Expectation Maximization and Gaussian Mixture Model

Lecture 12

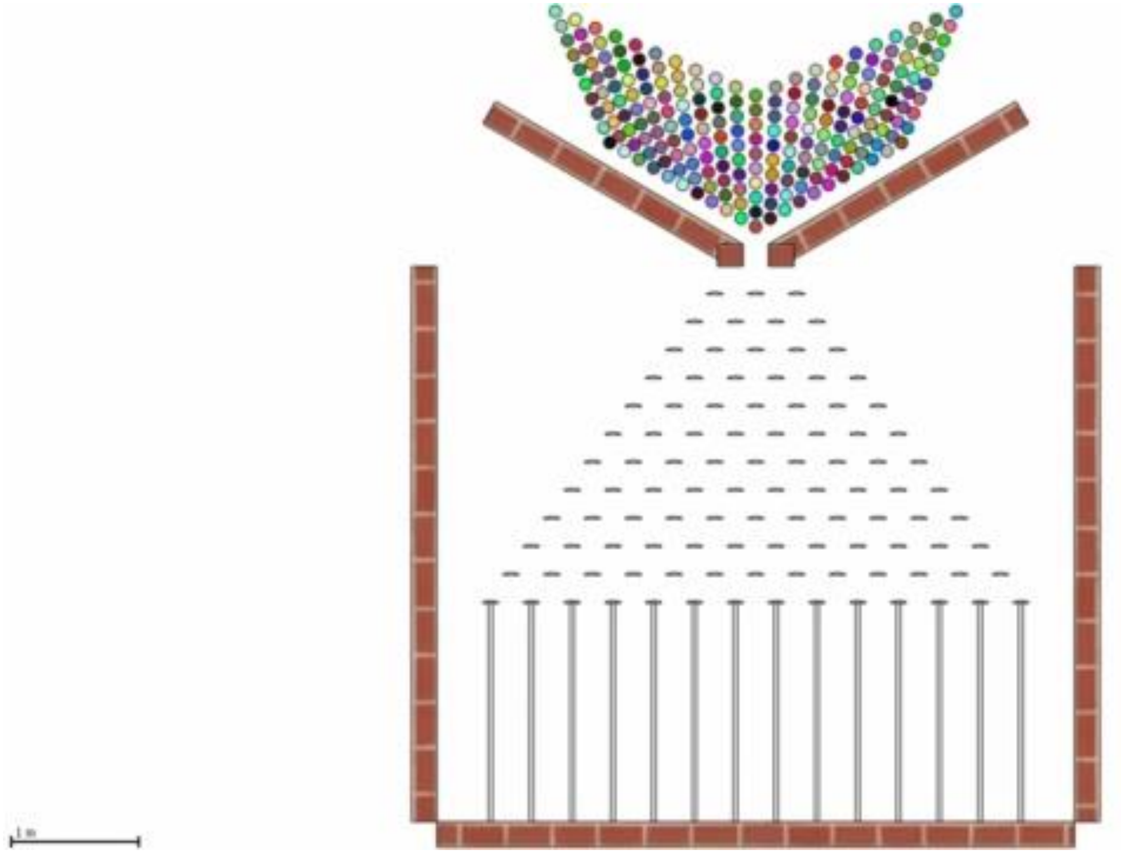
Jeong-Yean Yang

2020/12/10

Multi Dimensional Probabilistic Distribution

1

Gaussian Distribution

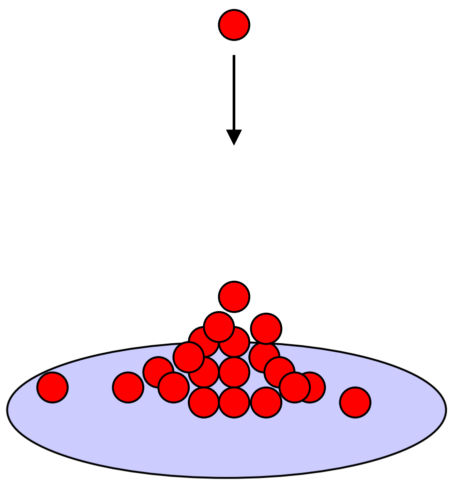


$$\Pr(x) = \int_{-\infty}^{x} p(x)dx, \quad PDF = p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma} \right)^2 \right)$$

With C++ or Python, How to Generate Gaussian Distribution?

- Rand() returns integer from 0 to RAND_MAX(32767)
  - Rand() is NOT Gaussian(Normal) distribution

- Remind the video

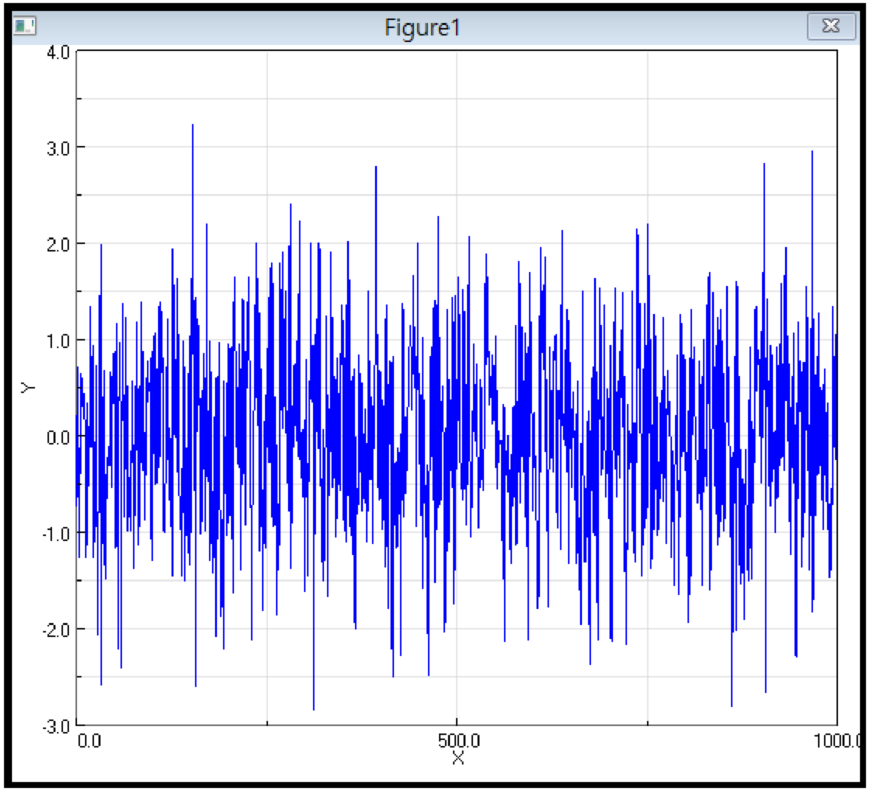


*Marsaglia polar method

$$r \sim N(0,1)$$

```
double u,v,r;
while(1)
{
    u=2*rand()/((double)RAND_MAX)-1;
    v=2*rand()/((double)RAND_MAX)-1;

    r=u*u+v*v;
    if (r==0 || r>1)    continue;
    break;
}

r   = sqrt(-2*log(r)/r);
r   = u*r;
```

N(0,1) returns Gaussian Distribution



1000 samples

randn(1,1000) generates 1000 samples

Question:

How we generate x with mean and standard deviation?

$$x \sim N(0,1)$$

$$x' \sim N(\mu, \sigma^2)?$$

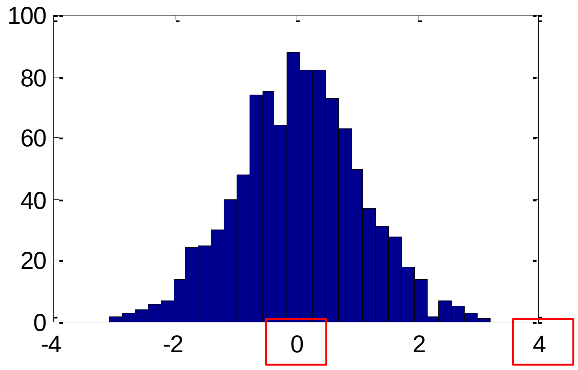
Gaussian Generation $x' \sim N(\mu, \sigma^2)$

- Mean value: μ is a offset from 0

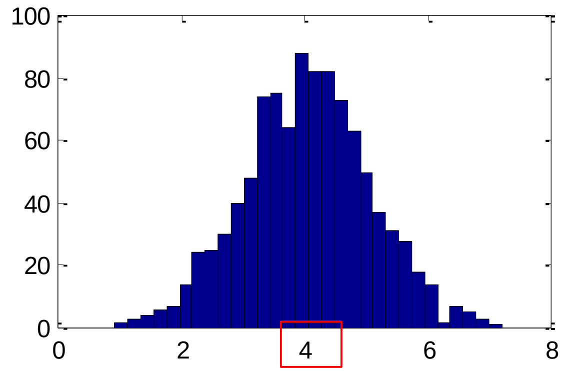
$$x \sim N(0,1) \quad \Rightarrow \quad x' \sim N(0,1) + \mu = N(\mu,1)$$

- Standard deviation

$$x \sim N(0,1) \quad \Rightarrow \quad x' \sim \sigma N(0,1) = N(0,\sigma^2)$$

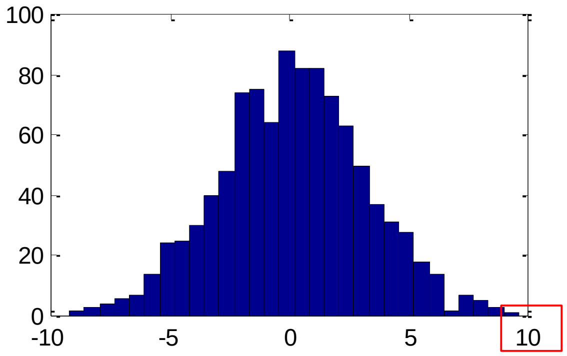


$x \sim N(0,1)$



$x' = x + 4$

$x' \sim N(0,1) + 4 = N(4,1)$



$x' = 3x$

$x' \sim 3N(0,1) = N(0,3^2)$

Gaussian Distribution or Normal Distribution(Z)

$$\mathrm{z} \sim N(0,1) \qquad \mathrm{z} = \frac{x-\mu}{\sigma} \sim N(0,1)$$

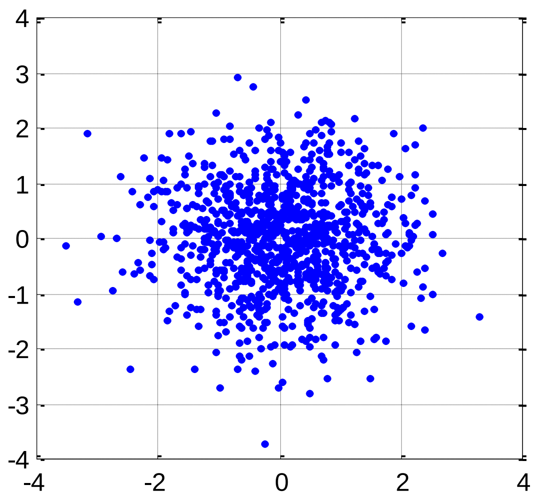
$$x \sim \sigma N(0,1) + \mu = N(\mu, \sigma^2)$$

- We learn it at high school, TT.
- Z is called “Normal Distribution” $\mathrm{PDF(z)} = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2} z^2\right)$
- X is normalized with mean and standard deviation

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

Probability in 2D Space

- How to generate 2D Gaussian Distribution?
  - Easy. A= randn(1000,2) and plot(A(:,1),A(:,2),'.')

