**Course Description**

This course provides a broad introduction to machine learning and statistical pattern recognition. Topics include: supervised learning (generative/discriminative learning, parametric/non-parametric learning, neural networks, support vector machines); unsupervised learning (clustering, dimensionality reduction, kernel methods); learning theory (bias/variance tradeoffs; VC theory; large margins); reinforcement learning and adaptive control. The course will also discuss recent applications of machine learning, such as to robotic control, data mining, autonomous navigation, bioinformatics, speech recognition, and text and web data processing.

**Prerequisites**

Students are expected to have the following background:

* Knowledge of basic computer science principles and skills, at a level sufficient to write a reasonably non-trivial computer program.
* Familiarity with the probability theory. (CS 109 or STATS 116)
* Familiarity with linear algebra (any one of Math 104, Math 113, or CS 205)

All REQUIRED COURSE MATERIALS ON : <http://cs229.stanford.edu/materials.html>

Pathway to Mastering ML : <https://www.analyticsvidhya.com/learning-path-learn-machine-learning/>

------------------------------------------------------------------------------------------------------------------------------------------

DAY 1

Hello guys. Roshan here, today I am gonna talk about the ML course provided by Stanford Uni (Andrew NG). So lets do it. Isn’t ML the most interesting topic? ML has large impact in our day to day lives including economy, science, engineering, health etc.

Q. What is Machine Learning?

* 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 .

Syllabus :

1. **Supervised learning.**

Intro to Machine Learning   
 Supervised learning setup. LMS.  
 Logistic regression. Perceptron. Exponential family.   
  Generative learning algorithms. Gaussian discriminant analysis. Naive Bayes.   
 Support vector machines.   
 Model selection and feature selection.   
 Ensemble methods: Bagging, boosting.   
  Evaluating and debugging learning algorithms.

1. **Learning theory.**

Bias/variance tradeoff. Union and Chernoff/Hoeffding bounds.   
VC dimension. Worst case (online) learning.   
Practical advice on how to use learning algorithms.

1. **Unsupervised learning.**

Clustering. K-means.   
EM. Mixture of Gaussians.   
Factor analysis.   
PCA (Principal components analysis).   
ICA (Independent components analysis).

1. **Reinforcement learning and control.**

MDPs. Bellman equations.   
Value iteration and policy iteration.   
Linear quadratic regulation (LQR). LQG.   
Q-learning. Value function approximation.   
Policy search. Reinforce. POMDPs.

Let’s Start then with some general idea about each of above topics ……………….

* **Supervised Learning**

**Supervised learning** is the [machine learning](https://en.wikipedia.org/wiki/Machine_learning) task of inferring a function from labeled training data.[[1]](https://en.wikipedia.org/wiki/Supervised_learning#cite_note-1) The [training data](https://en.wikipedia.org/wiki/Training_set) consist of a set of *training examples*. In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

**Classes of Supervised Learning Problems:**

* **Regression Problem( Continuous Variables)**

Output can be guessed using two continuous variables range in regression problem. Example would be you selling a house in bay area, but you don’t know the actual selling price. So you can feed your algorithm a set of inputs including the price and specifications of previous houses sold, which acts as generalized formula generator to predict other output.

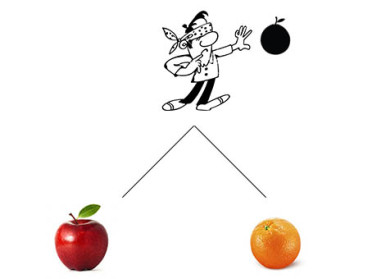
* **Classification Problem( Discrete Variables)**

Tumor can either be Malignant or Benign (harmless). So such output can be guessed using two Discrete variables 1 or 0. 1 means Malignant here. Such problems come under classification problem. In [machine learning](https://en.wikipedia.org/wiki/Machine_learning) and [statistics](https://en.wikipedia.org/wiki/Statistics), **classification** is the problem of identifying to which of a set of [categories](https://en.wikipedia.org/wiki/Categorical_data) (sub-populations) a new [observation](https://en.wikipedia.org/wiki/Observation) belongs, on the basis of a [training set](https://en.wikipedia.org/wiki/Training_set) of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into ["spam" or "non-spam"](https://en.wikipedia.org/wiki/Spam_filtering) classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of [pattern recognition](https://en.wikipedia.org/wiki/Pattern_recognition).

