**CS401 Machine Learning**

Lecture 11 *03-11-2014*

Last Week: Probability Methods were covered.

x y y hat

f w

noise

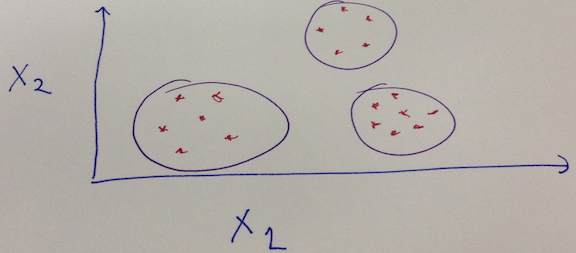
x: input

y: output

yhat: output + noise = true data set

Tried to find w that made out best y.

**K-Means Algorithm:**



***k*-means clustering** is a method of [vector quantization](http://en.wikipedia.org/wiki/Vector_quantization), that is popular for [cluster analysis](http://en.wikipedia.org/wiki/Cluster_analysis) in [data mining](http://en.wikipedia.org/wiki/Data_mining). *k*-means clustering aims to [partition](http://en.wikipedia.org/wiki/Partition_of_a_set) *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest [mean](http://en.wikipedia.org/wiki/Mean), serving as a [prototype](http://en.wikipedia.org/wiki/Prototype) of the cluster. ~Wikipedia

* A clustering algorithm.
* We have k clusters of data, we find points near the centre and adjust our centre iteratively until we find the best centre. Centre of lumps would be our clustered centre.

x

Generator

w

Generator parameterized by w, uses some probability distribution to output x.

X ~ p(x;w) *//Probability of x parameterized by w.*

Use Bayes Rule: To find w to max most probable w.

Formula:

max P (Model | Data)

w

max P (Data | Model ) P(Model)

w P(Data)

max P (Data | Model ) ~~P(Model)~~\*

w ~~P(Data)~~\* \**We ignore these as they’re constants*

max π

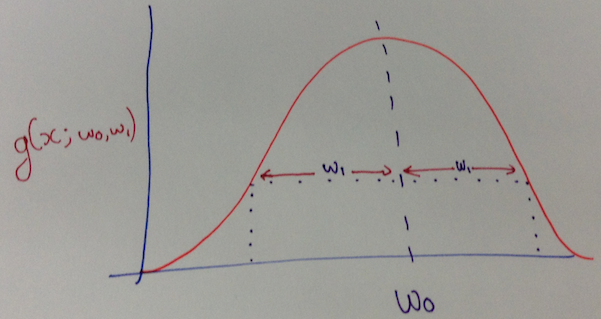
w i P( x (i) | Model)

max ∑

w i P (xi,w)

*Example:* Gaussian Distribution, w is giving us the mean

p(x;w) = g(x;w0,w1) where w0 = Mean, w1 = Standard Deviation



1 . e -½((x-w0)/w1)

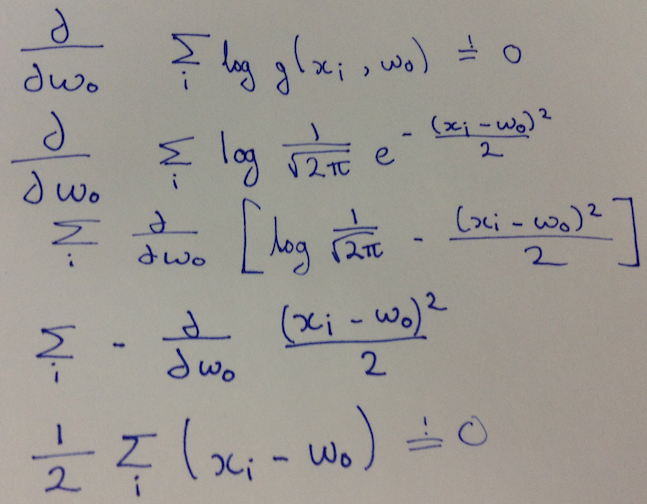
2π w12

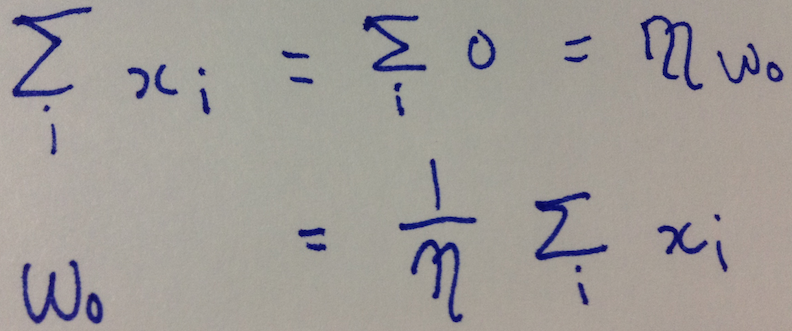
1 . e -((x-w0)2/2)