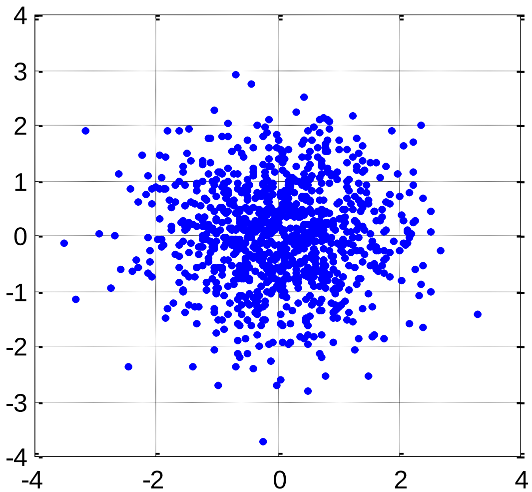


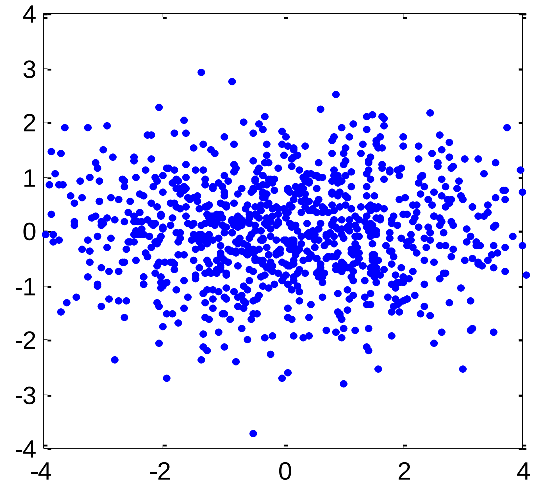
Plot(A(:,1),A(:,2),'.')

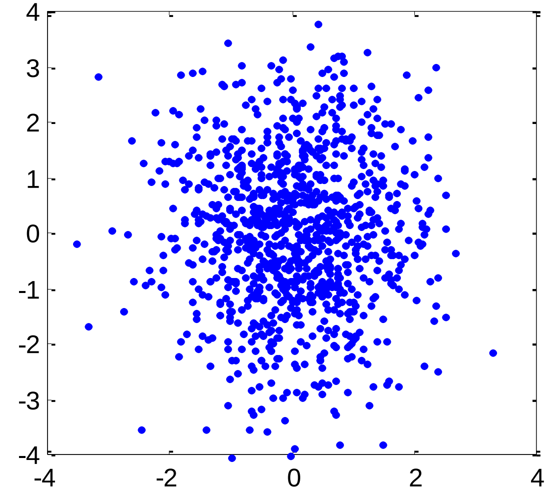
1 DIM $z_1 \sim N(0,1)$

2 DIM $z_2 = \begin{pmatrix} x \\ y \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \sigma^2\right)$

$$\mu = \begin{pmatrix} x_{mean} \\ y_{mean} \end{pmatrix} \qquad \sigma = ?$$







Plot(A(:,1),A(:,2),'.')

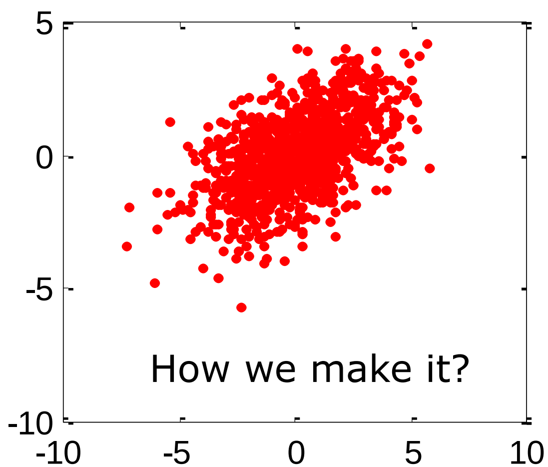
Plot(2*A(:,1),A(:,2),'.')

Plot(A(:,1), 1.5*A(:,2),'.')

$$z_2 = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$z'_2 = \begin{pmatrix} 2x \\ y \end{pmatrix}$$

$$z'_2 = \begin{pmatrix} x \\ 1.5y \end{pmatrix}$$



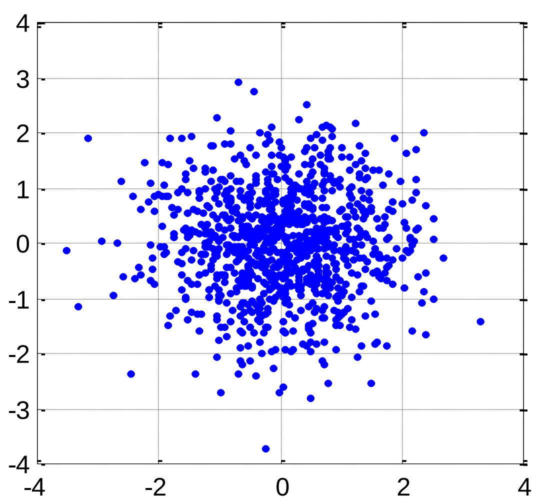
$$z' = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \Sigma\begin{pmatrix} x \\ y \end{pmatrix}$$

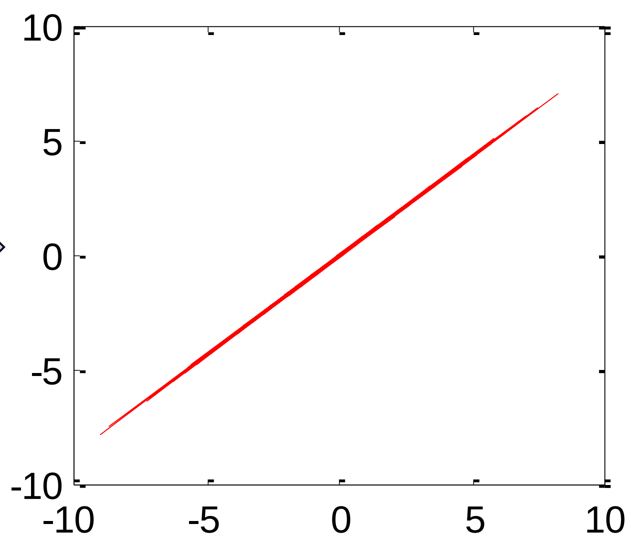
Quiz 1

$$z' = \begin{pmatrix} 2 & \sqrt{3} \\ \sqrt{3} & 1.5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

How it will distribute?

$$\text{Hint}: Det \begin{pmatrix} 2 & \sqrt{3} \\ \sqrt{3} & 1.5 \end{pmatrix} = 3 - 3 = 0$$





Quiz 2
Why PDF is Over One?

- What is PDF?

$$\Pr(x) = \int_{-\infty}^{x} p(x)dx,\ \text{PDF}=\ p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$$

- PDF is not a Probability. p(0) may be over 1.

$$p(x) = p(0)\Big|_{\substack{\sigma=0.1 \\ \mu=0}} = \frac{1}{0.1\sqrt{2\pi}} \exp\left(-\frac{1}{2}(0)^2\right) \simeq 3.99$$

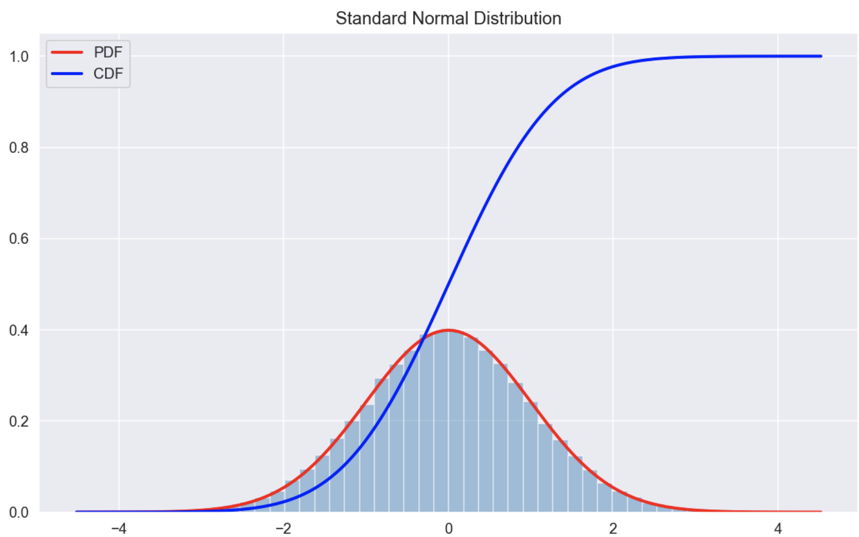
- Gaussian function is NOT a Probabilistic function
  But is a Probabilistic Density Function

Cumulative Distribution Function(CDF) is the integration of PDF

- Think Probability Exactly

$$\text{PDF}= \ g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right)$$

$$CDF = \int_{-\infty}^{x} g(x)dx = \Pr(x) = \text{Prob}(x)$$



$$\therefore \int_{-\infty}^{\infty} g(x)dx = 1$$

- d(CDF)/dx = PDF
- p(x) in PDF is NOT a probability

Probabilistic Density Function in n-dim. Space

- 1Dim

$$\Pr(x) = \int_{-\infty}^{x} g(x)dx, \ \text{PDF=} \ g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2 \right) \quad x \sim N(\mu, \sigma^2)$$

- N-Dim

$$g(\hat{x}) = \left((2\pi)^N Det(\Sigma)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu) \right) \quad \hat{x} \sim N(\hat{\mu}, \Sigma)$$

- Look, Sigma matrix

$$\Sigma = \begin{pmatrix} 2 & 0.5 \\ 0.5 & 1.5 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} 2 & 0 \\ 0 & 1.5 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \cdots & 0.5 \\ 0.5 & \cdots \end{pmatrix}$$

Scale factor for principal axis

Rotation

Important for Map matching

Two types of Probability

- A Priori Probability
  - When you use probability, you use a prior probability

$$\Pr(A) = 0.6$$

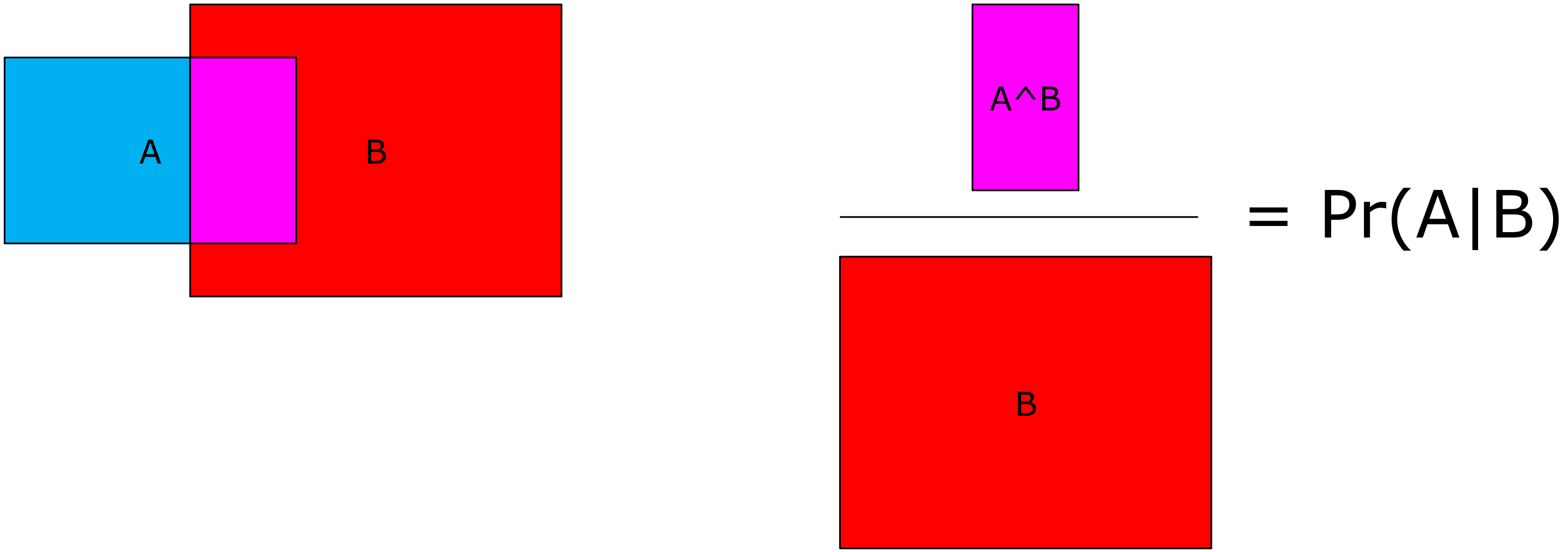
- Posterior Probability (Conditional probability)
  - Bayesian probability
  - Prob. Of A on condition that B occurs,

$$\Pr(A \mid B) = 0.6$$

- A prior and Posterior probability are very different.

