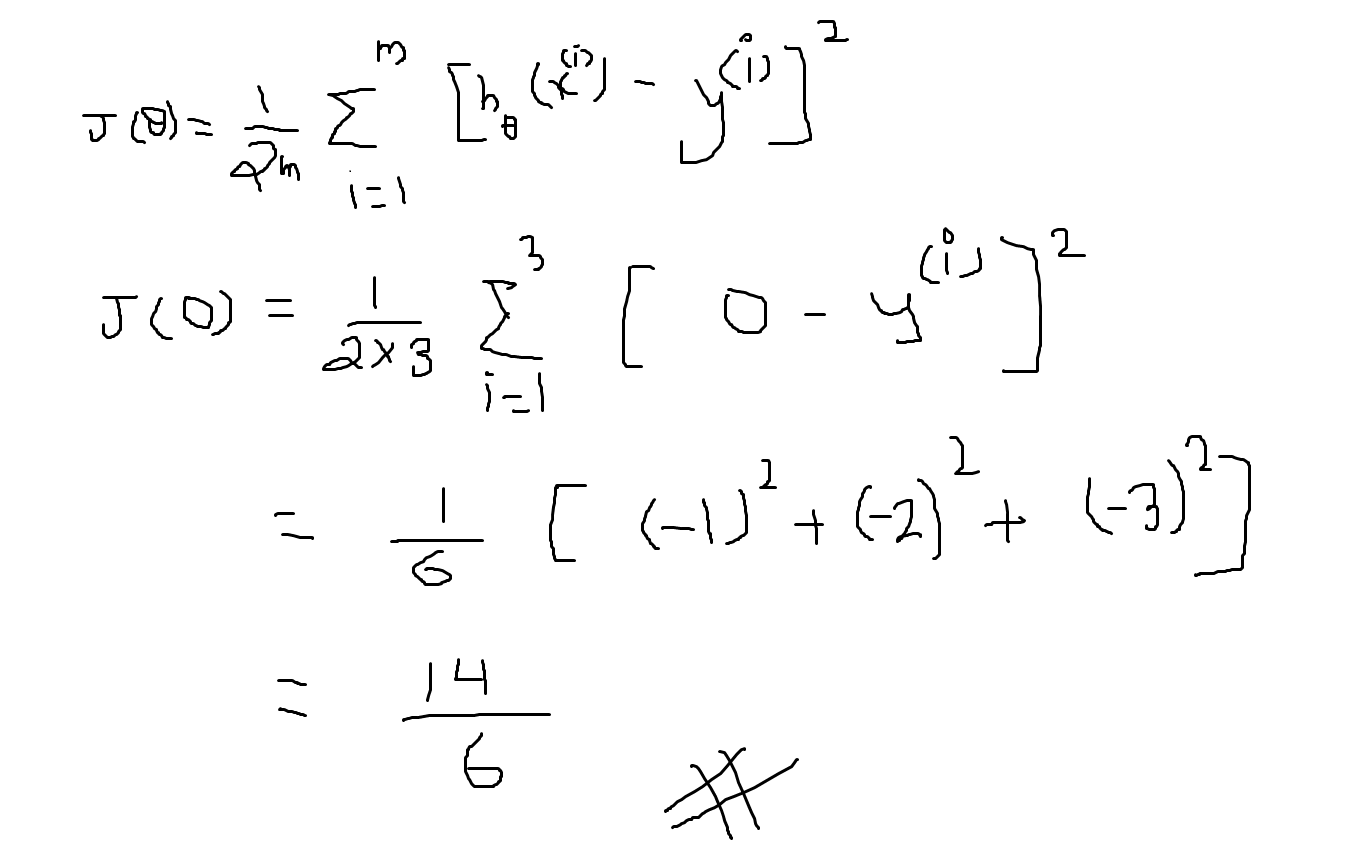
To break it apart, it is \frac{1}{2}21​ \bar{x}*x*ˉ where \bar{x}*x*ˉ is the mean of the squares of h\_\theta (x\_{i}) - y\_{i}*hθ*​(*xi*​)−*yi*​ , or the difference between the predicted value and the actual value.

This function is otherwise called the "Squared error function", or "Mean squared error". The mean is halved \left(\frac{1}{2}\right)(21​) as a convenience for the computation of the gradient descent, as the derivative term of the square function will cancel out the \frac{1}{2}21​ term. The following image summarizes what the cost function does:





Ans: 14/6



# Cost Function - Intuition I

If we try to think of it in visual terms, our training data set is scattered on the x-y plane. We are trying to make a straight line (defined by h\_\theta(x)*hθ*​(*x*)) which passes through these scattered data points.

Our objective is to get the best possible line. The best possible line will be such so that the average squared vertical distances of the scattered points from the line will be the least. Ideally, the line should pass through all the points of our training data set. In such a case, the value of J(\theta\_0, \theta\_1)*J*(*θ*0​,*θ*1​) will be 0. The following example shows the ideal situation where we have a cost function of 0.



When \theta\_1 = 1*θ*1​=1, we get a slope of 1 which goes through every single data point in our model. Conversely, when \theta\_1 = 0.5*θ*1​=0.5, we see the vertical distance from our fit to the data points increase.



This increases our cost function to 0.58. Plotting several other points yields to the following graph:



Thus, as a goal, we should try to minimize the cost function. In this case, \theta\_1 = 1*θ*1​=1 is our global minimum.

# Cost Function - Intuition II

A contour plot is a graph that contains many contour lines. A contour line of a two variable function has a constant value at all points of the same line. An example of such a graph is the one to the right below.



Taking any color and going along the 'circle', one would expect to get the same value of the cost function. For example, the three green points found on the green line above have the same value for J(\theta\_0,\theta\_1)*J*(*θ*0​,*θ*1​) and as a result, they are found along the same line. The circled x displays the value of the cost function for the graph on the left when \theta\_0*θ*0​ = 800 and \theta\_1*θ*1​= -0.15. Taking another h(x) and plotting its contour plot, one gets the following graphs:



When \theta\_0*θ*0​ = 360 and \theta\_1*θ*1​ = 0, the value of J(\theta\_0,\theta\_1)*J*(*θ*0​,*θ*1​) in the contour plot gets closer to the center thus reducing the cost function error. Now giving our hypothesis function a slightly positive slope results in a better fit of the data.



The graph above minimizes the cost function as much as possible and consequently, the result of \theta\_1*θ*1​ and \theta\_0*θ*0​ tend to be around 0.12 and 250 respectively. Plotting those values on our graph to the right seems to put our point in the center of the inner most 'circle'.

Khom