Data Mining
Cluster Analysis: Advanced Concepts and Algorithms

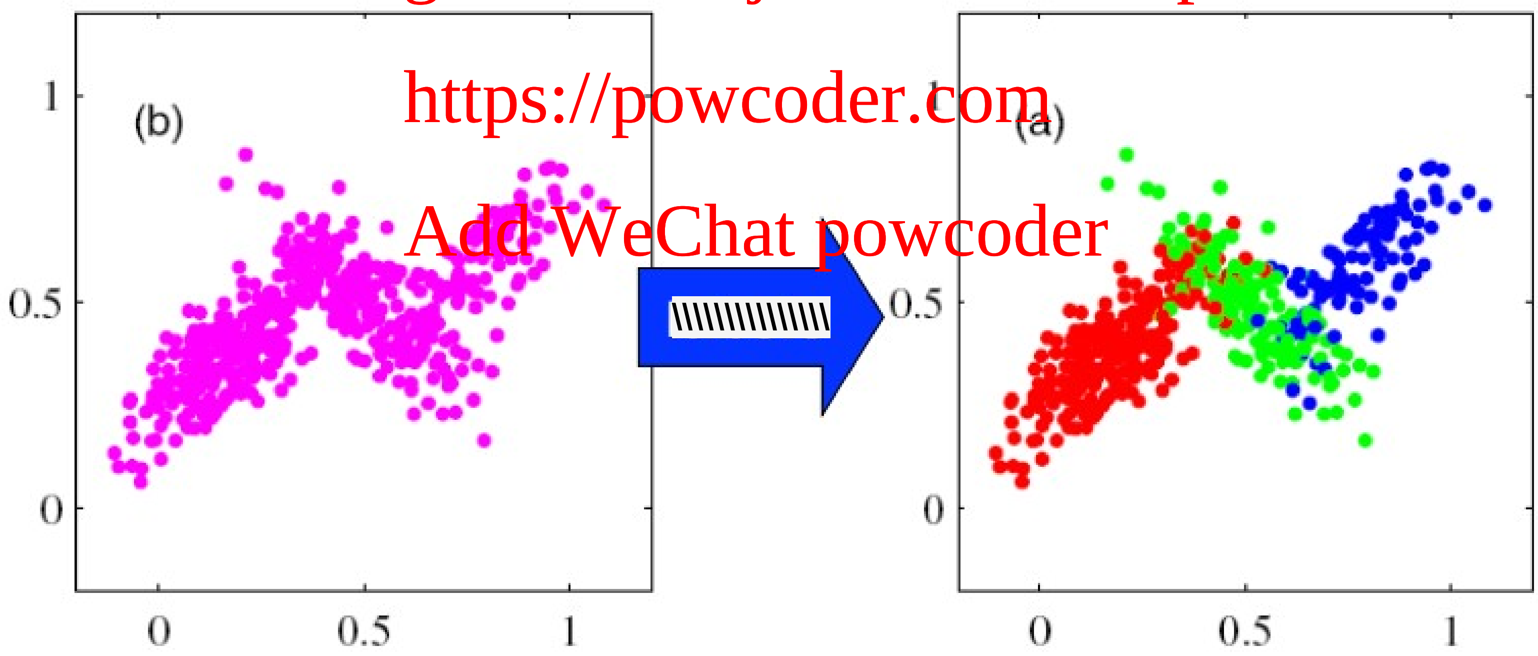
Introduction to Data Mining, 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

Outline

- Prototype-based clustering
  - Fuzzy c-means
  - Mixture Model Clustering



Hard (Crisp) vs Soft (Fuzzy) Clustering

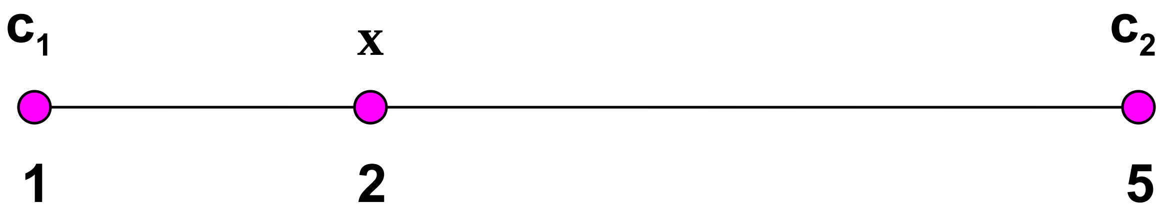
- Hard (Crisp) vs. Soft (Fuzzy) clustering
  - For soft clustering allow point to belong to more than one cluster
  - For K-means, generalize objective function

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}\, dist\left(\mathbf{x}_i, \mathbf{c}_j\right)^2 \qquad \sum_{j=1}^{k} w_{ij} = 1$$

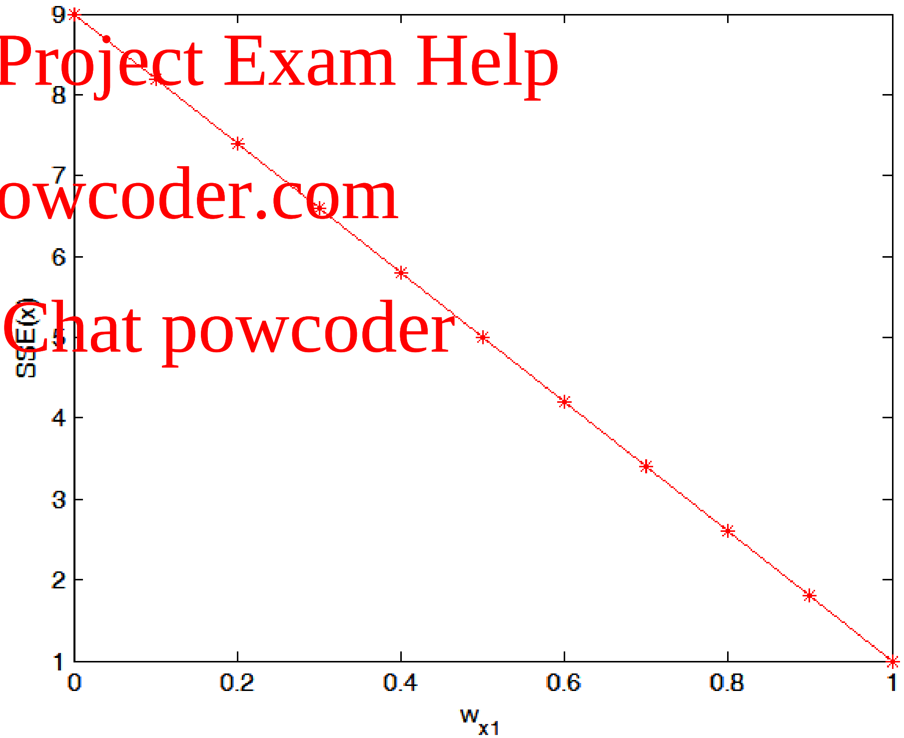
: weight with which object x_i belongs to cluster

  - To minimize SSE, repeat the following steps:
    - Fix and determine w(cluster assignment)
    - Fix w and recompute
  - Hard clustering: $w \in \{0,1\}$

Soft (Fuzzy) Clustering: Estimating Weights



$$
\begin{aligned}
SSE(x) &= w_{x1}(2-1)^2 + w_{x2}(5-2)^2 \\
&= w_{x1} + 9w_{x2}
\end{aligned}
$$



SSE(x) is minimized when w_{x1} = 1, w_{x2} = 0

Fuzzy C-means

- Objective function

p: fuzzifier (p > 1)

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{p} \, dist\left(\boldsymbol{x}_i, \boldsymbol{c}_j\right)^2 \qquad \sum_{j=1}^{k} w_{ij} = 1$$

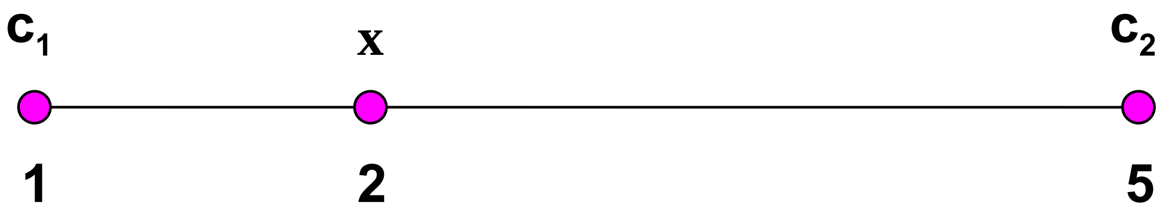
- : weight with which object belongs to cluster
- a power for the weight not a superscript and controls how “fuzzy” the clustering is

– To minimize objective function, repeat the following:
  - Fix and determine w
  - Fix w and recompute

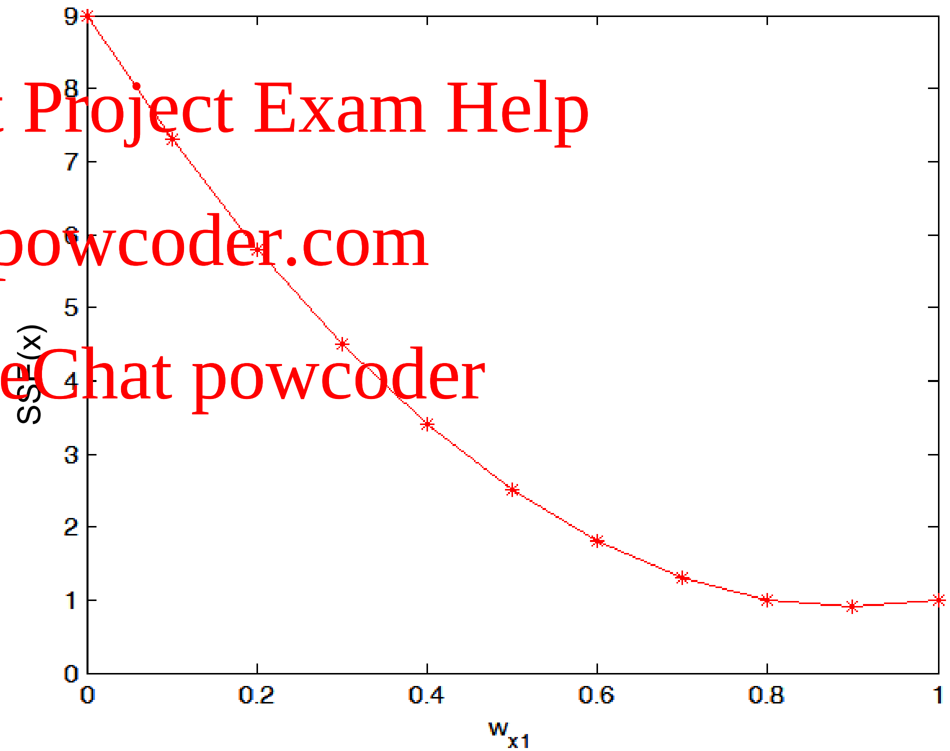
– Fuzzy c-means clustering: $w \in [0,1]$

Bezdek, James C. *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, 1981.