Conditional Probability

- What is Pr(A|B)?
  - Probability of A under the Probability of B
  - Or Probability of A within the given B



Posterior Prob.

- When events A and B occur,
- P(A): Probability of A occurrence
- P(B): Probability of B occurrence.
- P(A^B): Probability of Both A and B occurrence
- Definition:

$$\therefore \mathrm{P}(A|B) = \frac{P(\mathrm{A} \wedge \mathrm{B})}{P(B)}$$

$$P(A|B)P(B) = P(A \wedge B) = P(B|A)P(A)$$

$$\therefore \mathrm{P}(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

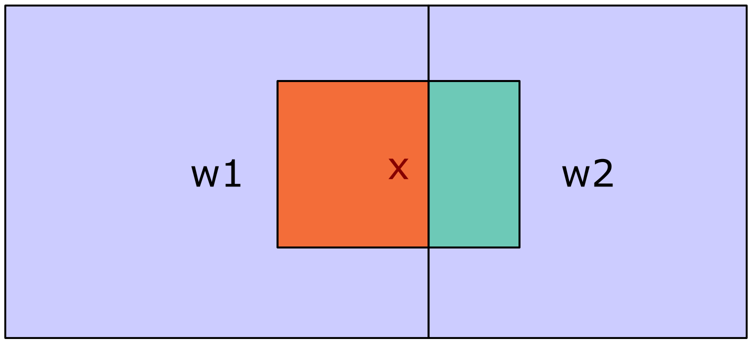
Engineering Notation

$$\mathrm{P}(\mathrm{w} \mid \mathrm{x}) = \frac{P(\mathrm{x} \mid \mathrm{w}) P(\mathrm{w})}{P(\mathrm{x})}$$

$$Posterior = \frac{likelihood \times prior}{Evidence}$$

In engineering, likelihood is one of the popular solution.

Prob. Of Event X between w1 and w2



Prior Prob. :$p(w_1), p(w_2)$

$$p(x) = ?$$

$$p(x) = \quad + $$

$$= p(x, w_1) + p(x, w_2)$$

$$= p(x \mid w_1)\, p(w_1) + p(x \mid w_2)\, p(w_2)$$

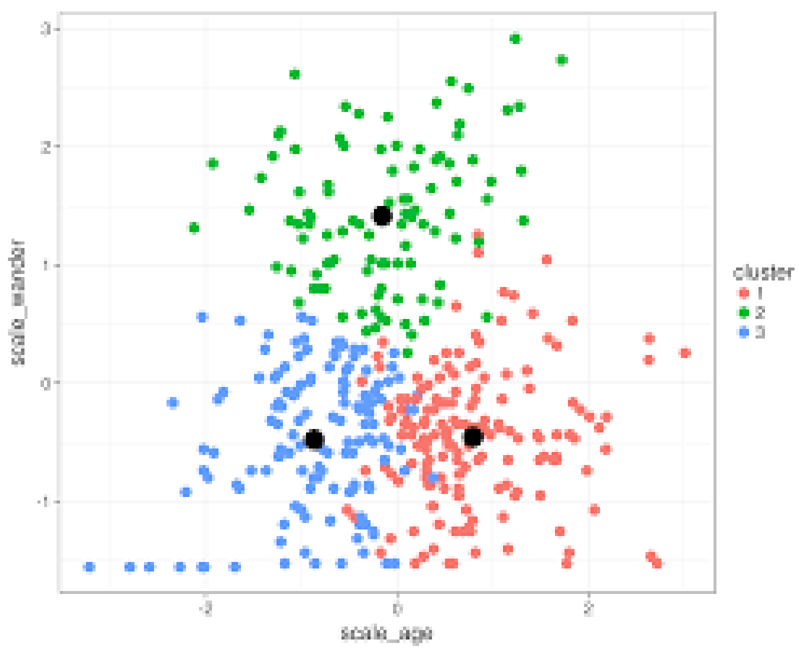
- p(x)= Probability of event x's occurrence
- Posterior probability must be required for Classification

$$p(w_1 \mid x) = \frac{p(x \mid w_1)\, p(w_1)}{p(x)} = \frac{p(x \mid w_1)\, p(w_1)}{\sum_i p(x_i \mid \mathrm{w}_i)\, p(w_i)}$$

2 Concept of Clustering

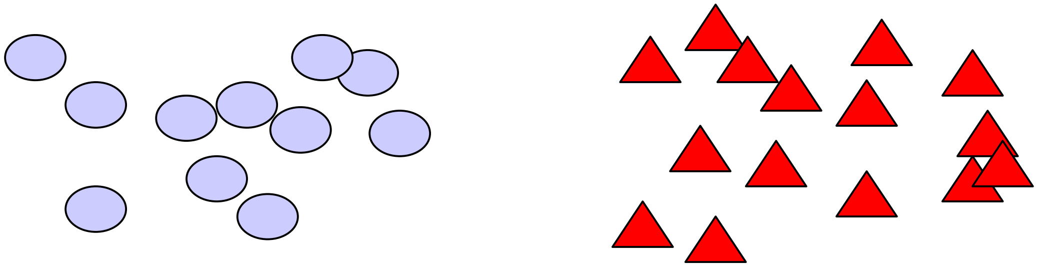
What is a Clustering?

- Grouping similar objects and labeling a Group
  - Labeling a Class
- Grouping a set of Objects which are more similar to each other than to those in other groups



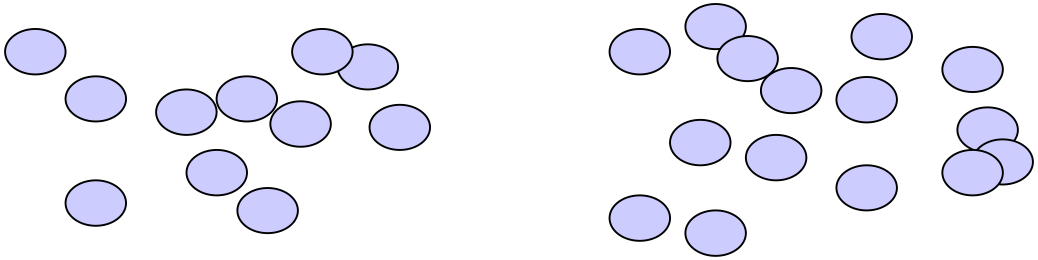
Clustering Method
Important Tools for Intelligent Robotics

- Pattern recognition requires Class definition



2 classes

- How many classes here?



- There are only two lumps → Two clusters.

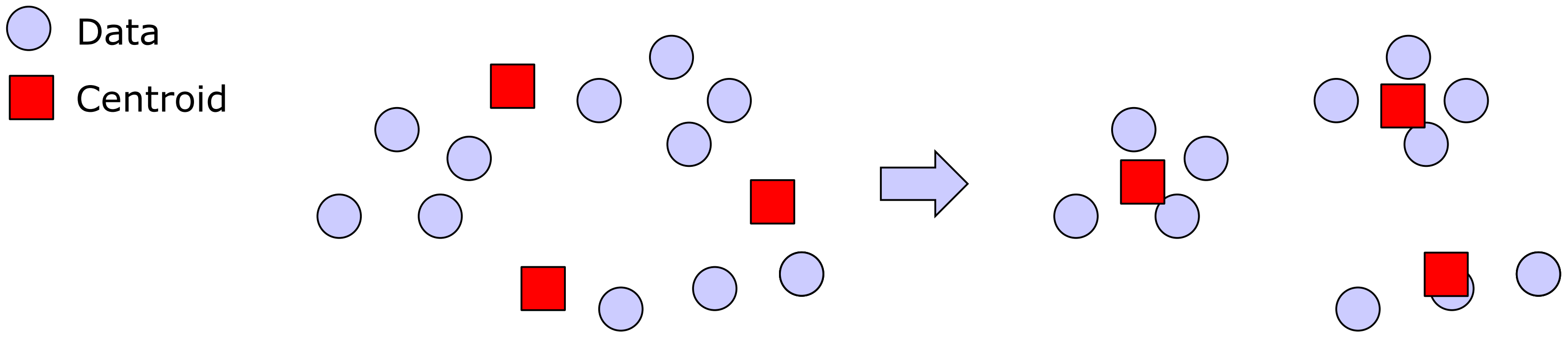
Famous Clustering method

- 1. K-Means Clustering method
  - Geometry based method
  - Simple and low computational burdens.
  - Shortcoming: Initial guess determines the final result

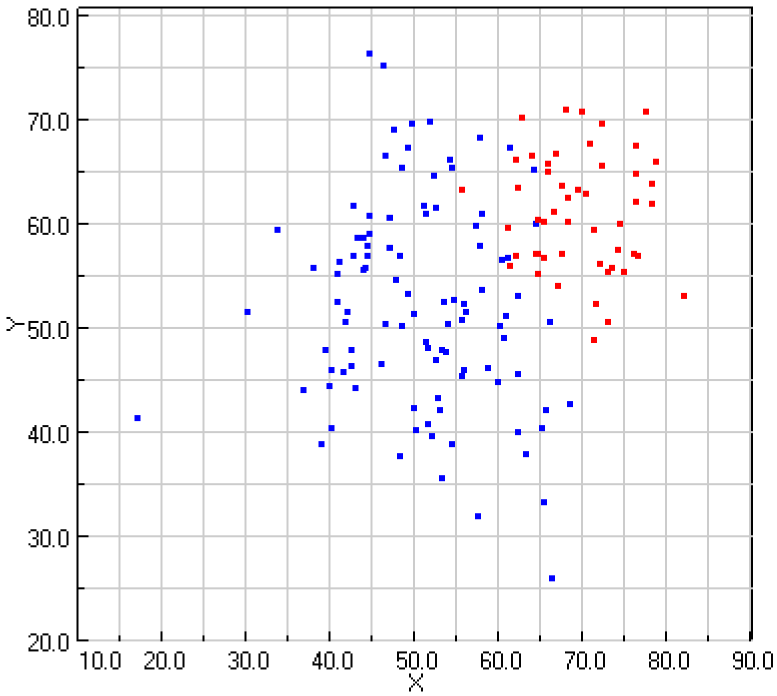
- 2. Expectation Maximization method
  - Probabilistic method
  - Very popular for **fitting Mixture Distribution**
  - Back bone of Gaussian Mixture Model (GMM)

K-Means Clustering

- Find Mean value (Centroid) for each cluster
- Algorithm
- 1. Assume there are K clusters.
- 2. Guess each centroid of cluster.
- 3. Find k points to closest centroid
- 4. Recompute the centroid of each cluster.