In the terminology of machine learning,[[1]](https://en.wikipedia.org/wiki/Statistical_classification#cite_note-1) classification is considered an instance of [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning), i.e. learning where a training set of correctly identified observations is available.

**Example of Regression vs Classification :**

Let’s understand what classification is with the help of a small exercise. How do you or I recognize a particular fruit, for example say,  an orange or an apple? For this experiment, assume that you are blindfolded and allowed to touch the fruit.

What is the first thing you would do? Hold the fruit and try to figure out the shape of it. If it isn’t conical or has a roughly spherical or oval shape, you can cut down on a lot of options like the banana or the pineapple. The next best thing you can do is, to compare the size of the fruit with the size of your palm. If the fruit fits your palm exactly, you can drop the options of it being a cherry or a grape. What would you do next? Can you get some information using the texture of the surface of the fruit? If the surface is relatively smooth, it can be an apple or a guava. If it’s rough, it can be an orange or an avocado. Is this information sufficient to tell if the fruit is an apple or an orange? No, we would obviously need some more information. Let’s say that you are allowed to taste a small piece of the fruit. If it tastes sweet, there is a chance the fruit is an apple and if it’s sour, it might be an orange. Even this information along with the past knowledge of the shape and texture might not be sufficient sometimes to differentiate between them. Now if the color of the fruit is revealed, you can confirm the fruit with some confidence but then again, there are apples which are green. Finally when you see the fruit, you can safely say if it’s an apple or an orange or neither. One might ask, why were you asked to think about this experiment in the first place? Would you be able to guess the fruit without all the information you gathered using touch and taste? Each experience gave you some additional information about the nature of the fruit. This is how we as humans perceive objects around us. We try to compare the new information obtained now, to our past experiences and if there is match, we try to group them together. If there is no match found, we add it as a new set of information for future comparisons. To keep it simple, this is how an apple/orange is understood by our brain. These small sets of so called experiences are called “**features**” in machine learning language. These features help us understand the nature of the fruit (can be extended to other objects too) under observation. The types of fruits i.e., oranges and apples here, are referred to as classes (2 in this case but can be more than that).

This (apple vs orange) is a simple example of a **binary classification problem** i.e. given a set of “features” classify the fruit into one of the two classes. The next question to ask is, can you do this entire experiment without having a previous experience of touching or tasting an orange or apple (or both)? Not quite possible right? If you didn’t see or touch an apple (or orange) previously in your life, it is not possible to classify the fruit as an apple (or an orange). So the previous experiences with the fruits you have, becomes very important before thinking about classification.

The branch of machine learning is entirely based on this premise. Let us break down the term machine learning. You are helping the machine learn something, just like you teach a kid. How is it possible? The simplest way is to give the machine enough examples (in this case, experiences with apples and oranges) so that when it encounters a new fruit it tries to use its previous experience to understand if the fruit is apple/orange. In a way you are telling the machine that, if the fruit is red/green and tastes sweet it might be an apple and if it’s sour and orange in color, it is an orange. So you have played the role of a “teacher” or a “**supervisor**” by telling the machine each time that the particular experience is caused by an apple or an orange. Since there is a supervisor who tells the machine that the experience corresponds to one of the classes (orange or apple), this is called **Supervised Learning**. It might sometimes happen that you share your experiences (smooth, tastes sweet, red in color etc.) with the machine but don’t provide any information about the class (orange/apple) of the fruit and expect the machine to learn on its own. This kind of technique is called **Unsupervised Learning**.  Unsupervised because you are not really telling the machine which fruit it is, but just providing it with some experiences (features).The video you saw in the beginning of the post uses Supervised learning to train the car to drive on its own(You might have heard the speaker say, the driver drives the car first for a few seconds and ALVINN observes what the driver is doing). It is like your father teaching you how to ride a cycle or a bike. You first observe what he is doing and then try to do the same. Simple isn’t it?