Fuzzy C-means



$$SSE(x) = w_{x1}^2(2-1)^2 + w_{x2}^2(5-2)^2$$
$$= w_{x1}^2 + 9w_{x2}^2$$



SSE(x) is minimized when w_{x1} = 0.9, w_{x2} = 0.1

Fuzzy C-means

- Objective function:

p: fuzzifier ($p > 1$)

$$SSE = \sum_{j=1}^{k} \sum_{i=1}^{m} w_{ij}^{p} \, dist(\boldsymbol{x}_i, \boldsymbol{c}_j)^2 \qquad \sum_{j=1}^{k} w_{ij} = 1$$

- Initialization: choose the weights w_{ij} randomly

- Repeat:

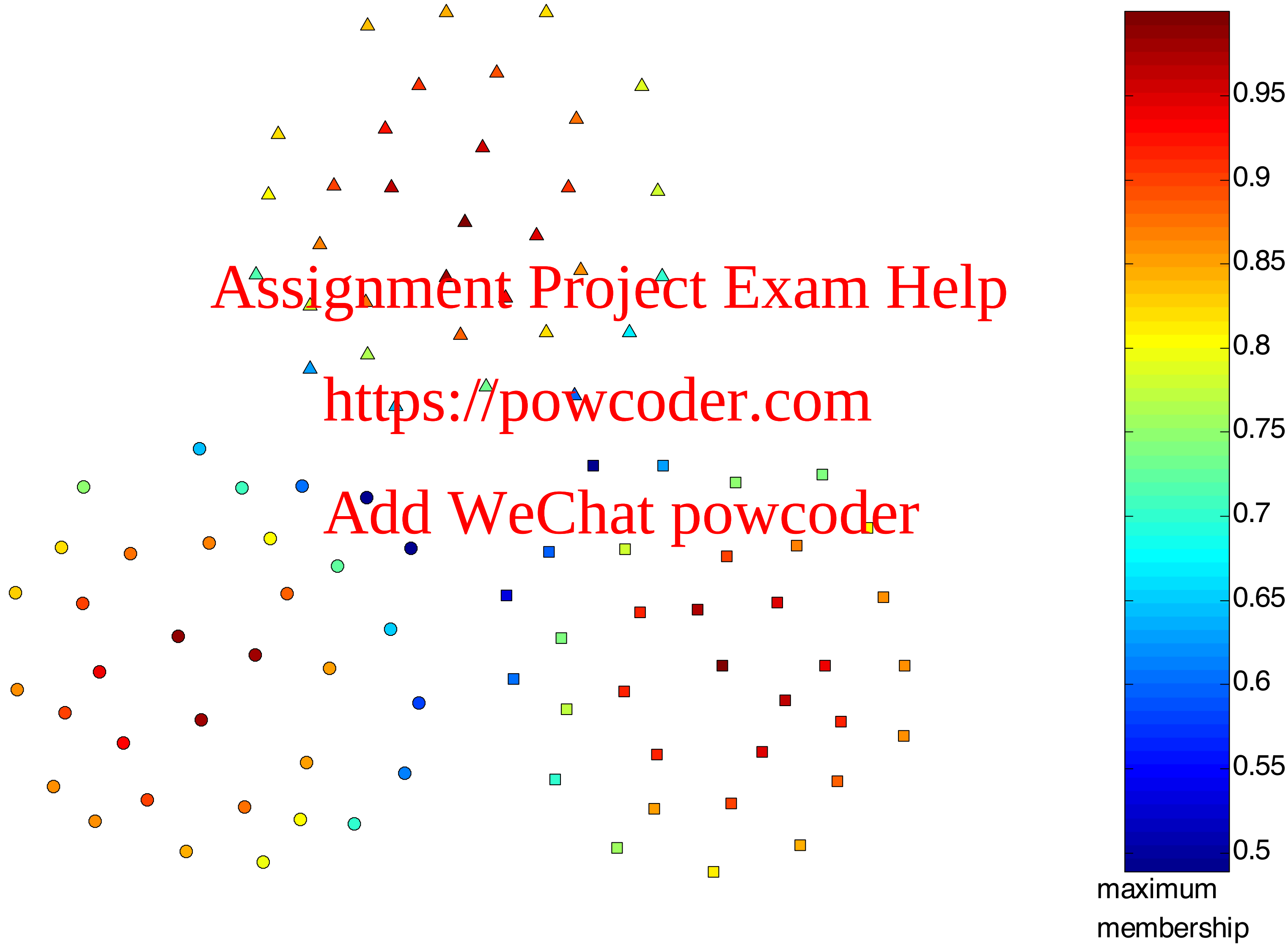
  – Update centroids:

$$\boldsymbol{c}_j = \sum_{i=1}^{m} w_{ij} \boldsymbol{x}_i / \sum_{i=1}^{m} w_{ij}$$

  – Update weights:

$$w_{ij} = \left(1/dist(\boldsymbol{x}_i, \boldsymbol{c}_j)^2\right)^{\frac{1}{p-1}} / \sum_{j=1}^{k} \left(1/dist(\boldsymbol{x}_i, \boldsymbol{c}_j)^2\right)^{\frac{1}{p-1}}$$

Fuzzy K-means Applied to Sample Data

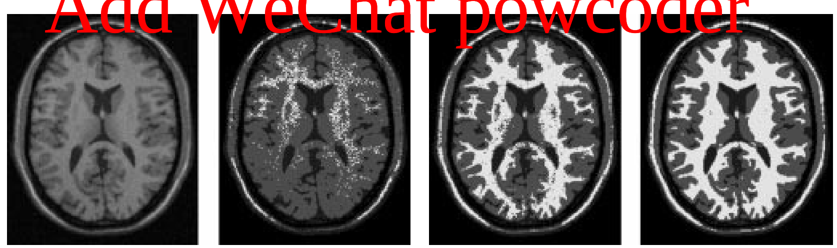


An Example Application: Image Segmentation

- Modified versions of fuzzy c-means have been used for image segmentation
  - Especially fMRI images (functional magnetic resonance images)

- References

  - Gong, Maoguo, Yan Liang, Jiao Shi, Wenping Ma, and Jingjing Ma. "Fuzzy c-means clustering with local information and kernel metric for image segmentation." *Image Processing, IEEE Transactions on* 22, no. 2 (2013): 573-584.



From left to right: original images, fuzzy c-means, EM, BCFCM

  - Ahmed, Mohamed N., Sameh M. Yamany, Nevin Mohamed, Aly A. Farag, and Thomas Moriarty. "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data." *Medical Imaging, IEEE Transactions on* 21, no. 3 (2002): 193-199.

Hard (Crisp) vs Soft (Probabilistic) Clustering

- Idea is to model the set of data points as arising from a mixture of distributions
  - Typically, normal (Gaussian) distribution is used
  - But other distributions have been very profitably used

- Clusters are found by estimating the parameters of the statistical distributions

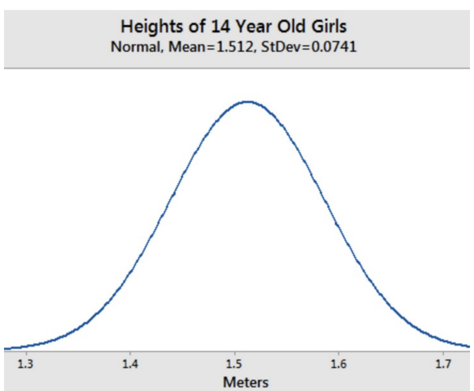
  - Can use a k-means like algorithm, called the Expectation-Maximization (EM) algorithm, to estimate these parameters
    - Actually, k-means is a special case of this approach
  - Provides a compact representation of clusters
  - The probabilities with which point belongs to each cluster provide a functionality similar to fuzzy clustering.

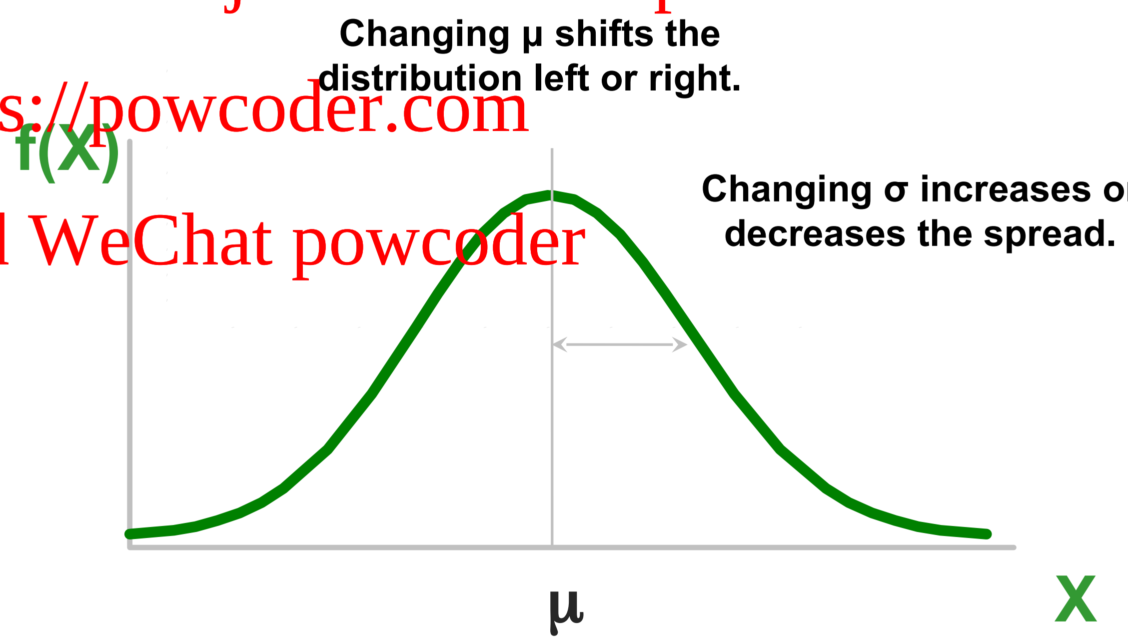
The Normal Distribution

Formula

$$\sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N}}$$

Data. Heights of 14-year-old girls: 1.34, 1.5, 1.43, 1.52, 1.60, 1.58, 1.49, ….





μ: The mean/median/expectation

σ (sigma): Standard deviation

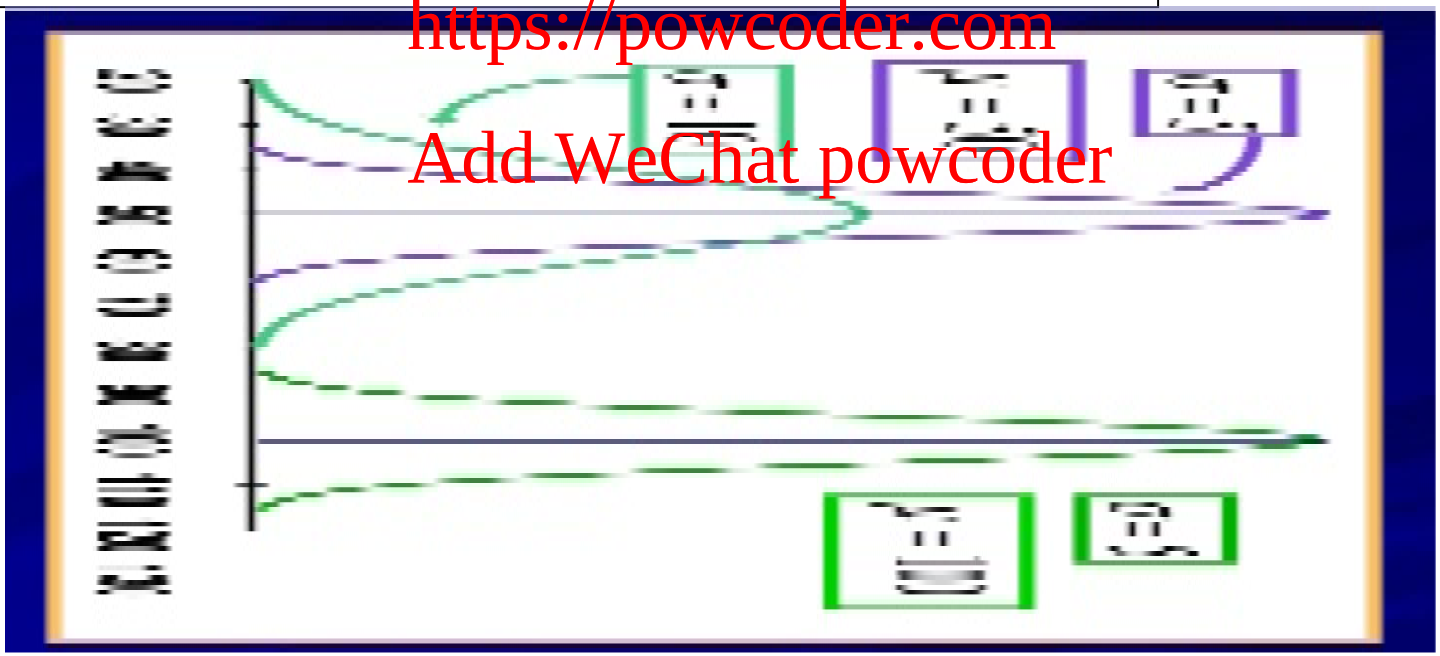
Probability density function (PDF)