ex/ml/l12kmean.py



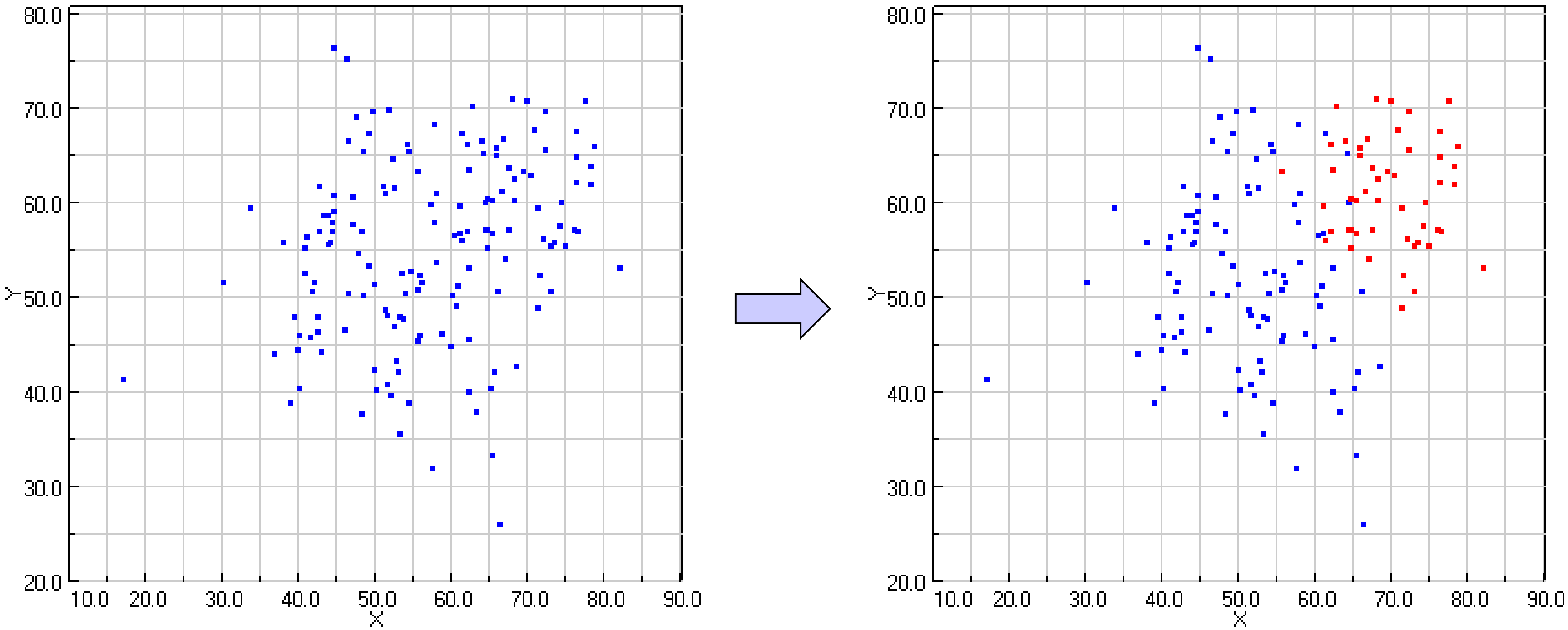

```
def gendata():
    a=randn(100,2)*10
    a[:,1] = a[:,1] + 50
    a[:,2] = a[:,2] + 50

    b=randn(50,2)*5
    b[:,1] = b[:,1] +70
    b[:,2] = b[:,2] +60
```

$$\text{Blue} \sim N(\mu,\sigma^2) = N([50,50],\begin{bmatrix} 10^2 & 0 \\ 0 & 10^2 \end{bmatrix})$$

$$\text{Red} \sim N(\mu,\sigma^2) = N([70,60],\begin{bmatrix} 5^2 & 0 \\ 0 & 5^2 \end{bmatrix})$$

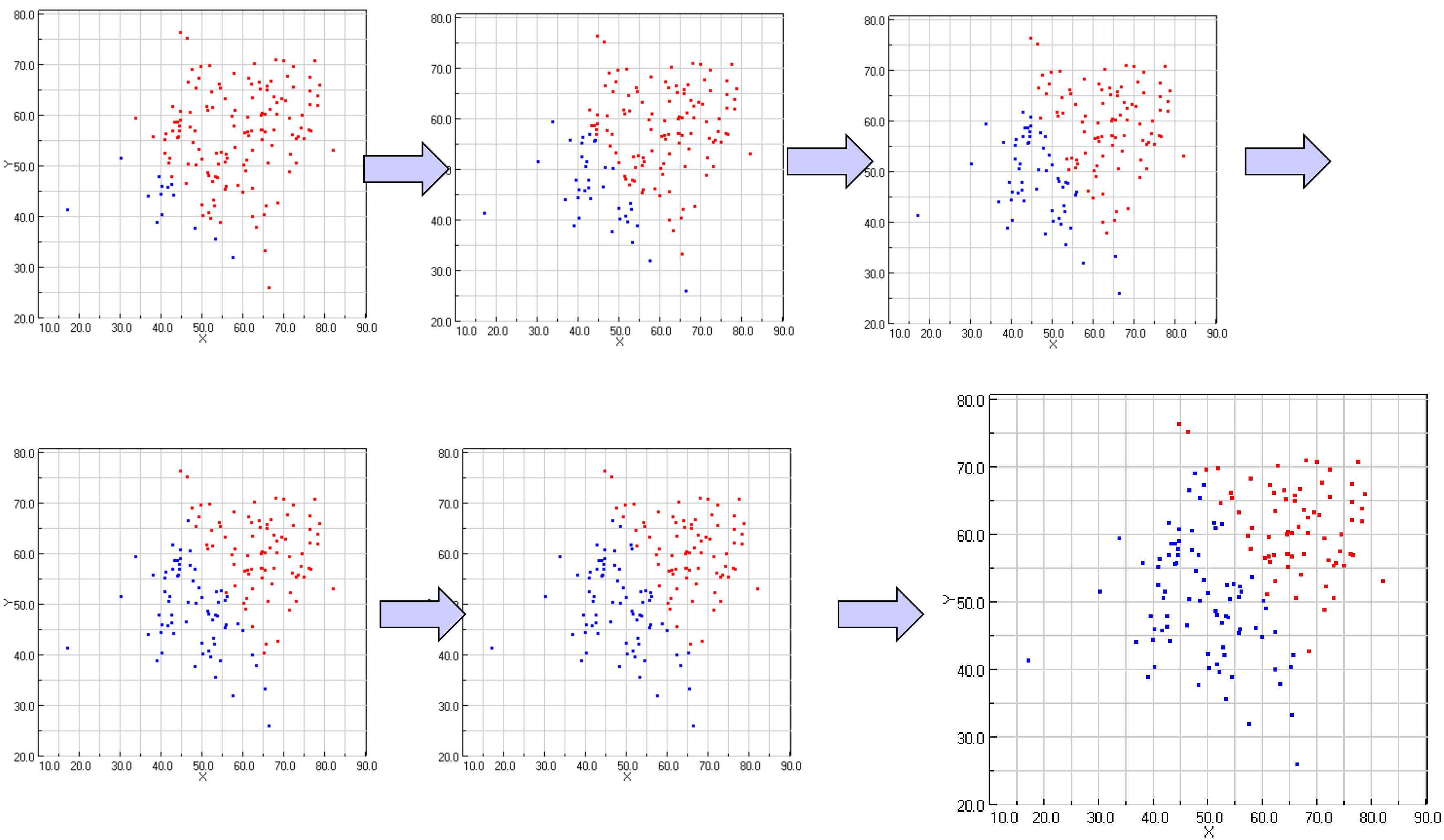
- Two groups with Blue and Red
- It looks easy to find two groups

Real Problem is to find Two Groups

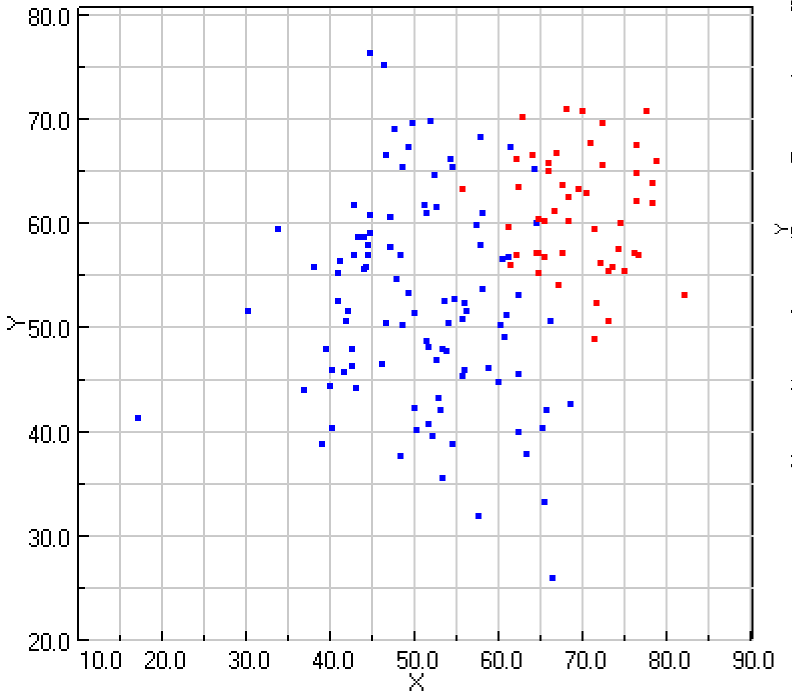


- It is NOT easy.
- By iteration, we find two groups from initial guesses.

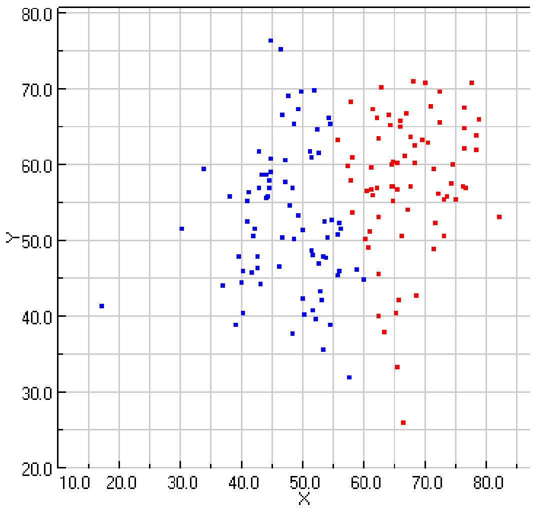
$112\,\mathrm{kmean.test}(\mu_{1x}, \mu_{1y}, \mu_{2x}, \mu_{2y}, iteration)$



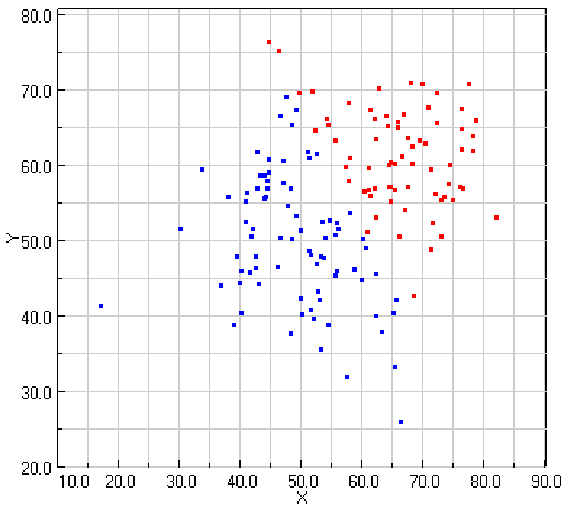
l2kmean.test with Different Guesses



True value



(40,50) and (80,50)



(20,30) and (80,80)

- The Results are strongly affected by Initial Guesses

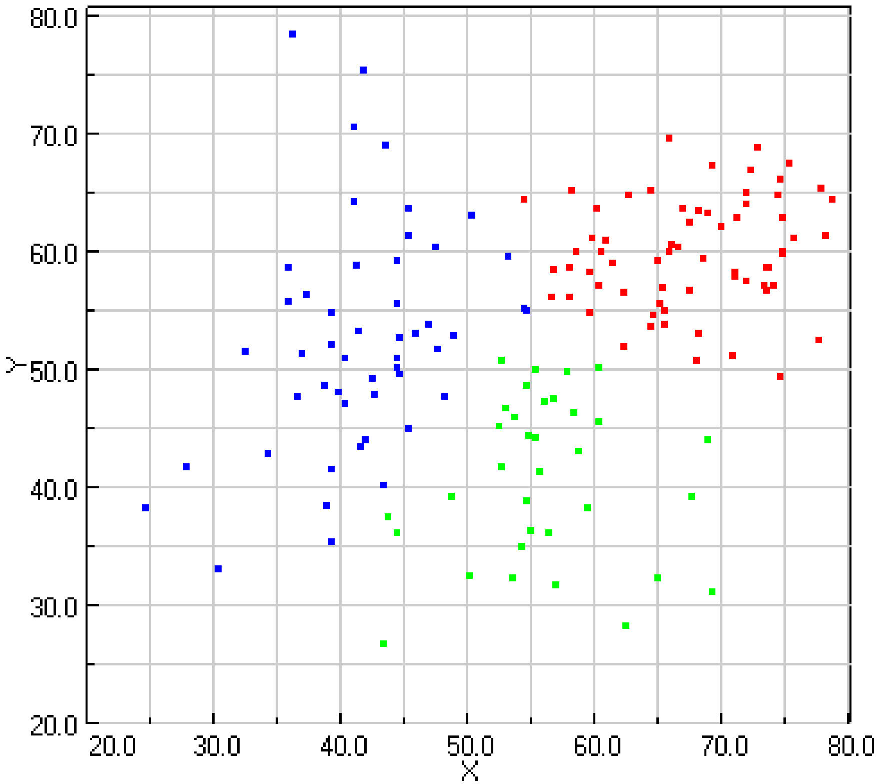
Centroid of Cluster
What is it?