There are other types of learning paradigms other than supervised and unsupervised learning, which we would deal in future articles.

The experiences you had with these fruits, which you shared with the machine, is usually called **training data**. Using the training data, the machine develops rules to classify a new experience to one of the classes (apple or orange). But will the machine be able to differentiate between apple and oranges if the training data given to it only has experiences with apple (or oranges but not both). No, it should also see or feel how an orange looks and tastes like to differentiate it from an apple. So a valid training dataset would have experiences of both apples and oranges. The new experiences on which the machine is evaluated is called the testing data. So the testing dataset would only have the features of the fruits but not the labels. All this said, classification can shortly be described as, given valid training data, classify a testing sample to one of the available classes.

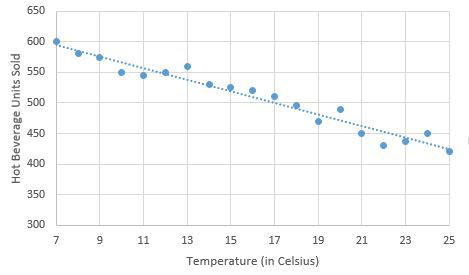
Wouldn’t it be nice if everything is so ideal and as simple as explained above? The world is a cruel place and no problem you encounter in real life is as easy as explained above. We would explain how the training data affects the learning and how the classification rules are automatically generated from the training data in future articles. Now let’s move onto the next section of the article where the other kind of machine learning technique i.e. regression is discussed.

## ****Regression****

Now that you have a fair understanding of what classification is,  let us explore the other kind of problems where machine learning can be applied. Given that the chilly winter season has now set in, let’s start by taking an example of a hot chocolate serving outlet my friend has opened recently. As you might have guessed, his sales would increase if the air temperature falls down since people would yearn for a hot drink in the freezing conditions. But as every other person who is new to business, he made a rookie mistake. He overestimated the demand initially and since the stuff that goes into making the different beverages are perishable, he suffered from losses. So, he started looking for ways to forecast the demand approximately to avoid the losses. He started by collecting the data of the sales of different items on the menu and the temperature of that day.

Now that he has the relevant data, his next step would be to forecast the sales, for say tomorrow, given the temperature estimates from the weather channel. This kind of problem falls under the second category of machine learning called “**Regression**”.  So basically, regression is the estimation of a continuous output variable from a series of other changing (mostly independent) variables. As you can tell, it’s quite different from classification in terms of the nature of the dependent variable. In a classification problem, like the one discussed earlier, the number of classes were limited (ex: orange, apple, banana etc.). In case of regression, the dependent variable (output variable), to be predicted, is generally continuous in nature (ex: the sales for the next day which can have a broad range from 0 to say 10,000).

Now that you have an idea of what regression is, the easiest and most intuitive way to approach this problem is by plotting the data points and examining the relationship between the dependent and the independent variables. For this example, we get the below scatter plot by using the historical sales data.



From the plot, it can be seen that the sales exhibit a **linear** relationship with the temperature. So we can use a straight line equation, given below, to model the relationship between the sales and the temperature.

**y = a\*x + c**

Substituting the respective terms, we have:

**Sales = a \* Temperature + c**

The above formulation is termed as **Simple Linear Regression** and by intuition, it represents the straight line which best fits the historical data points. ‘Simple’ because there is only one independent variable (temperature in this case) used to estimate the dependent variable and ‘linear’ because we assume a linear relationship between sales and temperature.

Let us take a closer look at the above equation. ‘a’ is called the **regression coefficient** for the temperature variable and is the amount by which the sales will change for a unit increase in temperature. By intuition, ‘a’ should be negative since sales increase if the temperature falls. ‘c’ is called the intercept (like in a straight line equation) and can be interpreted as the sales that would take place when the temperature is zero. The intercept can also be interpreted as the constant part of the sales which cannot be explained by the variance in temperature.

As you would have probably guessed, the sales on a particular day would not only depend on the temperature but also on various other factors such as discounts offered, whether it is a holiday, availability of special items etc. These factors can be easily incorporated into the model as shown below.