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Note constants:

π=3.14159

e=2.71828



Multivariate Gaussian

Each data point has multiple variables (e.g. height and weight of a person)

- Multivariate Gaussian

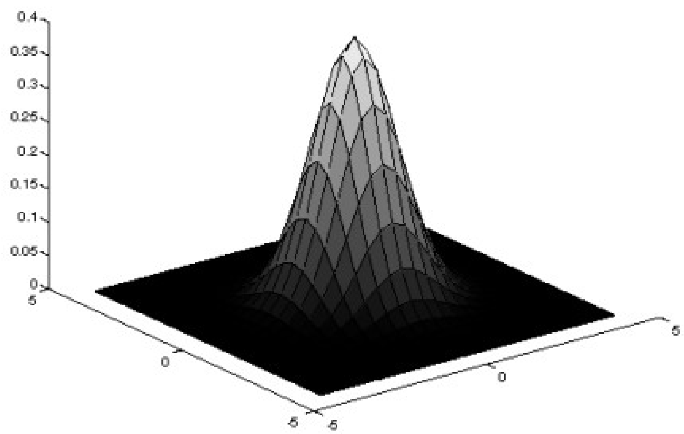
$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\Sigma) = \frac{1}{((2\pi)^d|\Sigma|)^{1/2}} \exp\underbrace{\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\top}\Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\}}_{\text{Evaluates to a number}}$$

mean
dx1 vector

covariance
dxd matrix

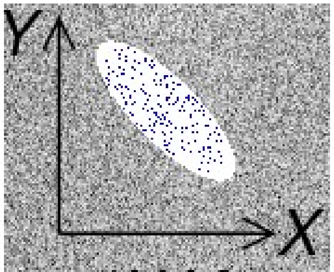
d: number of dimensions (features)

e.g. Bivariate Gaussian

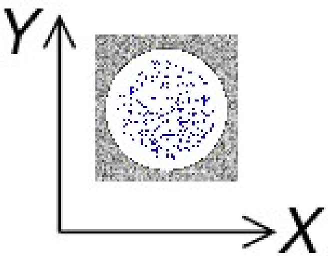


(to observe the shapes, use: http://personal.kenyon.edu/hartlaub/MellonProject/Bivariate2.html)

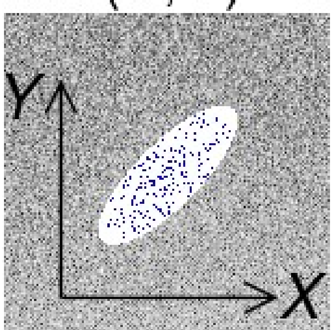
Bivariant Gaussian distribution (scatter plot)



$\mathrm{cov}(X,Y) < 0$

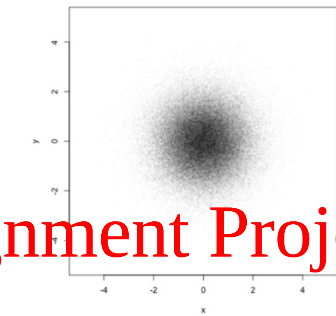


$\mathrm{cov}(X,Y) \approx 0$

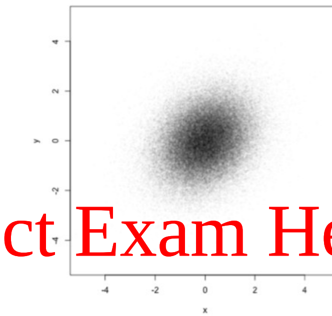


$\mathrm{cov}(X,Y) > 0$

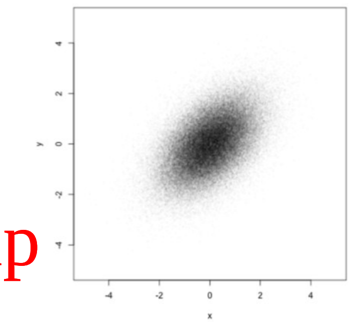
https://en.wikipedia.org/wiki/Covariance



$X \sim \mathcal{N}(0,1),\ Y \sim \mathcal{N}(0,1)$

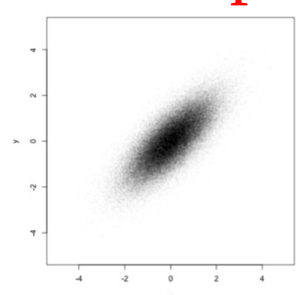


$X \sim \mathcal{N}(0,1),\ Y \sim \mathcal{N}(0,1)$

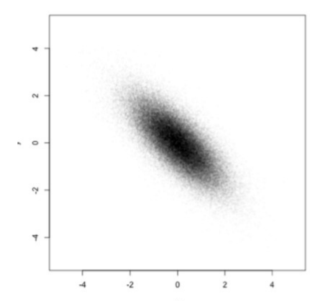


$X \sim \mathcal{N}(0,1),\ Y \sim \mathcal{N}(0,1)$

From left to right, increase of COV(x, y) from 0 to bigger positive numbers



$X \sim \mathcal{N}(0,1),\ Y \sim \mathcal{N}(0,1)$



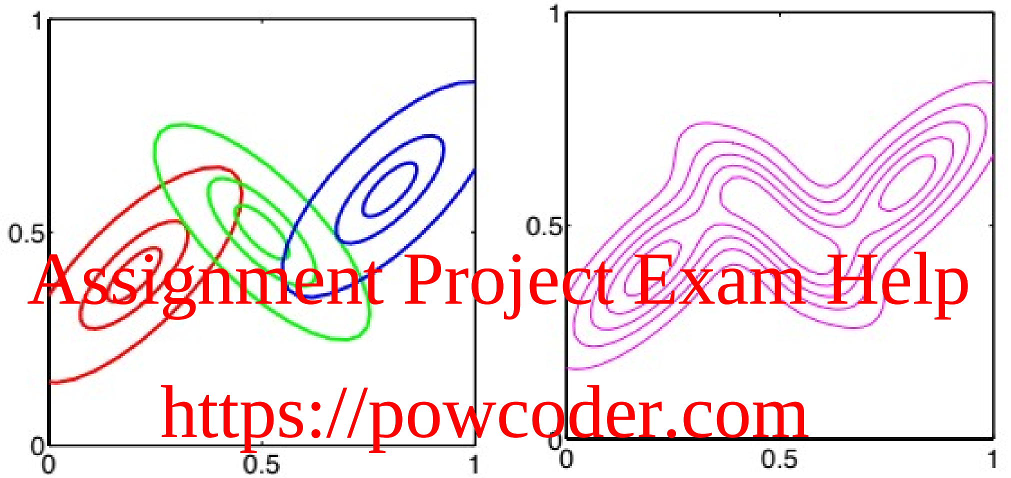
$X \sim \mathcal{N}(0,1),\ Y \sim \mathcal{N}(0,1)$

Negative covariance

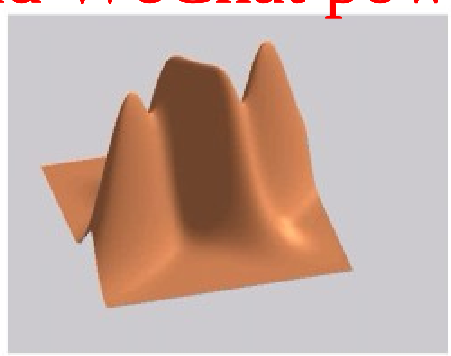
Mixture of 3 Gaussians

$p(\mathbf{x}|k)$

K=1,2,3



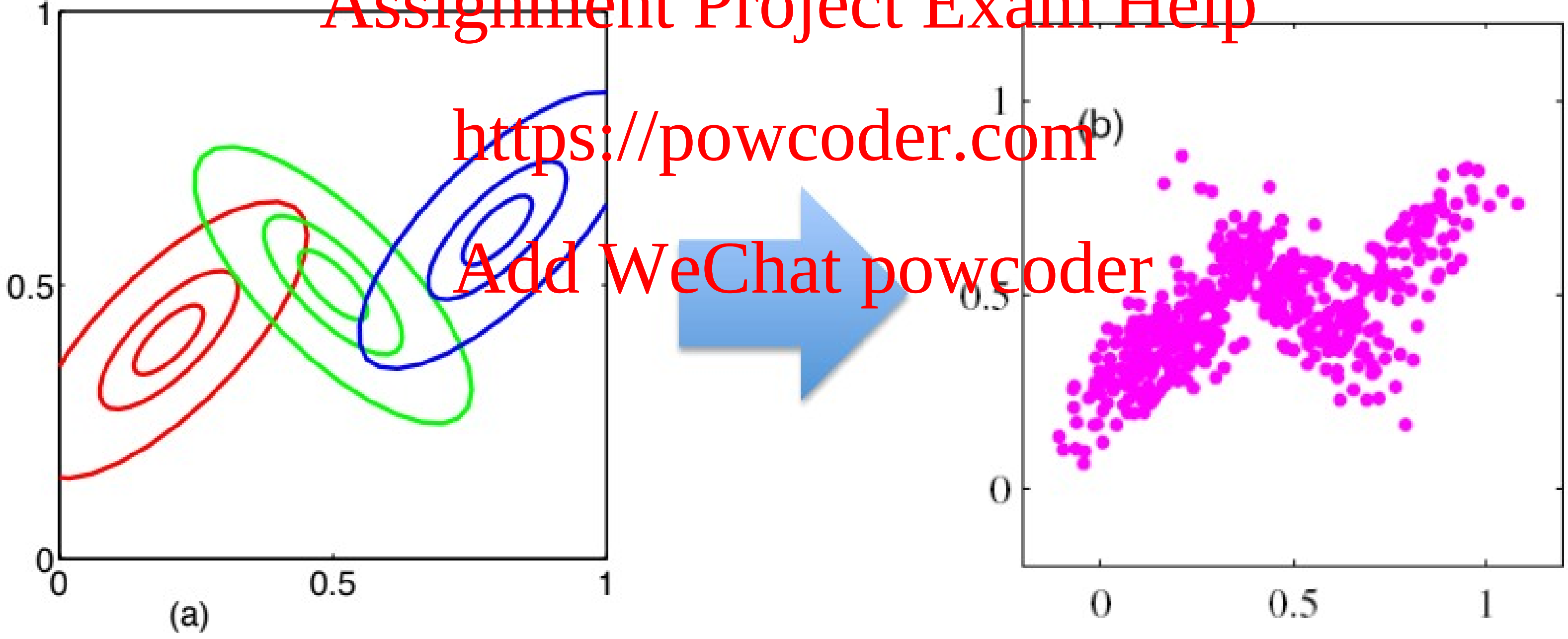
$p(\mathbf{x})$



Probability density function (pdf) of $p(\mathbf{x})$

Sampling from a mixture model

$$p(\mathbf{x}) = \sum_{k=1}^{K} \boxed{\pi_k} \mathcal{N}(\mathbf{x}|\boxed{\boldsymbol{\mu}_k}, \boxed{\boldsymbol{\Sigma}_k})$$