- In k means cluster,
  - Centroid approaches mean value of the test distribution.
  - But, it is not on the Exact mean value.
  - Why?

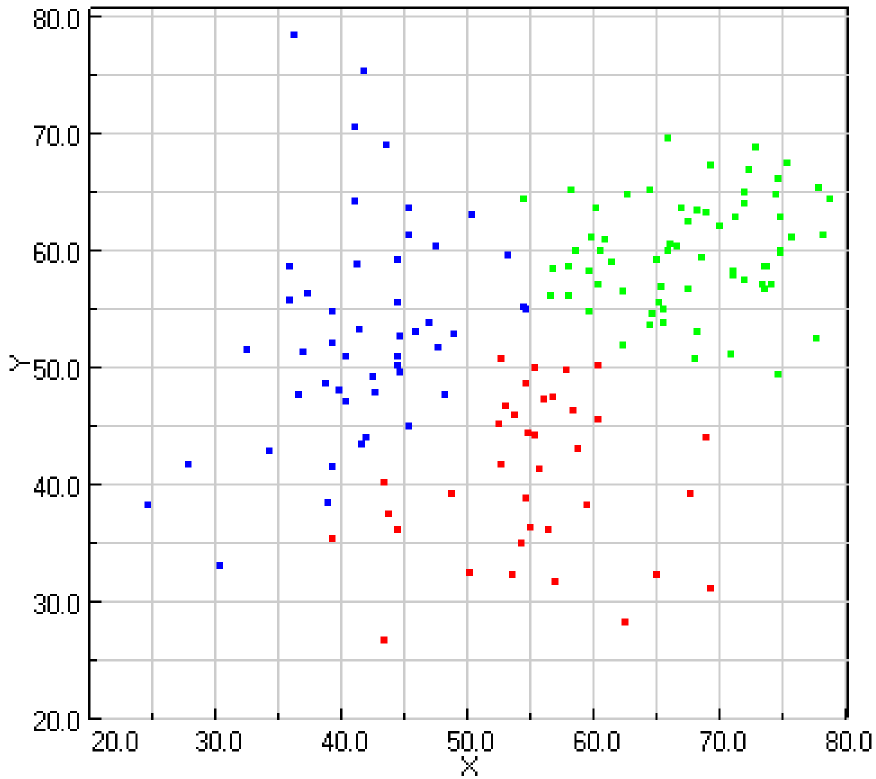
- Think the role of K mean cluster.
  - K closest points are Not whole data. Just Sample.

    → In each turn, K mean clustering method find the centroid of K closest points.
  - If Initial centroid is biased, centroid is sometimes biased.
- If we guess wrong number of centroid, how it works?

Wrong Number of Groups



kmean.test3(50,50,70,70,60,30,20)



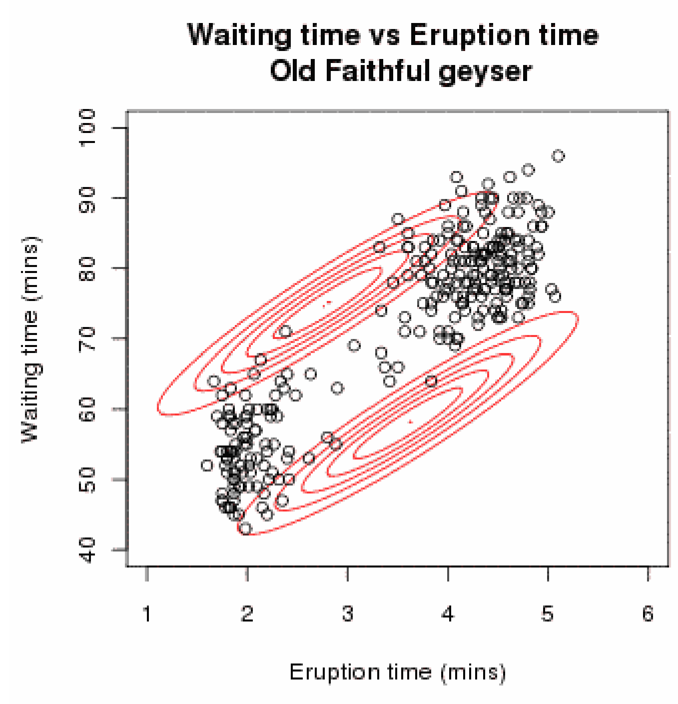
kmean.test3(40,80,70,30,50,50,20)

- Thus, what is the Answer? → No answer in General.

3 Expectation Maximization

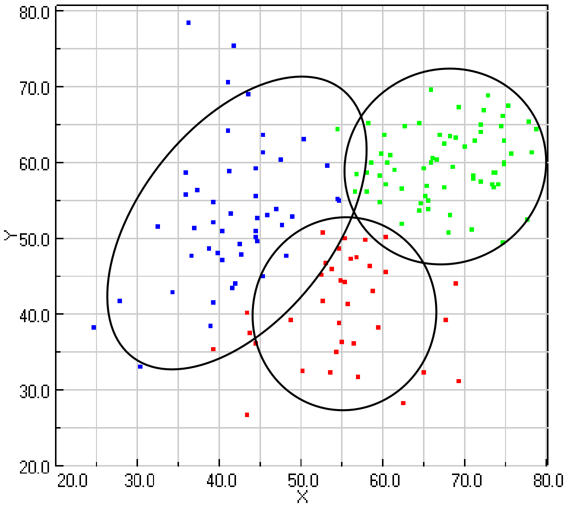
Introduction to Expectation Maximization

- Let's think EM in a simple way.
- We have random variable, X
- Maybe, X has two groups.
- How we separate X with two groups, probabilistically?



EM has two Steps

- Clusters are represented by Probability Distribution
  - K-means Clustering is a set of data around centroids.
  - But, clusters in EM are the Probabilistic Distribution
- Assumption:
  - Data are the Mixture of Gaussian Distributions
  - Blue, Red, and Green points are mixed with Gaussian distribution

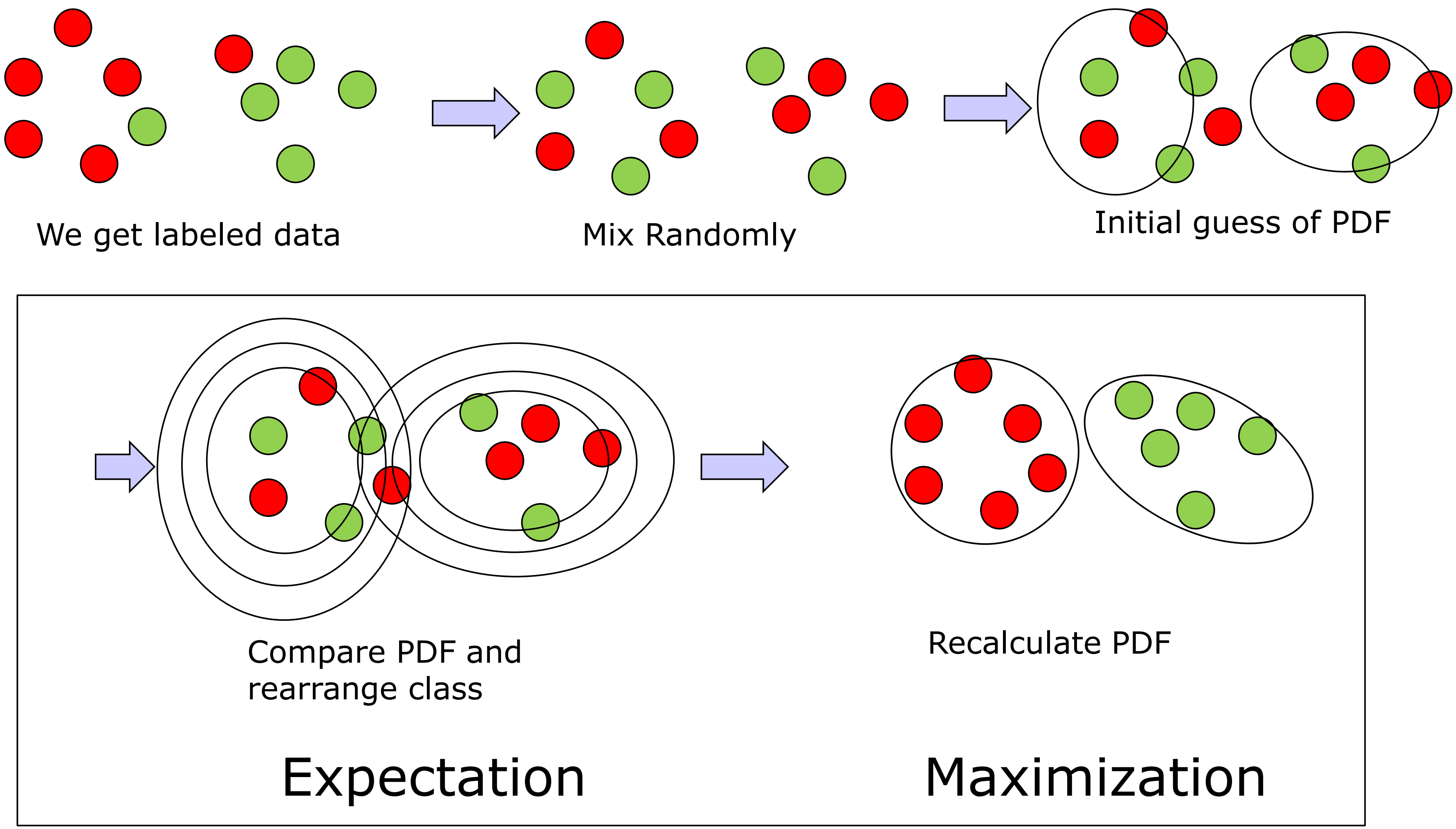


$$g(\hat{x}) = \left((2\pi)^d Det(\Sigma)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

$$p(\hat{x}_{Blue} \mid C_{Blue}),\ p(\hat{x}_{Red} \mid C_{Red}),\ p(\hat{x}_{Green} \mid C_{Green})$$

$$\hat{x} = \{\hat{x}_{Blue}, \hat{x}_{Red}, \hat{x}_{Green}\}$$

Simple EM Procedure



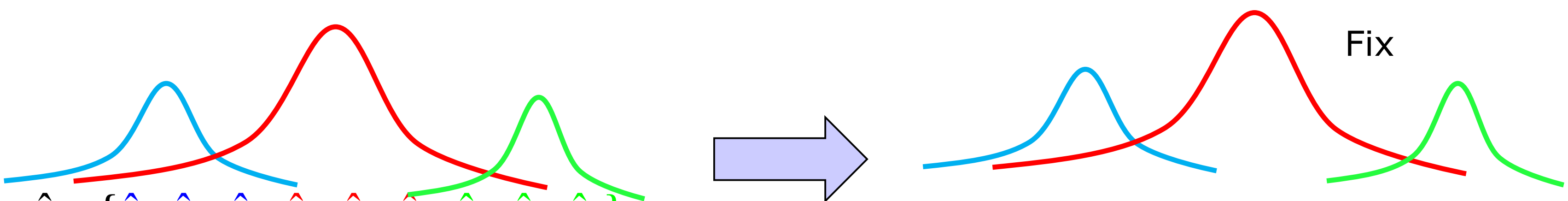
Repeat E-M

Probabilistic Density Function has mean and variance

- 0. Data is given $\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, \hat{x}_9\}$

Expectation

- 1. Guess groups $\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, \hat{x}_9\}$
- 2. maximum PDF is wrong in some data



$\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, \hat{x}_9\}$

Red?