**Sales = a1 \* Temperature+a2 \* Discount+a3 \* isHoliday+a4 \* SpcItemsPresent+ c**

The above formulation is termed as **multiple linear regression** and by intuition, it represents the plane which best fits the historical data points. ‘Multiple’ because there are now more than one independent variable used to estimate the dependent variable and ‘linear’ because the relationship between the dependent and each of the independent variable is still assumed to be linear.

Here, the terms ‘a1’, ‘a2’, ‘a3’ and ‘a4’ are the regression coefficients of the respective independent variables and c is the intercept. The variable “isHoliday” and “SpclItemsPresent” are indicator variables i.e. they take only two values, 0 or 1. So if it’s a holiday, the “Holiday” variable becomes 1 and contributes to the sales by an amount “a2”. “SpclItemsPresent” also behaves similarly as the “isHoliday” variable

Now that the formulations have been done, the next step involves the estimation of  the unknown terms, ‘a1’,’a2’,’a3’,’a4’ and ‘c’, in the above equation. This is where the machine learning would come into picture. But before we get into the estimation part, let us look at the purpose of this estimation process again. We want a system which gives a reliable forecast of the sales given the set of observable independent variable. So the ideal values of unknown terms in the above equation would be those which would give the most accurate forecasts. To put it in another way, we would like to choose those values for the unknown terms which would reduce the average forecast error. Now you would have understood why I wanted to re-iterate over the objective. In the next article, we shall discuss about how the coefficients can be chosen to minimize this forecast error.

Interestingly, a regression problem can be converted into a classification problem, if the dependent variable can be segmented into bins. Let me break that up for you. Let us assume that our friend is only interested in finding if the sales are below 1000, between 1000-10000 or above 10000. Now, we can impose these conditions on our sales data and create a new variable which takes the value of 1, 2 and 3 respectively if the respective conditions are satisfied. Now that we have a reduced number of classes, we can treat this as a classification problem. But there is a downfall to it. As you might have noticed, when we are binning the sales variable to create a categorical variable, we lost granular information about the sales. We can now just know when the sales are below 1000, between 1000-10000 or above 10000 but can’t get the exact volume of the sales.

So to summarize, classification is the task of assigning an object to one of the classes based upon it’s features and  regression is the problem of finding a line/plane which best fits the given data points.

* **Learning Theory**

In [computer science](https://en.wikipedia.org/wiki/Computer_science), **computational learning theory** (or just **learning theory**) is a subfield of [Artificial Intelligence](https://en.wikipedia.org/wiki/Artificial_Intelligence) devoted to studying the design and analysis of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) algorithms

* **Unsupervised Learning**

**Unsupervised learning** is a type of machine **learning** algorithm used to draw inferences from datasets consisting of input data without labeled responses. The most common **unsupervised learning** method is cluster analysis, which is used for exploratory data analysis to find hidden patterns or grouping in data.

**Unsupervised machine learning** is the [machine learning](https://en.wikipedia.org/wiki/Machine_learning) task of inferring a function to describe hidden structure from "unlabeled" data (a classification or categorization is not included in the observations). Since the examples given to the learner are unlabeled, there is no objective evaluation of the accuracy of the structure that is output by the relevant algorithm—which is one way of distinguishing unsupervised learning from [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) .  
Unsupervised learning can be implemented in pattern recognition and sound analysis . Example: Andrew Ng’s experiment of (Uno,Dos,Tres…. + One,Two,Three = separate countings) . Also images pixel clustering is an example of UL where cluster of pixels are drawn as per the images provided . Note: That sound analysis part can be done using Matlab .

* **Reinforcement Learning**

Reinforcement learning , unlike one decision taken in supervised learning , here to get a job done, a series of good or bad decisions are made which is then learnt by the machine to maximize the positives thereby minimizing the negatives and getting the result perfectly ! Eg : Training a dog requires both kind of Dog’s decision so that you give rewards and punishments accordingly and dog will learn that also accordingly taking its time which is Reinforcement Learning .