Sampling from a Mixture model

Generate u = uniform random number between 0 and 1

If $u < \pi_1$

generate $x \sim N(x \mid \mu_1, \Sigma_1)$

elseif $u < \pi_1 + \pi_2$

generate $x \sim N(x \mid \mu_2, \Sigma_2)$

⋮

elseif $u < \pi_1 + \pi_2 + \ldots + \pi_{K-1}$

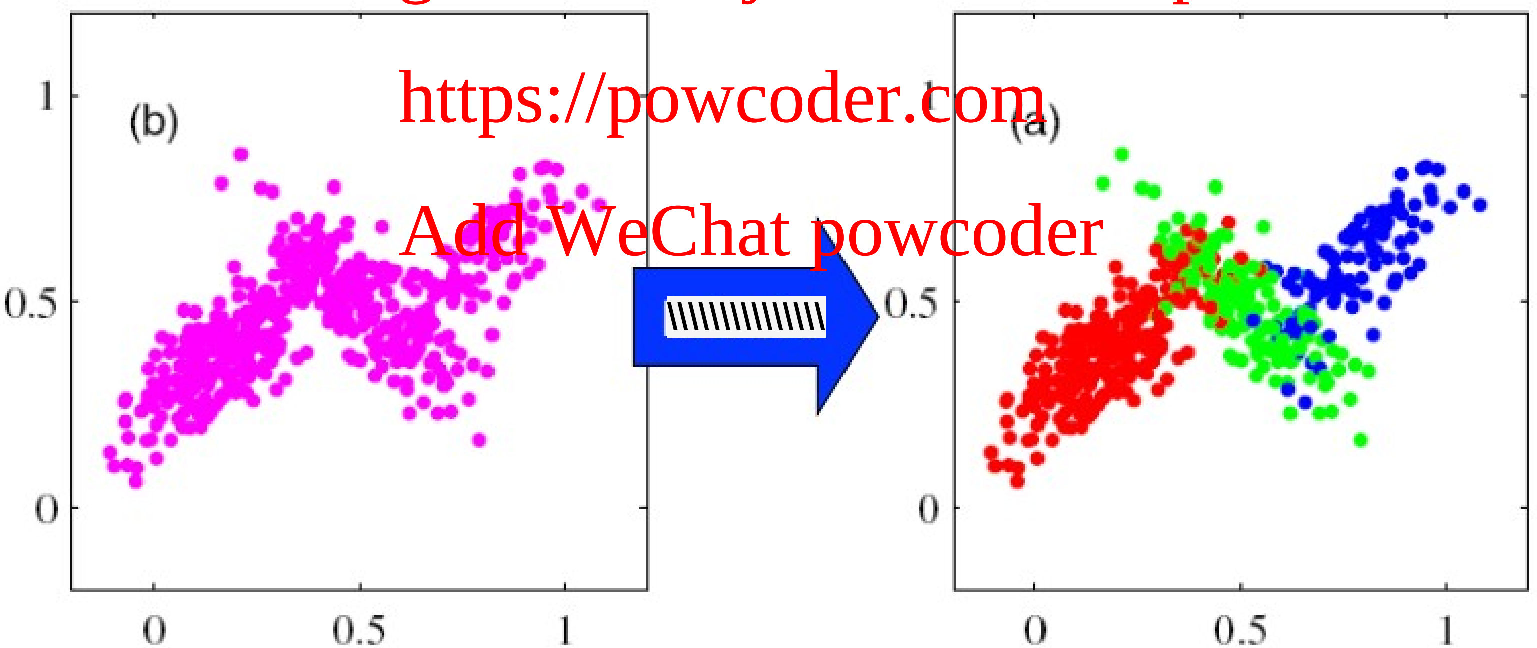
generate $x \sim N(x \mid \mu_{K-1}, \Sigma_{K-1})$

else

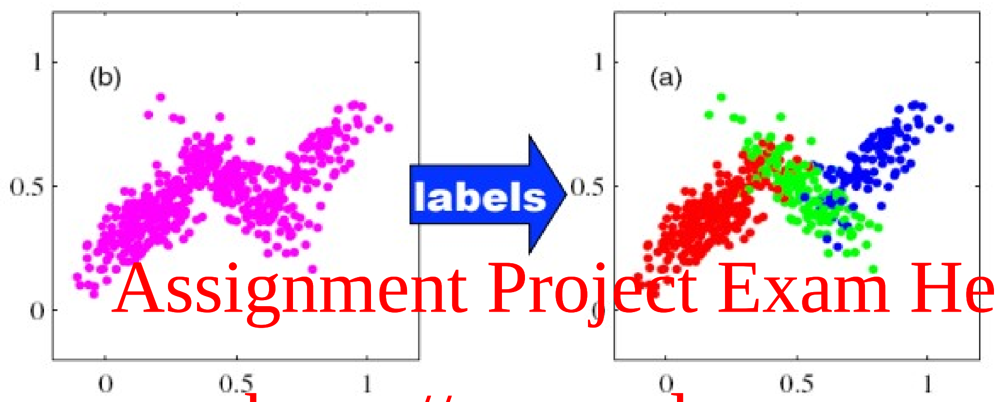
generate $x \sim N(x \mid \mu_K, \Sigma_K)$

Clustering problem using Gaussian mixture model

- Given a set of training data, which are produced by Gaussian mixture models, how can you figure out each component Gaussian distribution?

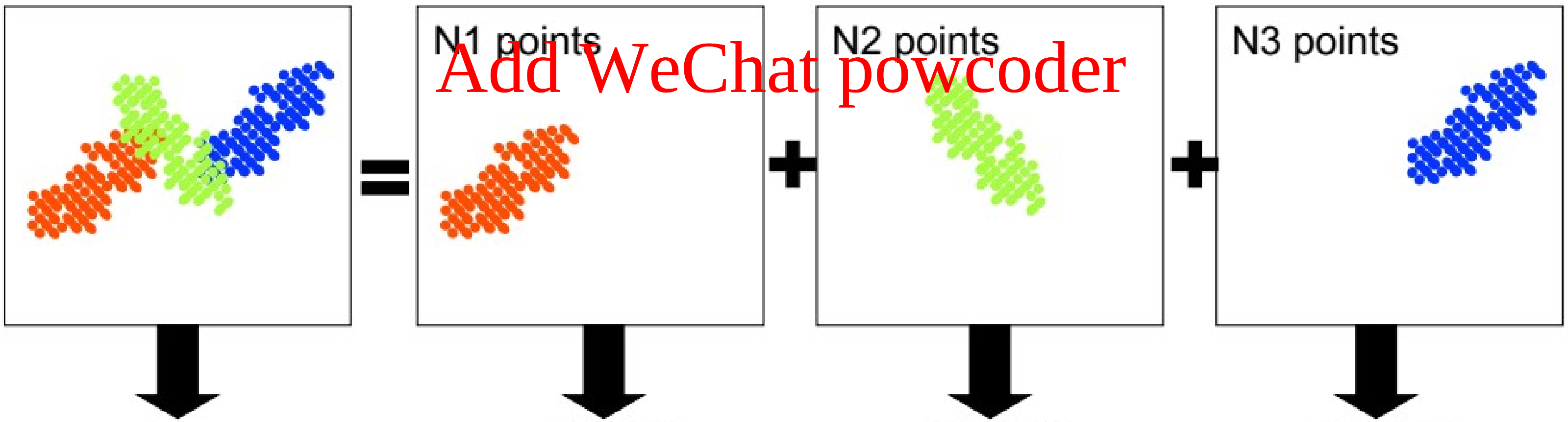


EM-based clustering algorithm (Expectation Maximization)



1. Maximization: Using maximum likelihood to estimate the parameters of all the Gaussian distributions
2. Expectation: compute the weight that each data point belongs to every point belongs to every distribution

And we can easily estimate each Gaussian, along with the mixture weights!



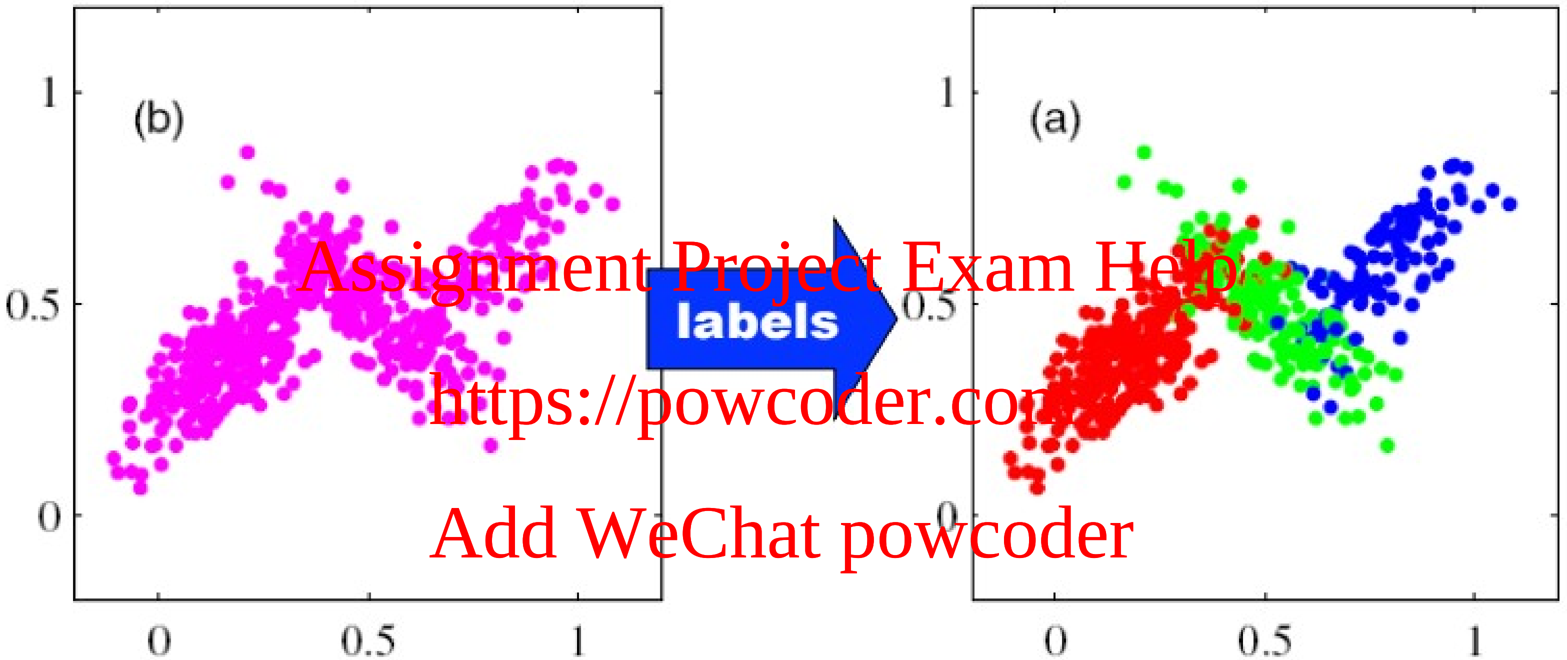
estimate
pi_k = Nk / N

estimate
mu_1, Sigma_1

estimate
mu_2, Sigma_2

estimate
mu_3, Sigma_3

Expectation stage



The initial estimated parameters are not accurate because you don't really know the "membership" of these data points. So, in the expectation stage, we will re-evaluate the membership distribution by computing a "latent" variable z_nk ( ).

For hard clustering, z_nk = 1 if the data point n comes from the kth component Gaussian (or 0 if not). For soft/fuzzy clustering, z_nk is the weight of the data point n comes from the kth component Gaussian. Thus, it is between 0 and 1.