$\hat{x} = \{\hat{x}_1, \hat{x}_2, \underline{\hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7}, \hat{x}_8, \hat{x}_9\}$

Maximization

- 3. Find mean and variance for each group

$$\hat{\mu}_1 = mean(\hat{x}_1, \hat{x}_2) \qquad s_1 = std(\hat{x}_1, \hat{x}_2)$$

$$\hat{\mu}_2 = \mathrm{mean}(\hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7) \qquad s_2 = std(\hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7)$$

$$\hat{\mu}_3 = \mathrm{mean}(\hat{x}_8, \hat{x}_9) \qquad s_3 = std(\hat{x}_8, \hat{x}_9)$$

Expectation and Maximization
Step 1. Expectation

- Density function, p(x|c) for each cluster, C

$$p(\hat{x} \mid C) = \left((2\pi)^d Det(\Sigma)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

- Density function, P(x) for clustering model, $M = \{C_0, C_1, \ldots, C_k\}$
  - W is the fraction of the Cluster C in the entire data

$$\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8, \hat{x}_9\} \longrightarrow W_{Blue} = W_{\mathrm{Red}} = W_{Green} = \frac{1}{3}$$

$$P(x) = \sum_i^k W_i \, p(x \mid C_i)$$

- Assign points to Clusters

$$P(\mathrm{C}_i \mid x) = W_i \frac{p(x \mid C_i)}{P(x)} = \frac{W_i \, p(x \mid C_i)}{\sum_i^k W_i \, p(x \mid C_i)}$$

Expectation and Maximization Step 2: Maximization

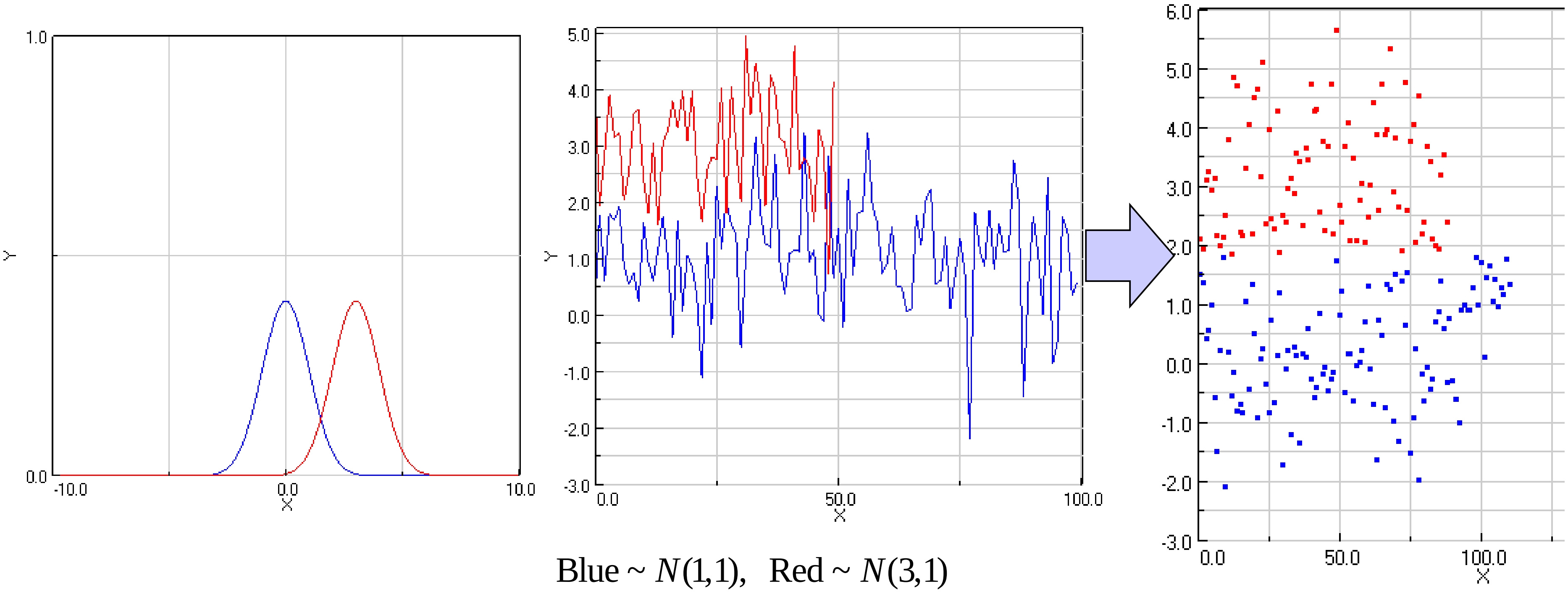
- Recompute Model $M' = \{C_0, C_1, ..., C_k\}$

$$W_i = \frac{1}{n} \sum_x P(C_i \mid x)$$

$$\mu_i = \frac{\sum_x x P(C_i \mid x)}{\sum_x P(C_i \mid x)}$$

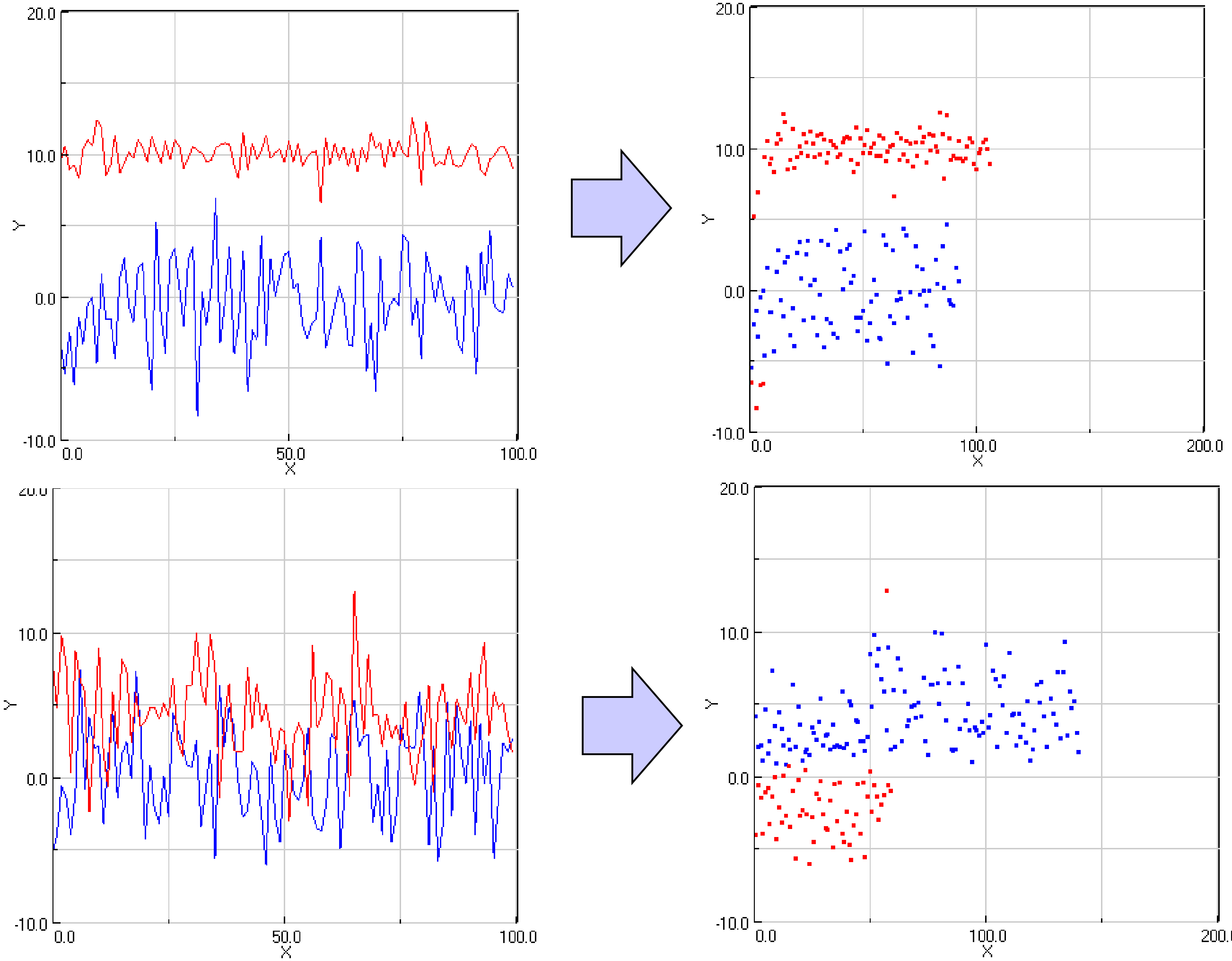
$$\Sigma_i = \frac{\sum_x (x - \mu_i)^2 P(C_i \mid x)}{\sum_x P(C_i \mid x)}$$

EM in 1 Dim.



Blue ~ $N(1,1)$, Red ~ $N(3,1)$

- Assume that there are 2 groups
- Guess x with Blue and Red groups



- Use same initial guess
- It is very Robust

$$\hat{\mu}_1 = 3, \hat{\mu}_2 = 5$$

$$\sigma_1 = \sigma_2 = 1$$

$$W_1 = W_2 = 0.5$$

But, EM is designed Carefully

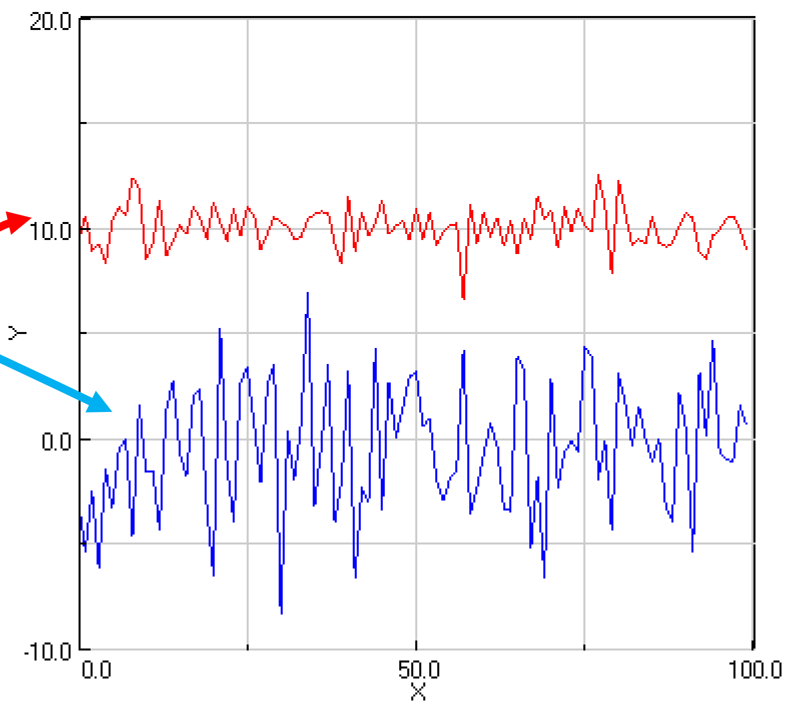
- EM looks simple.
- E-M or M-E shows very different result

- 1. Expectation with given parameters
  - Initial Guess of mean, variance, and fraction factor, W are first used.
  - At the first step, Do not calculate mean, variance, and so on
- 2. Maximization with p(c|x), and not with p(x|c)
  - E and M looks similar. It causes confusion
- 3. If M(calculate parameters) works first, EM often fails.