**Wiki says: Reinforcement learning** is an area of [machine learning](https://en.wikipedia.org/wiki/Machine_learning) inspired by [behaviorist psychology](https://en.wikipedia.org/wiki/Behaviorism), concerned with how [software agents](https://en.wikipedia.org/wiki/Software_agent) ought to take [*actions*](https://en.wikipedia.org/wiki/Action_selection) in an *environment* so as to maximize some notion of cumulative *reward*.

DAY 2

Refer : <http://cs229.stanford.edu/notes/cs229-notes1.pdf> for the Notes !

Welcome back. Today we gonna continue to talk about Supervised learning Algorithms. So let’s start.

|  |  |
| --- | --- |
| Living area(Square feet) | Price(1000$) |
| 2104 | 400 |
| 1600 | 330 |
| 2400 | 369 |
| 1416 | 232 |
| …. | … |
| …. | … |

Let’s start by talking about a few examples of supervised learning problems. Suppose we have a dataset giving the living areas and prices of 47 houses from Portland, Oregon:

Given data like this, how can we learn to predict the prices of other houses in Portland, as a function of the size of their living areas?

Let,

X(i) = “input” variables (living area in this example)/input features

y(i) = “output”/target variable that we are trying to predict (price).

A pair ( 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}—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. For historical reasons, this function h is called a **HYPOTHESIS**.

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**.

For ease, Let me use Linear representation to the Hypothesis. i.e.

h(x) = Ax+B --------🡪 Linear Equation on x (input)

**Part I Linear Regression**

To make our housing example more interesting, let’s consider a slightly richer dataset in which we also know the number of bedrooms in each house:

|  |  |  |
| --- | --- | --- |
| Living area(Square feet) | #bedrooms | Price(1000$) |
| 2104 | 3 | 400 |
| 1600 | 3 | 330 |
| 2400 | 2 | 369 |
| 1416 | 1 | 232 |
| …. | … |  |
| …. | … |  |
| …. | … |  |

Here, the x’s are two-dimensional vectors in R^2 . For instance, x1^(i) is the living area of the i-th house in the training set, and x2(i) is its number of bedrooms. (In general, when designing a learning problem, it will be up to you to decide what features to choose, so if you are out in Portland gathering housing data, you might also decide to include other features such as whether each house has a fireplace, the number of bathrooms, and so on. We’ll say more about feature selection later, but for now let’s take the features as given.) To perform supervised learning, we must decide how we’re going to represent functions/hypotheses h in a computer. As an initial choice, let’s say we decide to approximate y as a linear function of x: **hθ(x) = θ0 + θ1.x1 + θ2.x2**

Here, the **θi ’s are the parameters** (also called weights) parameterizing the space of linear functions mapping from X to Y. When there is no risk of confusion, we will drop the θ subscript in hθ(x), and write it more simply as h(x). To simplify our notation, we also introduce the convention of letting **x0 = 1** (this is the intercept term), so that

**hθ(x) = θ0.x0 + θ1.x1 + θ2.x2** AND

**h(x) =**

where on the right-hand side above we are viewing θ and x both as vectors, and here n is the number of input variables (not counting x0).

NOTE :

The thing is , Gradient Descent is a type of algorithm used for Minimizing a function .So we use it to minimize the Cost function . Now if you ask what really is the Cost function : A cost function is something you want to minimize. For example, your cost function might be the sum of squared errors over your training set. Gradient descent is a method for finding the minimum of a function of multiple variables. So you can use gradient descent to minimize your cost function.

Now check this for further understanding:

<https://spin.atomicobject.com/2014/06/24/gradient-descent-linear-regression/>

**COST FUNCTION**

Now, given a training set, how do we pick, or learn, the parameters θ? One reasonable method seems to be to make h(x) close to y, at least for the training examples we have. To formalize this, we will define a function that measures, for each value of the θ’s, how close the h(x (i) )’s are to the corresponding y (i) ’s. We define the cost function:

J(θ) =

If you’ve seen linear regression before, you may recognize this as the familiar least-squares cost function that gives rise to the ordinary least squares regression model. Whether or not you have seen it previously, let’s keep going, and we’ll eventually show this to be a special case of a much broader family of algorithms.