The sketch of the EM Algorithm

What EM proposes to do:

1) compute p(Z|X,theta), the posterior distribution over z_nk, given our current best guess at the values of theta

2) compute the expected value of the log likelihood ln(p(X,Z|theta)) with respect to the distribution p(Z|X,theta)

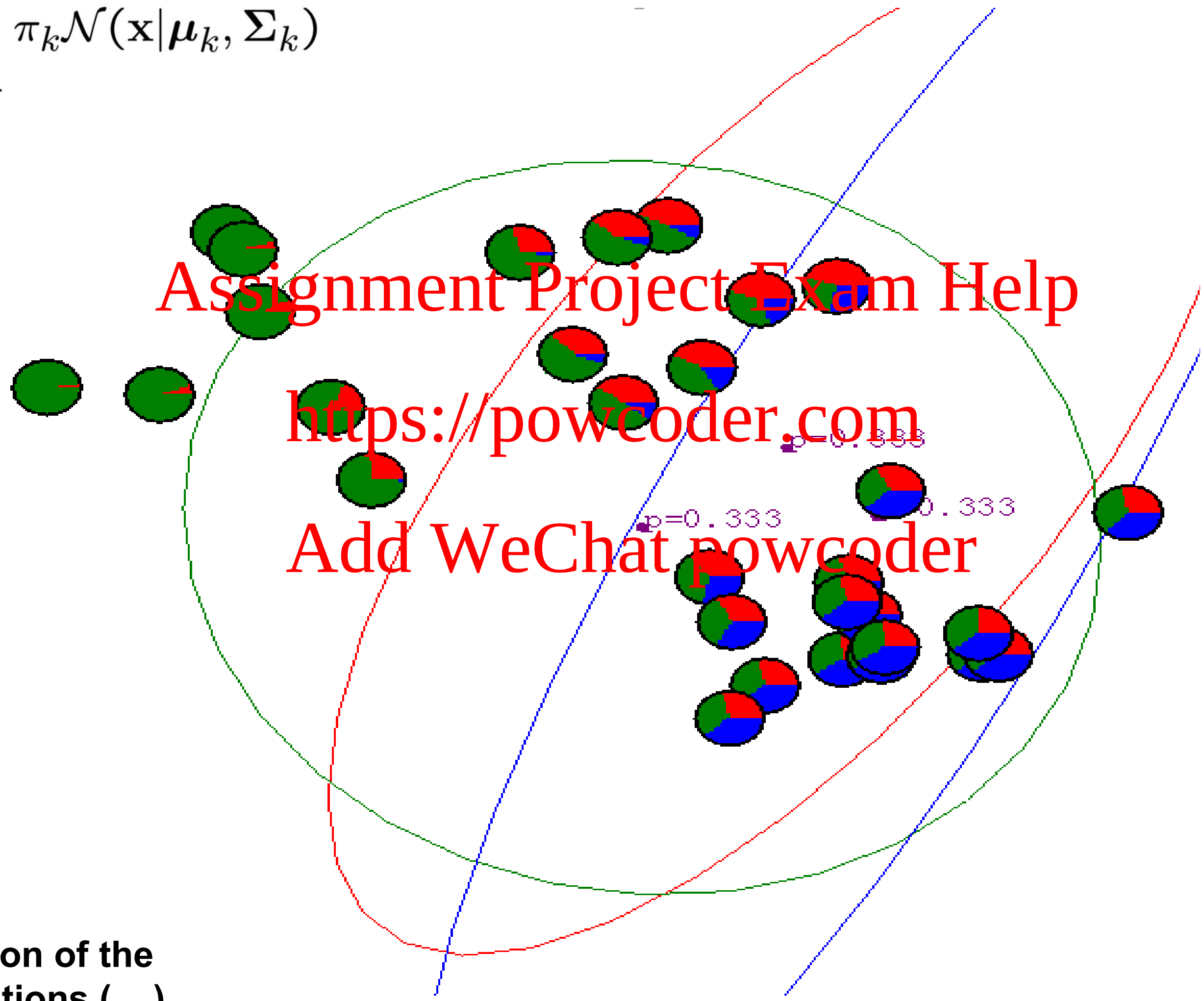
3) find theta_new that maximizes that function. This is our new best guess at the values of theta.

4) iterate...

Theta is the parameters for a Gaussian distribution. Z is the contribution of each sample to a model

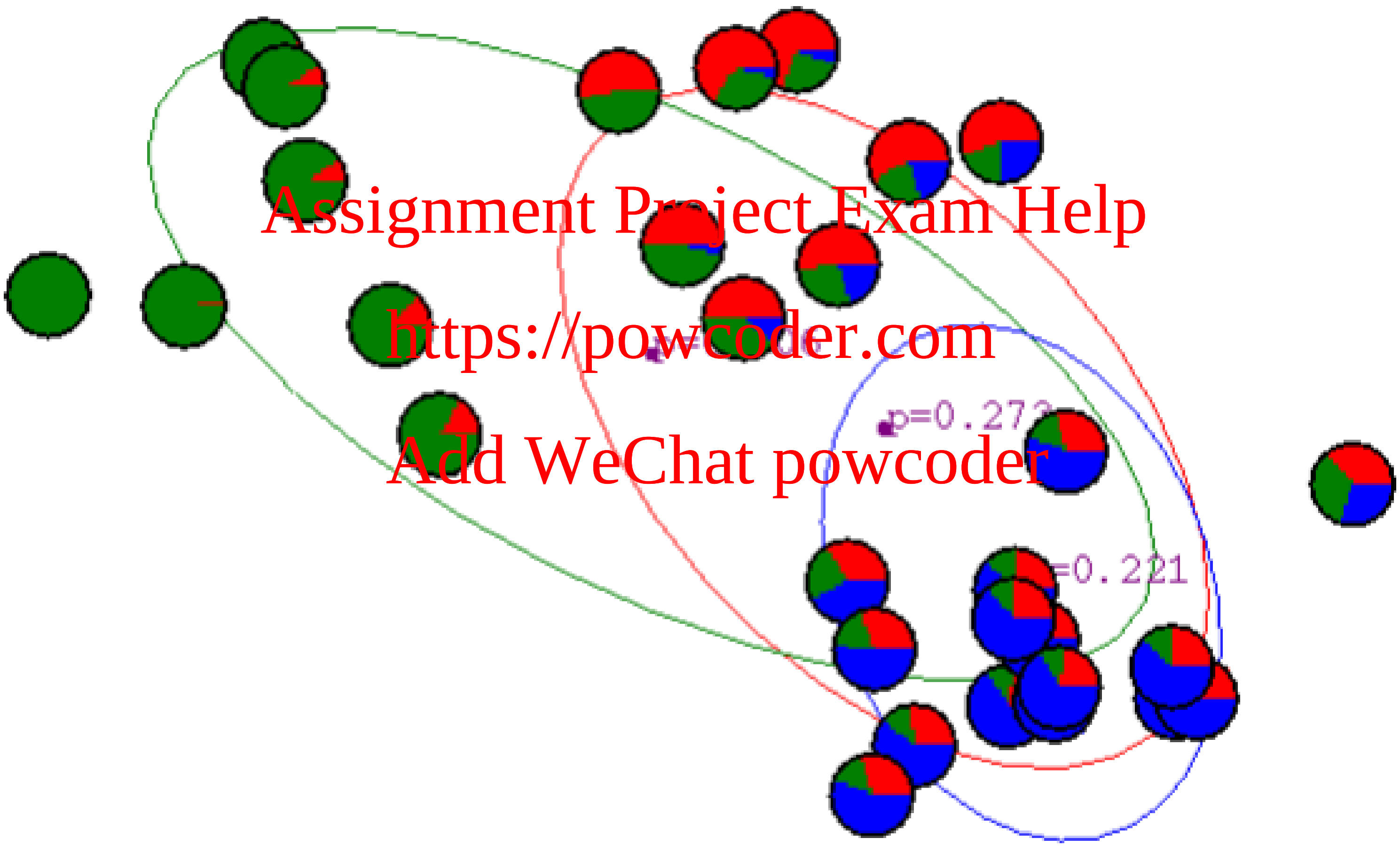
Gaussian Mixture example

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

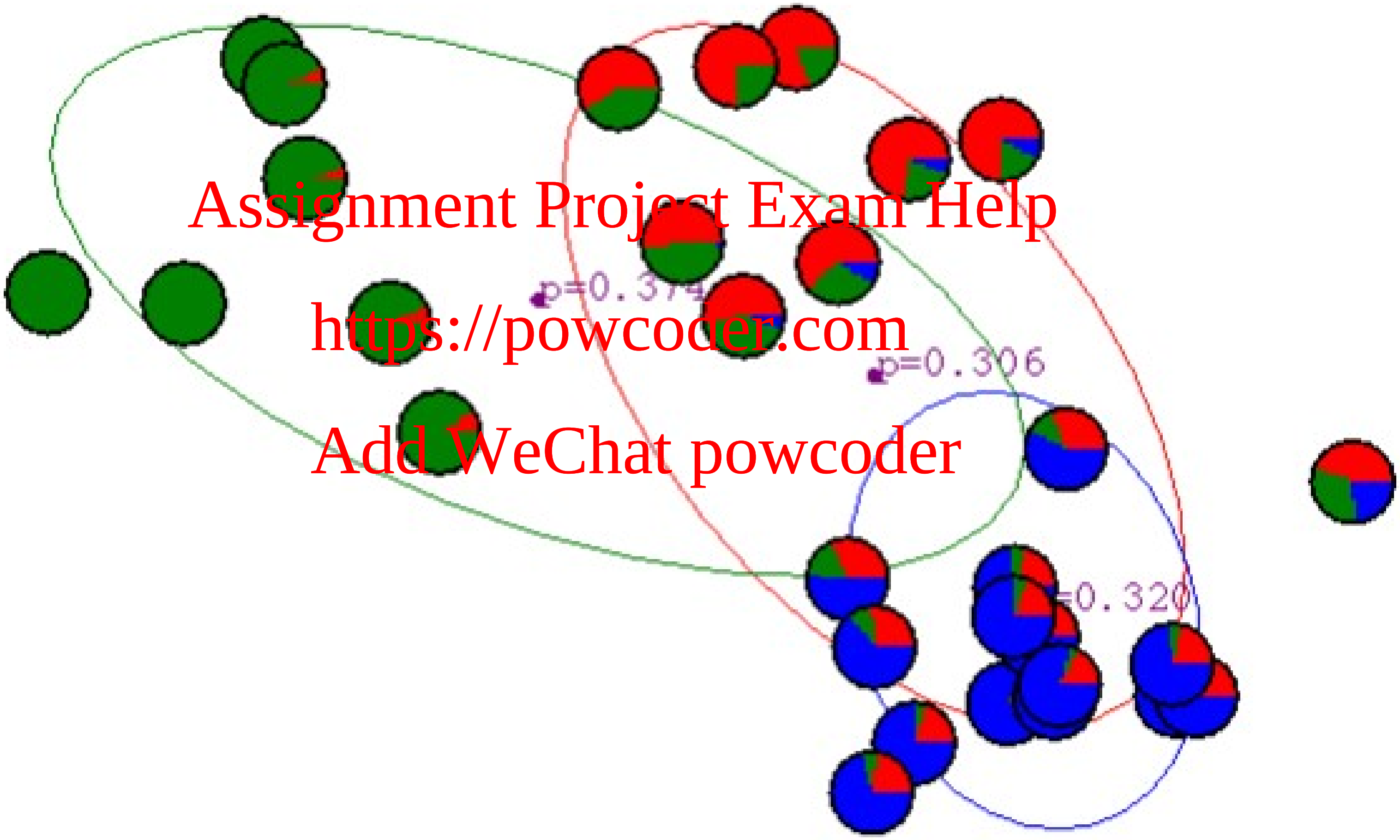


p is the portion of the three distributions ( )