Example) ex/ml/l12em1.py Generate Blue and Red

```
#Generate data
def gendata():
    global a,b,c
    figure(1)
    graph(1)
    drawGF(0,1,'b')
    graph(2)
    drawGF(5,1,'r')

    a=randn(100,1)*3
    b=randn(100,1)*1+10
    figure(2);
    clear()
    graph(1)
    plot(a)
    graph(2)
    plot(b,'r')

    c=array(200,1)
    c[1:100] = a
    c[101:200]=b
```

Blue ~ $N(0,3^2)$, Red ~ $N(10,1)$



Example) ex/ml/l12em1.py
Initial Guess

```
def em():
    # step 0: guess model
    n = size(c,1)
    x = array(n,2)
    x[:,1]  = c
    x[:,2]  = randint(2,n,1)
    W = array(1,2)
    mb = 3
    sb =1
    mr = 5
    sr =1
    W[1,1] =0.5
    W[1,2] =0.5
```

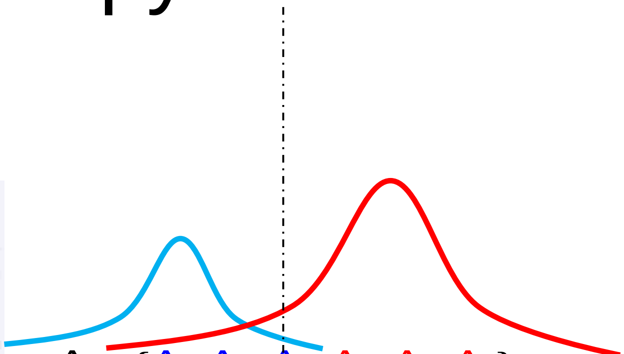
point	label
0.2	0
1.3	0
10.1	1
3.3	0
11.5	1

- Matrix X has two column
  - 1st column is random data
  - 2nd column, label 0 is blue and label 1 is red

- Mb=mean of blue
- Sb= standard deviation of blue
- Mr = mean of red
- Sr =standard deviation of red
- W[1,1] = W1
- W[1,2] = W2

Example) ex/ml/l12em1.py
Expectation

```
for i in range(0,50):
    # step 1: p(x|C)
    pb= normpdf(x[:,1],mb,sb)* W[1,1]
    pr= normpdf(x[:,1],mr,sr)* W[1,2]

    # step 2: who is better? P(C|x) with p(x|c) and P(x)
    for j in range(1,n+1):
        if (pb[j]>pr[j]):
            x[j,2] = 0
        else:
            x[j,2] = 1
```



$$\hat{x} = \{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6\}$$

1. This PDF is given by the previous(or initial) Parameters.

2. Blue p(x3) < Red p(x3)

Change x3's label is 1(red)

- P(x|C) is the p.d.f. of x with respect to a Cluster
- P(C|x) means a new Cluster, C is determined by p(x) comparison

Example) ex/ml/l12em1.py Maximization

```
# step 3: recompute
b = find(x[:,2]==0)
r = find(x[:,2]==1)
W[1,1] = size(b,1)/200
W[1,2] = size(r,1)/200

xb = x[b,1]
xr = x[r,1]

mb= mean(xb)
sb= std(xb)
mr= mean(xr)
sr= std(xr)
```

- With a new Model, M'

$$M' = \{C'_0, C'_1, \ldots, C'_k\}$$

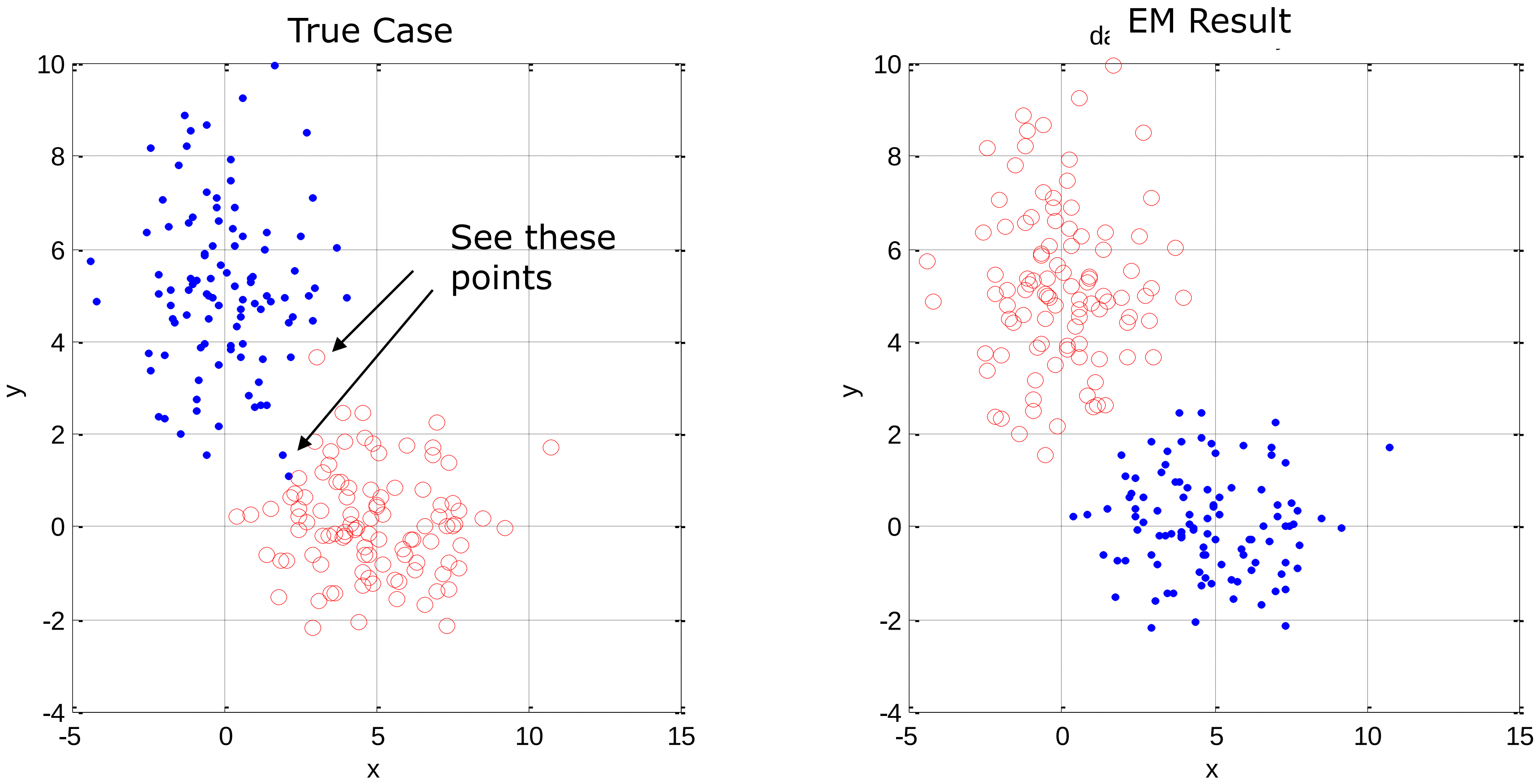
- Recompute Wi

$$W_i = \frac{1}{n} \sum_x P(C_i \mid x)$$

- New Mean and variance

$$\mu_i = \frac{\sum_x x P(C_i \mid x)}{\sum_x P(C_i \mid x)} \qquad \Sigma_i = \frac{\sum_x (x - \mu_i)^2 P(C_i \mid x)}{\sum_x P(C_i \mid x)}$$

EM in 2Dim

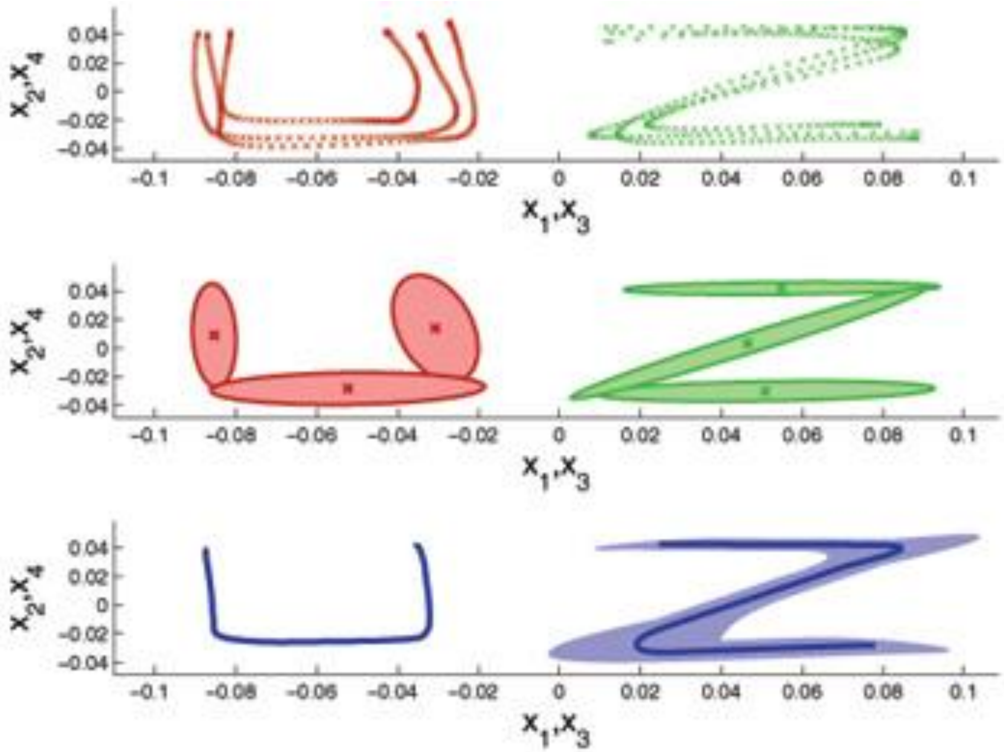


- Above two points are regarded as Blue one in the right picture.
  - Because, EM is based on a probabilistic distribution.

Why We Learn EM and GMM?