2π \* *\*Random constant to integrate to 1.*

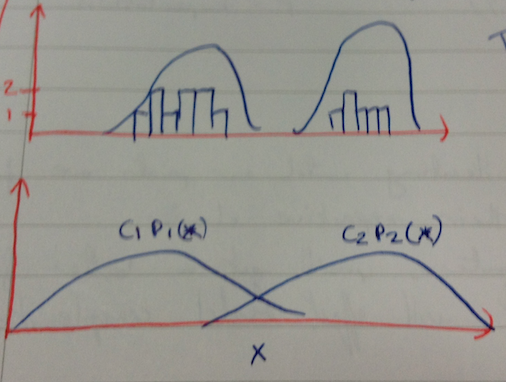
**Gibbs Equation:** P(x) = ½ e –e/t  *i.e. high energy makes it less probable.*

To min towards 0:





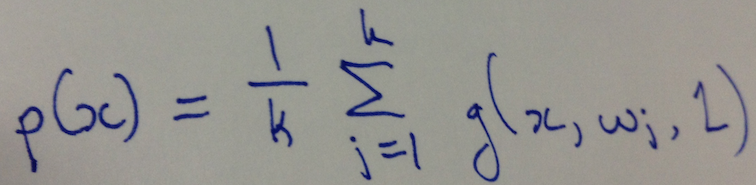
**Mixture Models:**

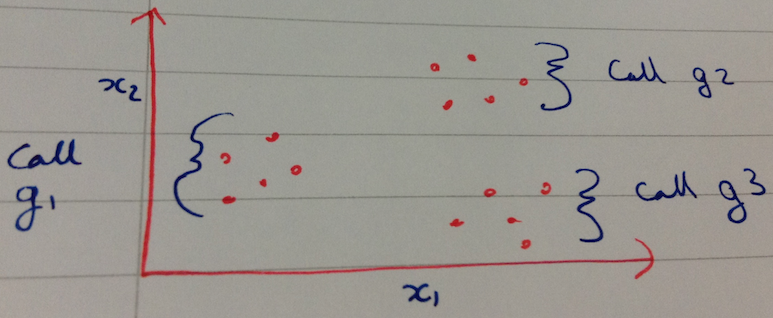


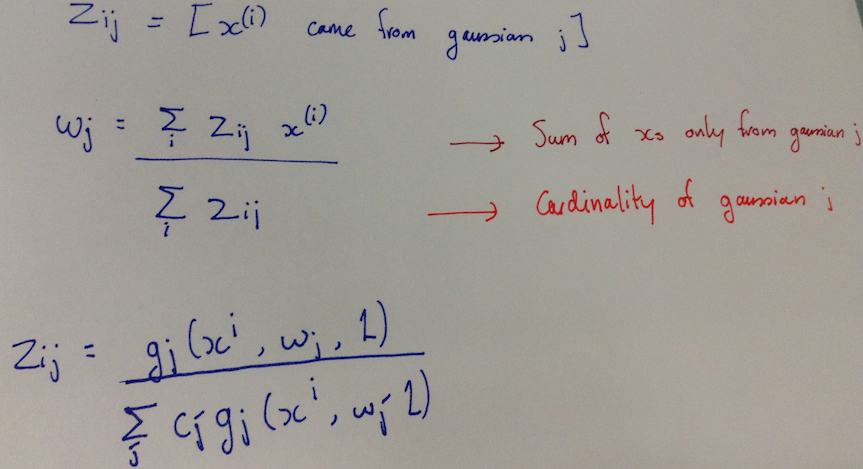
* Total of both distributions is 1.
* p(x) =c1p1(x) + c2p2(x) where c1+c2=1 and c1,c2>=0
  + c1,c2 are known as our mixing co-efficients.

We have:

* k gaussians
  + Gaussian j has a mean of wj.
  + Unit variant

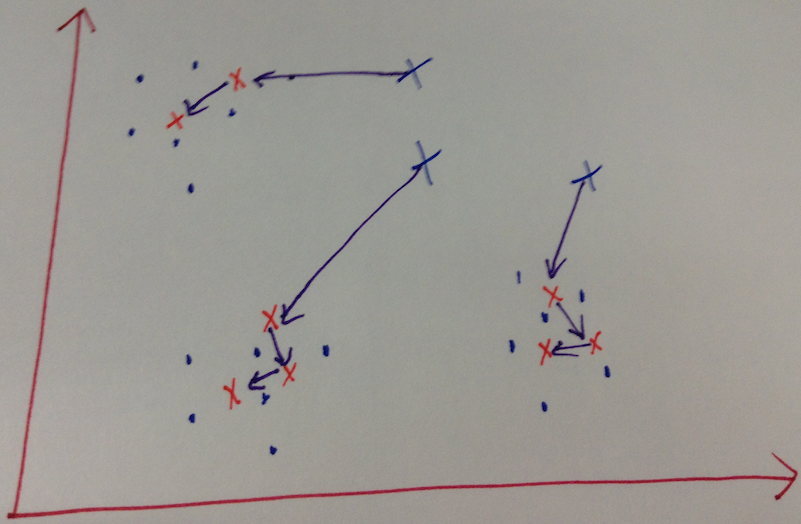






**E-M Algorithm:**

* Expectation, Maximisation



* We keep iterating until out points are fairly centre in their respective cluster.
* Above:
  + The blue dots represent our cluster, in this example we’ve 3.
  + The blue xs represent our original points
  + The purple arrows signify where we ended up with each point after each iteration until the final red x in each sequence.
* More clusters, the better the fit for the data but it will also increase the complexity.