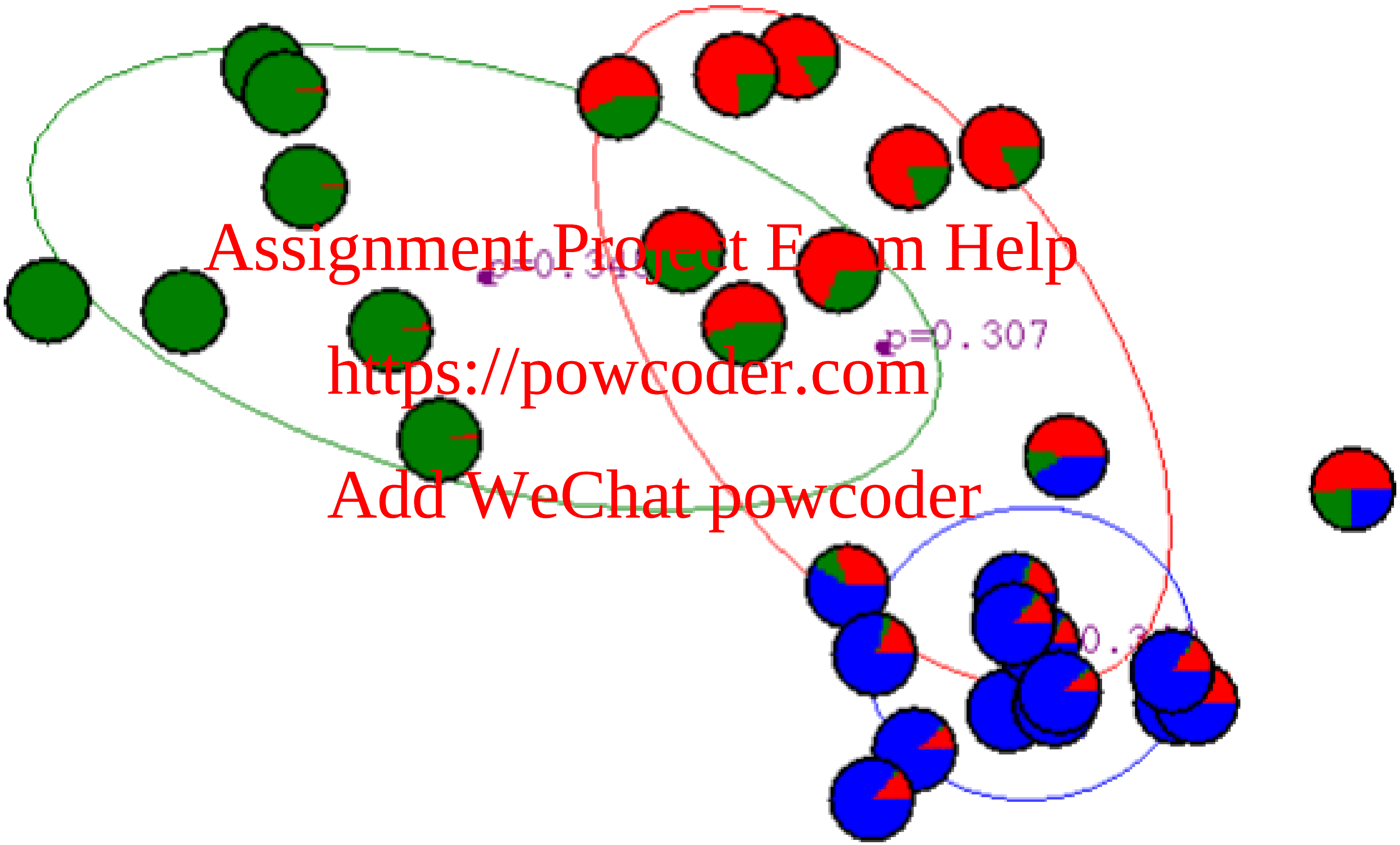
After 1st iteration



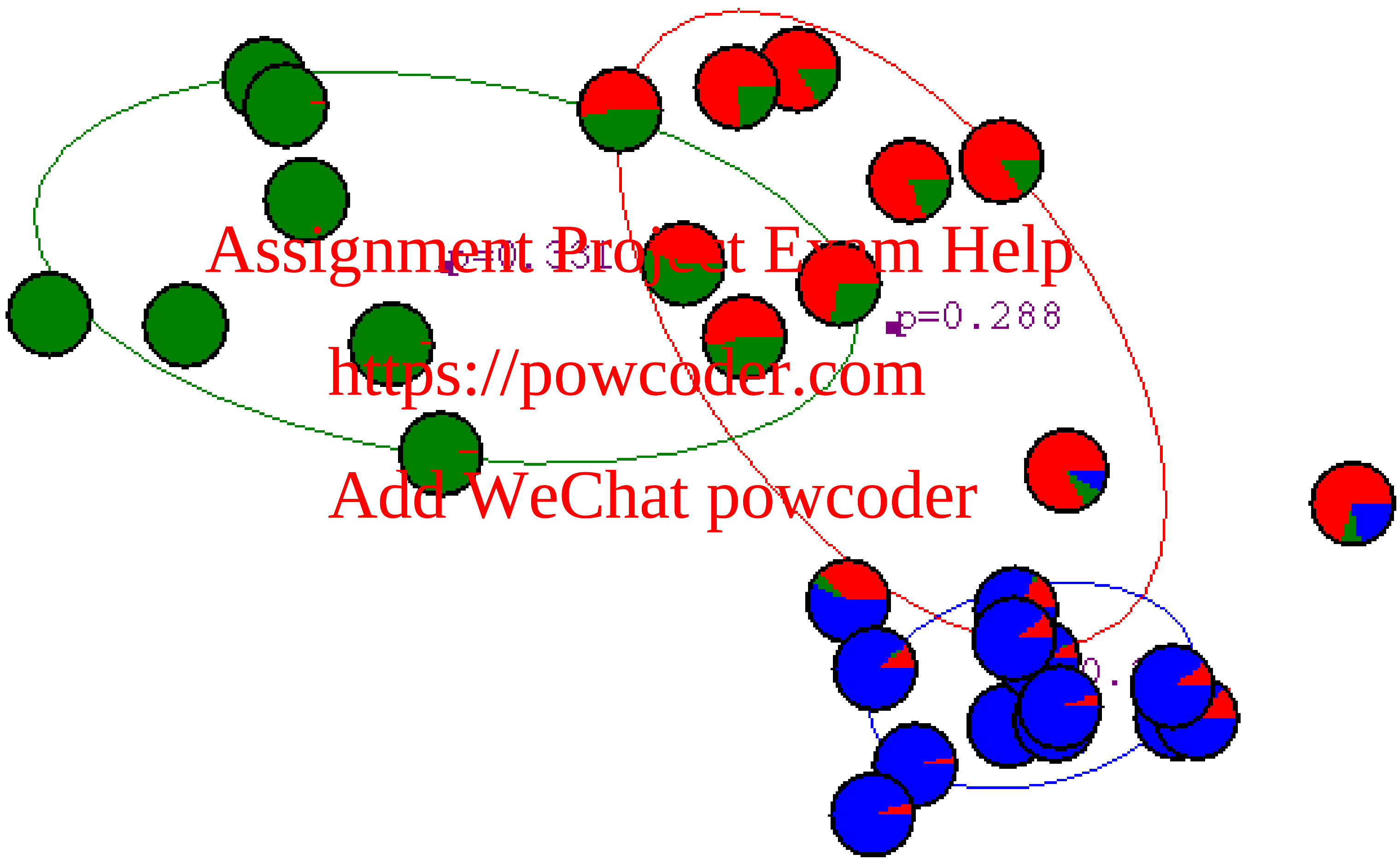
After 2nd iteration



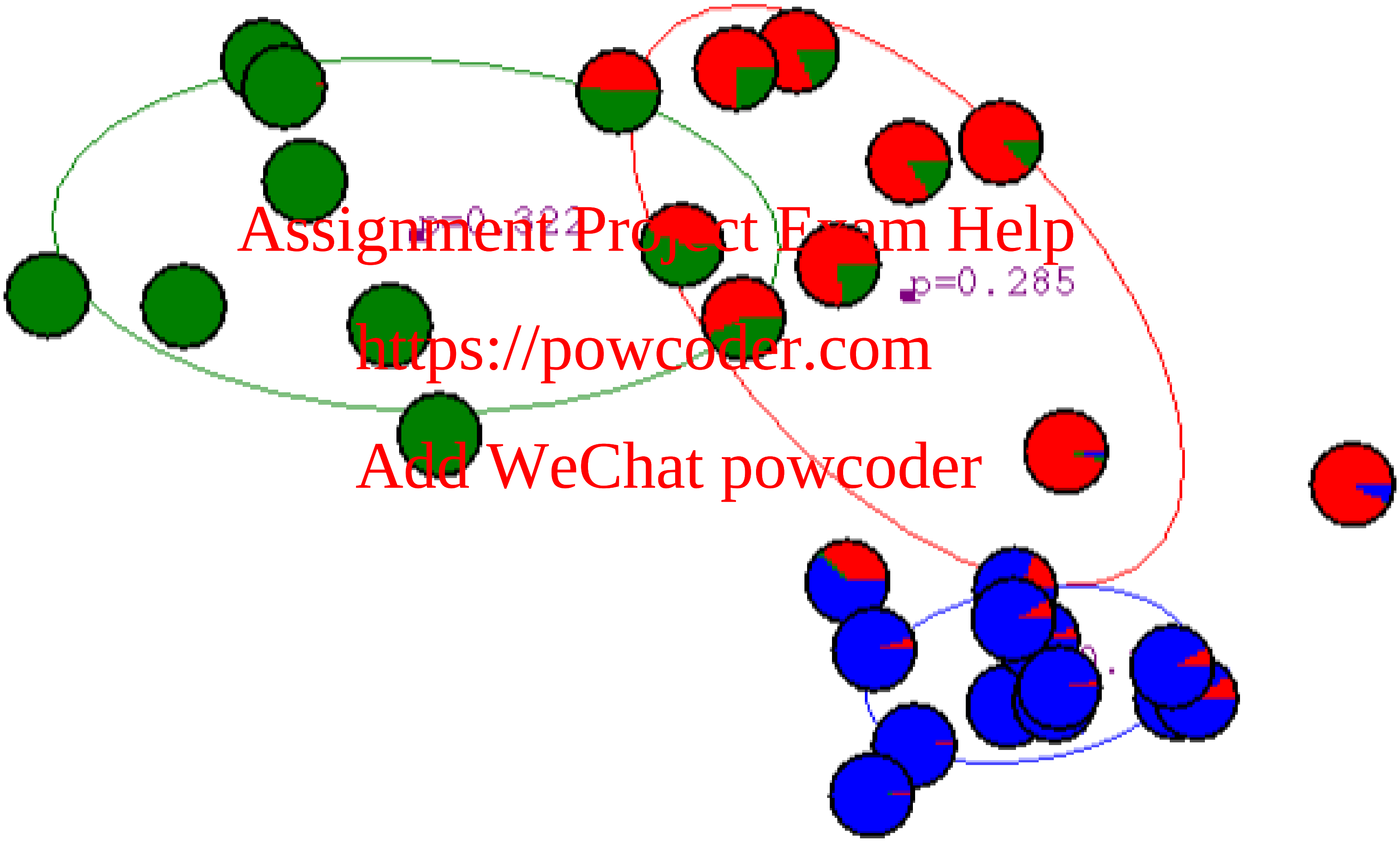
After 3rd iteration



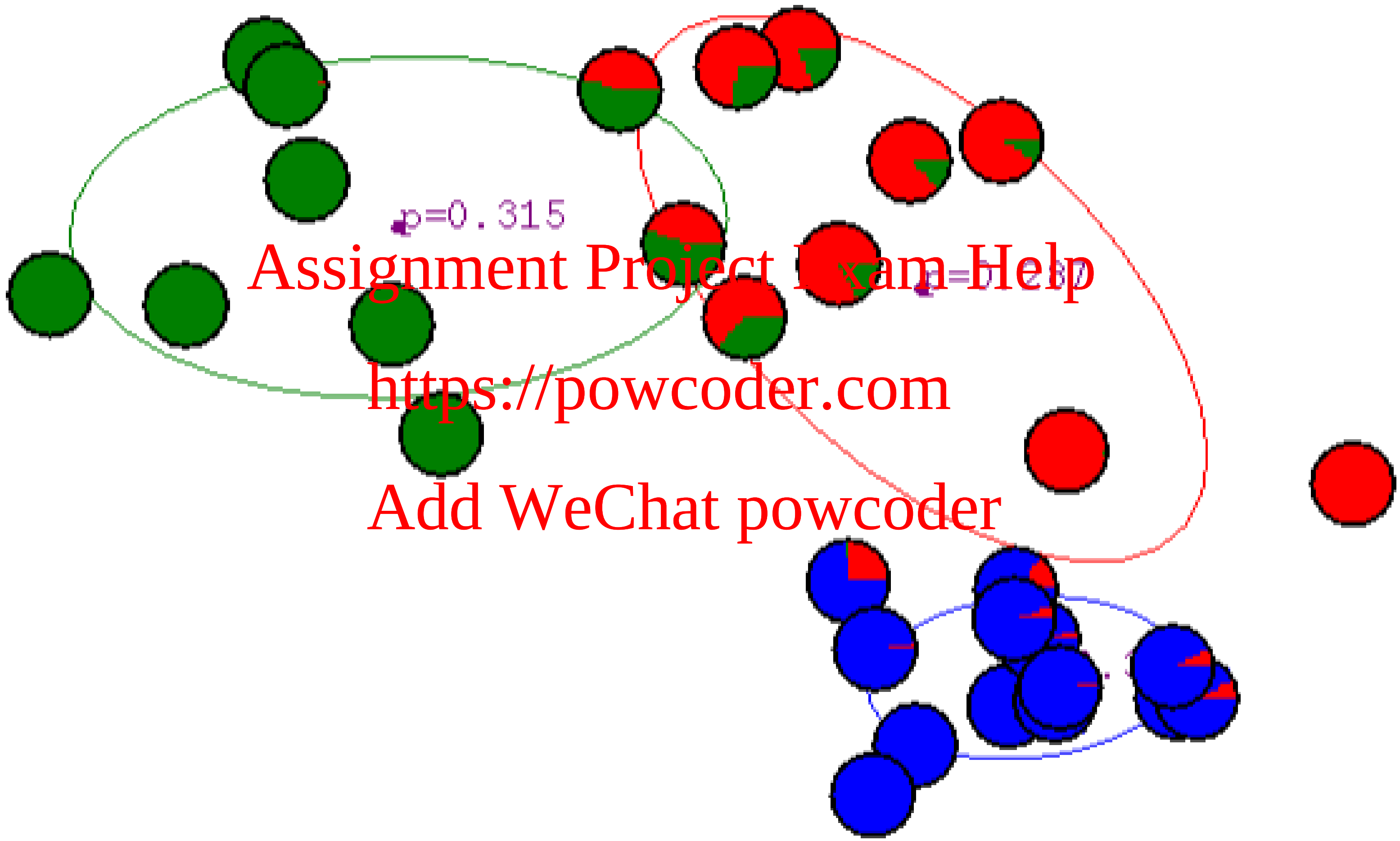
After 4th iteration



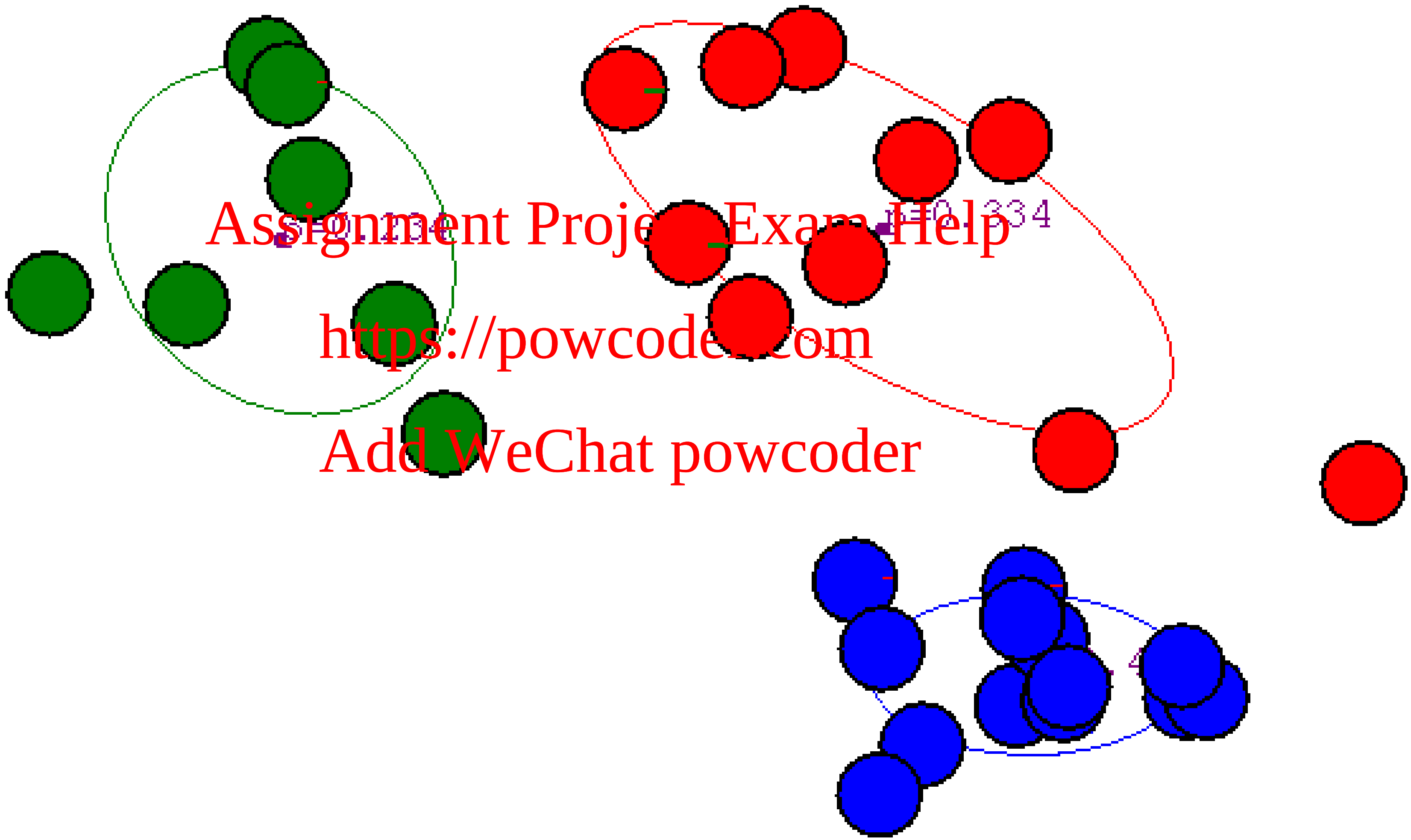
After 5th iteration



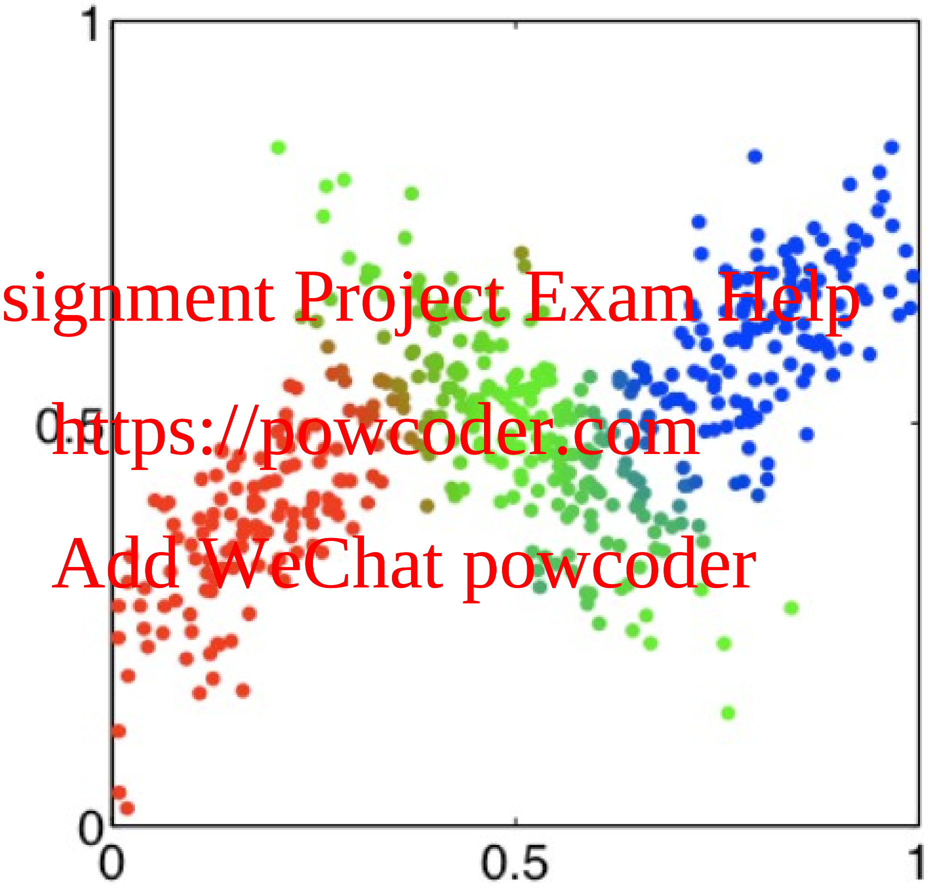
After 6^{th} iteration



After 20th iteration



EM produces "soft" labeling



each point makes a weighted contribution
to the estimation of ALL components

Formal equations of EM for GMMs

E