Imitation Learning is Not Doing Memorized Motion

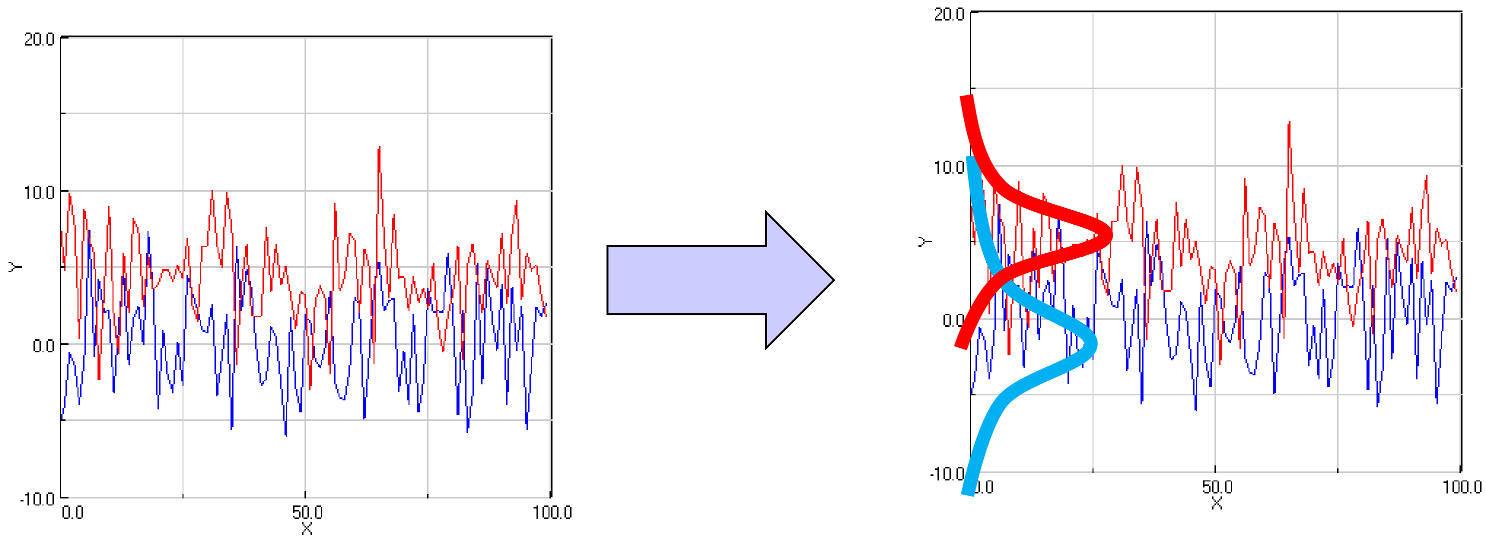
- 1990's: Encoder Recording and Replay
- After 2005: Trajectories are considered as the set of Stochastic Process



4 Gaussian Mixture Model

Gaussian Mixture Model

- Extend k-means Clustering into a Probabilistic framework as like EM method



- Left signal is the mixture of Two Different Gaussian
  - Goal of GMM is to find Multiple Gaussian Distributions

Modeling of GMM

- Assume that the *j* th point of the vector x belongs to the *i* th Cluster.

$$p(x) = \sum_i \pi_i p(x \mid \mu_i, \Sigma_i) \qquad \sum_i \pi_i = 1$$

- Gaussian PDF of the *i* th cluster is defined as,

$$G_i(x) = f(x, \mu_i, \Sigma_i) = \left((2\pi)^N Det(\Sigma)\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right)$$

x : the input vector

μ_i : the mean value of the ith cluster

Σ_i : the covariance(variance) of the ith cluster

Example
i for Cluster and j for input, x

$i=1$

$$x = \begin{bmatrix} 1 \\ 1.1 \\ 10 \\ 10.1 \end{bmatrix}$$

$i=2$

$$\pi_{i=1} = \frac{1}{2}, \pi_{i=2} = \frac{1}{2}$$

$$x^{j=1} = 1$$
$$x^{j=2} = 1.1$$
$$x^{j=3} = 10$$
$$x^{j=4} = 10.1$$

- π is the prior probability.

$$\pi_i = \Pr(x \in C_i)$$

Probability of the *j*th point belongs to the *i*th cluster

$$W_i^j = \frac{\pi_i G_i(x)}{\sum_{k=1} \pi_k G_k(x)}$$

$$x = \begin{bmatrix} 1 \\ 1.1 \\ 10 \\ 10.1 \end{bmatrix}$$

1, 1.1

10, 10.1

$$W_1^j = \frac{\pi_1 G_1(x_j)}{\pi_1 G_1(x_j) + \pi_2 G_2(x_j)}$$

$$\mu_1 = \frac{1+1.1}{2}$$

$$G_1(x) = f(x, \mu_1, \Sigma_1)$$

$$\mu_2 = \frac{10+10.1}{2}$$

$$G_2(x) = f(x, \mu_2, \Sigma_2)$$

Expectation Procedure: Probability of the *j*th point belongs to the *i*th cluster

$$W_i^j = \frac{\pi_i G_i(x)}{\sum_{k=1} \pi_k G_k(x)}$$

$$W_i^j = \frac{\pi_1 G_1(x_j)}{\pi_1 G_1(x_j) + \pi_2 G_2(x_j)}$$

$$x = \begin{bmatrix} 1 \\ 1.1 \\ 10 \\ 10.1 \end{bmatrix}$$

$$W_1^{j=1} = \frac{\pi_1 G_1(x_1 = 1)}{\pi_1 G_1(x_1 = 1) + \pi_2 G_2(x_1 = 1)}$$

$$W_1^{j=2} = \frac{\pi_1 G_1(x_2 = 1.1)}{\pi_1 G_1(x_2 = 1.1) + \pi_2 G_2(x_2 = 1.1)}$$

$$W_1^{j=3} = \frac{\pi_1 G_1(10)}{\pi_1 G_1(10) + \pi_2 G_2(10)}$$

$$W_1^{j=4} = \frac{\pi_1 G_1(10.1)}{\pi_1 G_1(10.1) + \pi_2 G_2(10.1)}$$

$$W_2^{j=1} = \frac{\pi_2 G_2(x_1 = 1)}{\pi_1 G_1(x_1 = 1) + \pi_2 G_2(x_1 = 1)}$$

$$W_2^{j=2} = \frac{\pi_2 G_2(x_2 = 1.1)}{\pi_1 G_1(x_2 = 1.1) + \pi_2 G_2(x_2 = 1.1)}$$

$$W_2^{j=3} = \frac{\pi_2 G_2(10)}{\pi_1 G_1(10) + \pi_2 G_2(10)}$$

$$W_2^{j=4} = \frac{\pi_2 G_2(10.1)}{\pi_1 G_1(10.1) + \pi_2 G_2(10.1)}$$

Maximization

- What is the objective function?

$$p(x) = \sum_i \pi_i p(x_i; \mu_i, \Sigma_i) \longrightarrow \quad Best\ \mu_i, \Sigma_i\ for\ Cluster\ \mathrm{i}$$

$$p(x_i) \to p\left(x_i^j, w_i^j\right) = p\left(x_i^j \mid w_i^j\right) p\left(w_i^j\right)$$

- Log likelihood

$$L(\pi, \mu, \Sigma) = \log \prod_j p(x^j; \pi, \mu, \Sigma)$$

$$= \sum_j \log p(x^j; \pi, \mu, \Sigma) = \sum_j \log p(x^j \mid \mathrm{w}^j; \mu, \Sigma) p(\mathrm{w}^j; \pi)$$

Maximization of Log likelihood

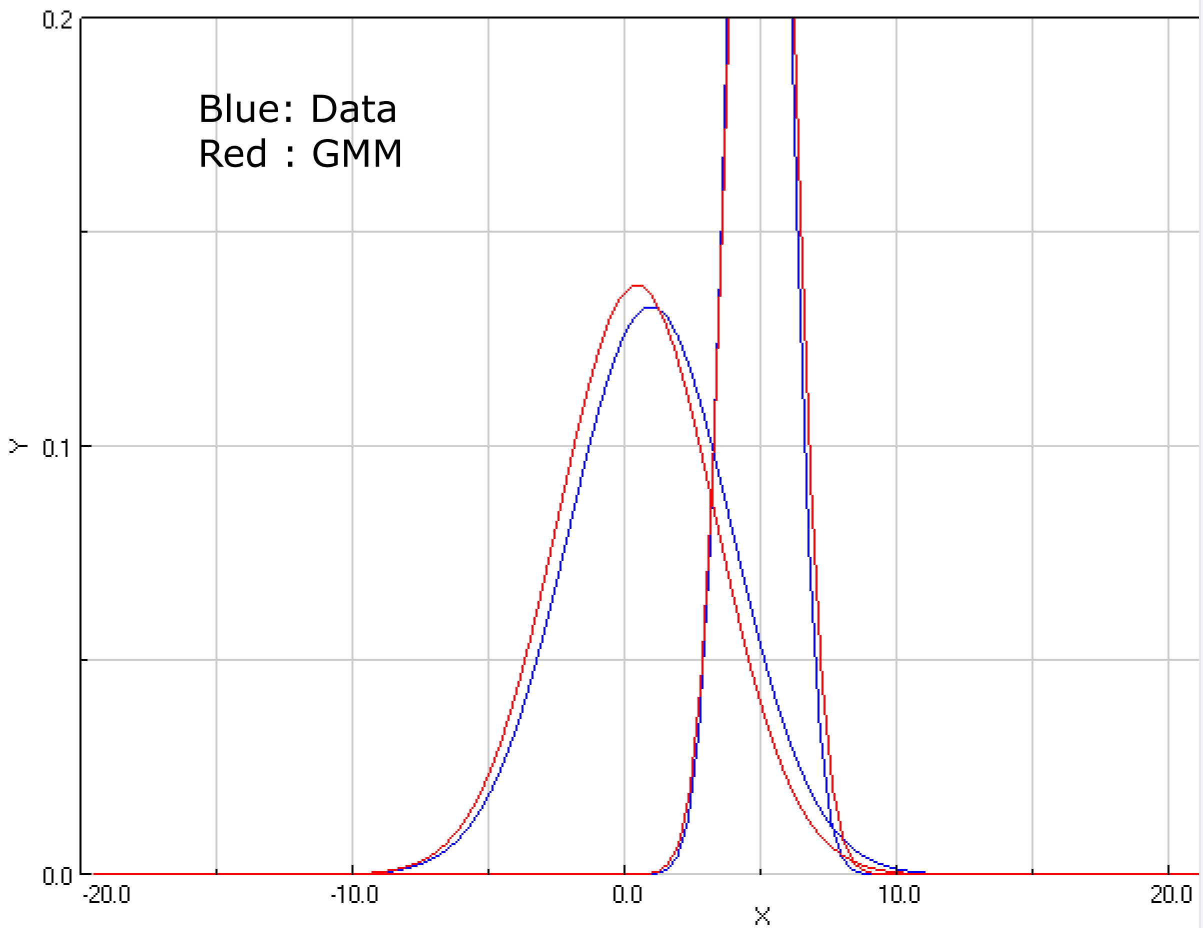
$$x = \{x^1, x^2, \ldots, x^N\}^T$$

$$\pi_i = \frac{1}{N}\sum_{j=1}^{N} W_i^j$$

$$\mu_i = \frac{\sum_{j=1}^{N} W_i^j x^j}{\sum_{j=1}^{N} W_i^j}, \quad \Sigma_i = \frac{\sum_{j=1}^{N} W_i^j (x^j - \mu_i)(x^j - \mu_i)^T}{\sum_{j=1}^{N} W_i^j}$$

Example of gmm1

- Edit ex/ml/gmm1




```
def gmm(x,k):
    N =size(x,1)

    # Guess initial model
    m=array(1,k)
    s=array(1,k)
    p=array(1,k)
    W=zeros(N,k)

    for i in range(1,k+1):
        m[i]= x[randint(N)+1]
        s[i]= sqrt(cov(x))
        p[i]= 1./k

    # Do EM
    for it in range(0,100):
        pdf=zeros(N,k)

        # Expectation
        for i in range(1,k+1):
            pdf[:,i] = normpdf(x,m[i],s[i])*p[i]
        sump = sum(pdf,2)
        for i in range(1,k+1):
            W[:,i]=pdf[:,i].div(sump)

        # Maximization
        p    = mean(W)
        for i in range(1,k+1):
            m[:,i]   = wmean(x,W[:,i])
            xm       = x-m[:,i]
            xm       = xm.mul(xm)
            v        = wmean(xm,W[:,i])
            s[:,i]   = sqrt(v)
        m.Print()

        # draw
        figure(2)
        # original data
        graph(1)
        plotgf(1,3)
        graph(2)
        plotgf(5,1)

        # GMM result
        for i in range(1,k+1):
            graph(2+i)
            plotgf(m[i],s[i],'r')
        loop.pause()
```

Ref:
Maximum Likelihood Estimation(MLE)

- Estimating parameters of a probability distribution
  - by maximizing a likelihood function

$$L(\theta; X) = p(X \mid \theta) = \int p(X, Z \mid \theta) dZ$$

$$Z : unobserved\ or\ latent\ data$$