$$\gamma(z_{nj}) = \frac{\pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \Sigma_j)}{\sum_k \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \Sigma_k)}$$

ownership weights (soft labels)

M

$$\boldsymbol{\mu}_j = \frac{\sum_{n=1}^{N} \gamma(z_{nj})\,\mathbf{x}_n}{\sum_{n=1}^{N} \gamma(z_{nj})}$$

means

$$\Sigma_j = \frac{\sum_{n=1}^{N} \gamma(z_{nj})\,(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^{\top}}{\sum_{n=1}^{N} \gamma(z_{nj})}$$

covariances

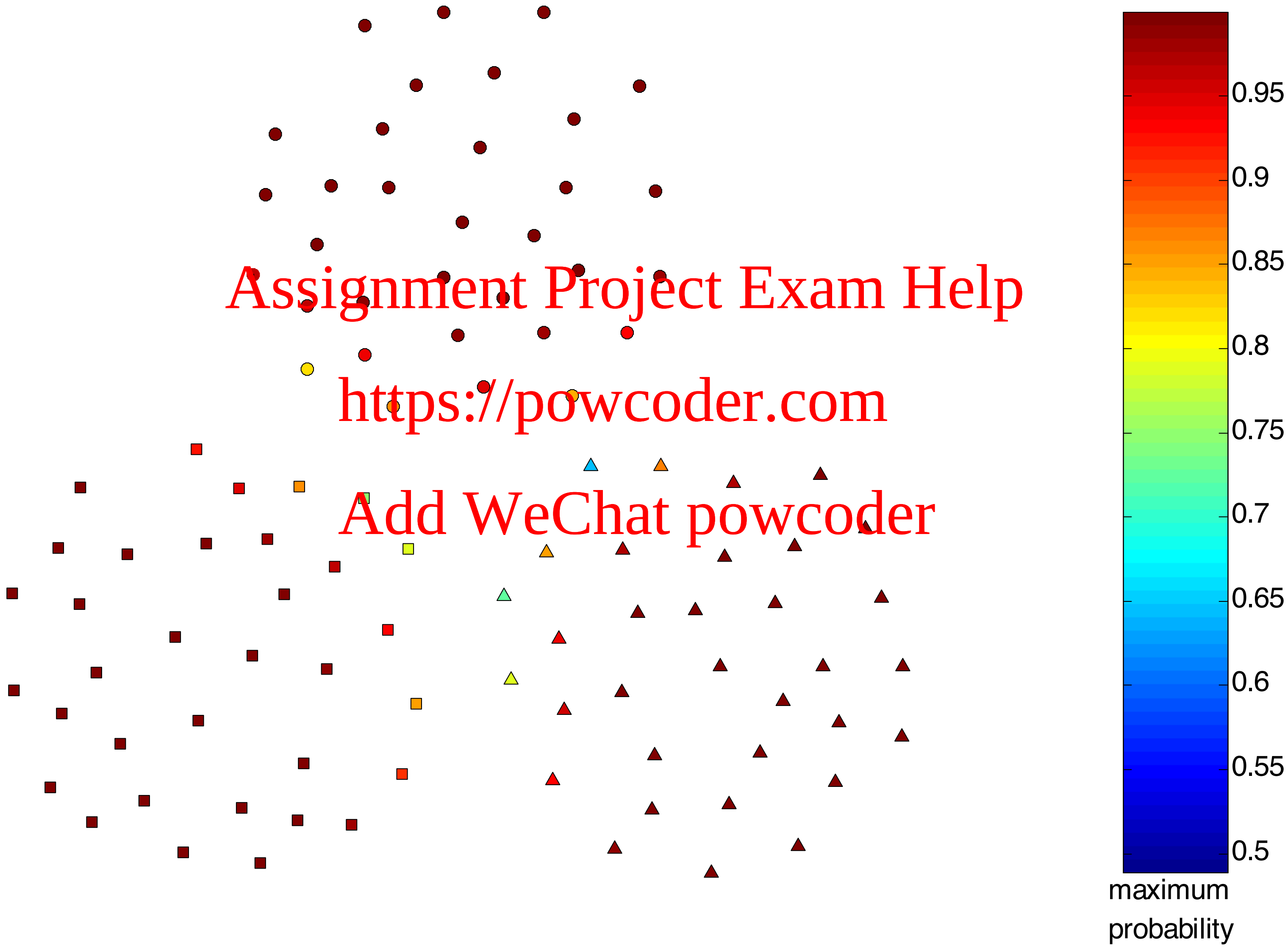
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nj})$$

mixing weights

Alternate E and M steps to convergence.

You need to know how to apply EM to one-dimensional data points

Probabilistic Clustering Applied to Sample Data



An example for 1 dimensional samples

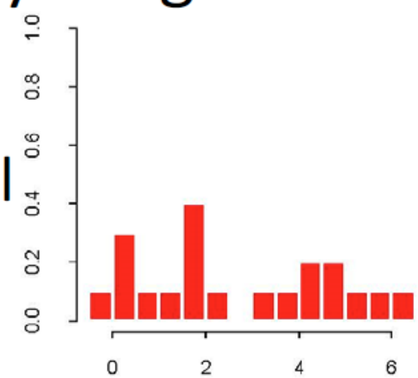
- Credit: The Elements of Statistical Learning by T. Hastie, R. Tibshirani, J. Friedman

Consider the following data set:

-0.39	0.12	0.94	1.67	1.76	2.44	3.72	4.28	4.92	5.53
0.06	0.48	1.01	1.68	1.80	3.25	4.12	4.60	5.28	6.22

- Model the density of the data points
- A simple and common way: single Gaussian model

From histogram of the data points, single Gaussian model is poor



Gaussian mixture model with 2 components

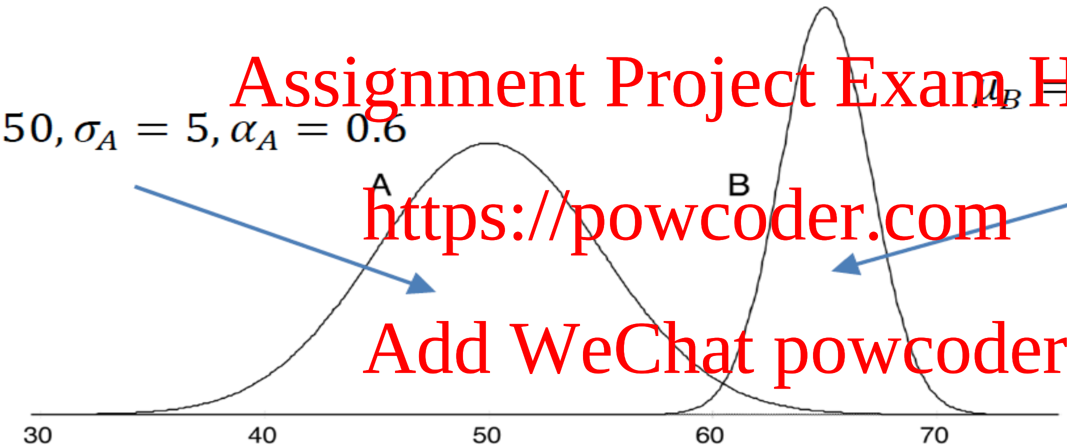
Mixture Model

Example

An example of Gaussian mixture model with 2 components.

$\mu_A = 50, \sigma_A = 5, \alpha_A = 0.6$

$\mu_B = 65, \sigma_B = 2, \alpha_B = 0.4$



Sample data points generated from the model

A	51	B	62	B	64	A	48	A	39	A	51
A	43	A	47	A	51	B	64	B	62	A	48
B	62	A	52	A	52	A	51	B	64	B	64
B	64	B	64	B	62	B	63	A	52	A	42
A	45	A	51	A	49	A	43	B	63	A	48
A	42	B	65	A	48	B	65	B	64	A	41
A	46	A	48	B	62	B	66	A	48		
A	45	A	49	A	43	B	65	B	64		
A	45	A	46	A	40	A	46	A	48		

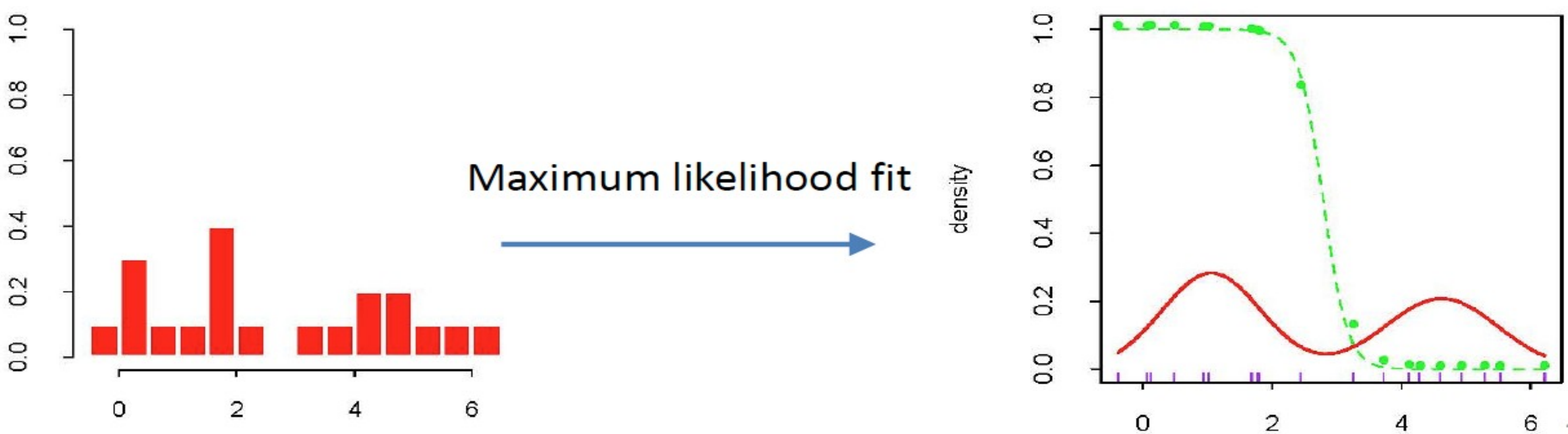
There are 32 A, 19 B. B: mean=1209/19=63.6. What is the standard deviation of B?

Mixture model learning

Sample Result

- Due to the apparent bi-modality
  → Single Gaussian distribution would not be appropriate

- A simple mixture model for density estimation
- Associated EM algorithm for carrying out maximum likelihood estimation



Green: membership weight

Goal: figure out the two distributions

- Two separate underlying regimes
  → instead model Y as mixture of two normal distributions:
  $$Y_1 \sim N(\mu_1, \sigma_1^2)$$
  $$Y_2 \sim N(\mu_2, \sigma_2^2)$$
  $$Y = (1 - \Delta) \cdot Y_1 + \Delta \cdot Y_2$$
  where $\Delta \in \{0, 1\}$ with $\Pr(\Delta = 1) = \pi$
- Generative representation is explicit: generate a $\Delta \in \{0, 1\}$ with probability π
- Depending on outcome, deliver Y_1 or Y_2

Generate u = uniform random number between 0 and 1
If $u < \pi_1$
  generate $x \sim N(x \mid \mu_1, \Sigma_1)$
elseif $u < \pi_1 + \pi_2$
  generate $x \sim N(x \mid \mu_2, \Sigma_2)$

Latent variable

- Maximum likelihood estimates:
  μ_1 and σ_1^2 - sample mean and variance for those data with $\Delta_i = 0$
  μ_2 and σ_2^2 - sample mean and variance for those data with $\Delta_i = 1$

- Estimate of π would be the proportion of $\Delta_i = 1$

- Δ_i is unknown → iterative fashion, substituting for each Δ_i in its expected value
  $$\gamma_i(\theta) = E(\Delta_i|\theta, \boldsymbol{Z}) = \Pr(\Delta_i = 1|\theta, \boldsymbol{Z})$$
- γ_i is also called *responsibility* of model 2 for observation i

Algorithm

EM algorithm for two-component Gaussian mixtures:

1. Take initial guesses for the parameters

$$\hat{\mu}_1, \hat{\sigma}_1^2, \hat{\mu}_2, \hat{\sigma}_2^2, \hat{\pi}$$

2. Expectation Step: compute the responsibilities

$$\hat{\gamma}_i = \frac{\hat{\pi}\phi_{\hat{\theta}_2}(y_i)}{(1-\hat{\pi})\phi_{\hat{\theta}_1}(y_i) + \hat{\pi}\phi_{\hat{\theta}_2}(y_i)}, i = 1, 2, \ldots, N$$

3. Maximization Step:

Compute the weighted means and variances

$$\hat{\mu}_1 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i) y_i}{\sum_{i=1}^N (1-\hat{\gamma}_i)}, \quad \hat{\sigma}_1^2 = \frac{\sum_{i=1}^N (1-\hat{\gamma}_i)(y_i-\hat{\mu}_1)^2}{\sum_{i=1}^N (1-\hat{\gamma}_i)}$$

$$\hat{\mu}_2 = \frac{\sum_{i=1}^N \hat{\gamma}_i y_i}{\sum_{i=1}^N \hat{\gamma}_i}, \quad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^N \hat{\gamma}_i (y_i-\hat{\mu}_2)^2}{\sum_{i=1}^N \hat{\gamma}_i}$$

and the mixing probability

$$\hat{\pi} = \sum_{i=1}^N \hat{\gamma}_i / N$$

4. Iterate steps 2 and 3 until convergence

Initialization

- Construct initial guesses for $\hat{\mu}_1$ and $\hat{\mu}_2$: choose two of the y_i at random
- Both $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ set equal to the overall sample variance $\sum_{i=1}^N (y_i - \bar{y})^2 / N$
- Mixing proportion $\hat{\pi}$ can be started at the value 0.5

e.g. at one iteration, the contribution of y1 to model 1 is 0.3 and to model 2 is 0.7.
Sample y2 to model1's contribution is 0.2 and to model 2 is 0.8.
Then the mean of model 1 from these two samples is (y1*0.3+y2*0.2)/(0.3+0.2)

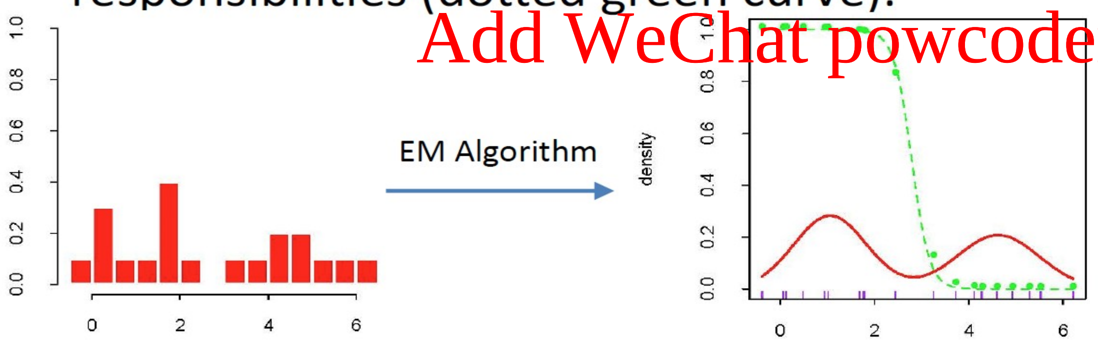
Example output

Example of Running EM

- The final maximum likelihood estimates:

$$\hat{\mu}_1 = 4.62, \qquad \hat{\sigma}_1^2 = 0.87$$
$$\hat{\mu}_2 = 1.06, \qquad \hat{\sigma}_2^2 = 0.77$$
$$\hat{\pi} = 0.546$$

- The estimated Gaussian mixture density from this procedure (solid red curve), along with the responsibilities (dotted green curve):

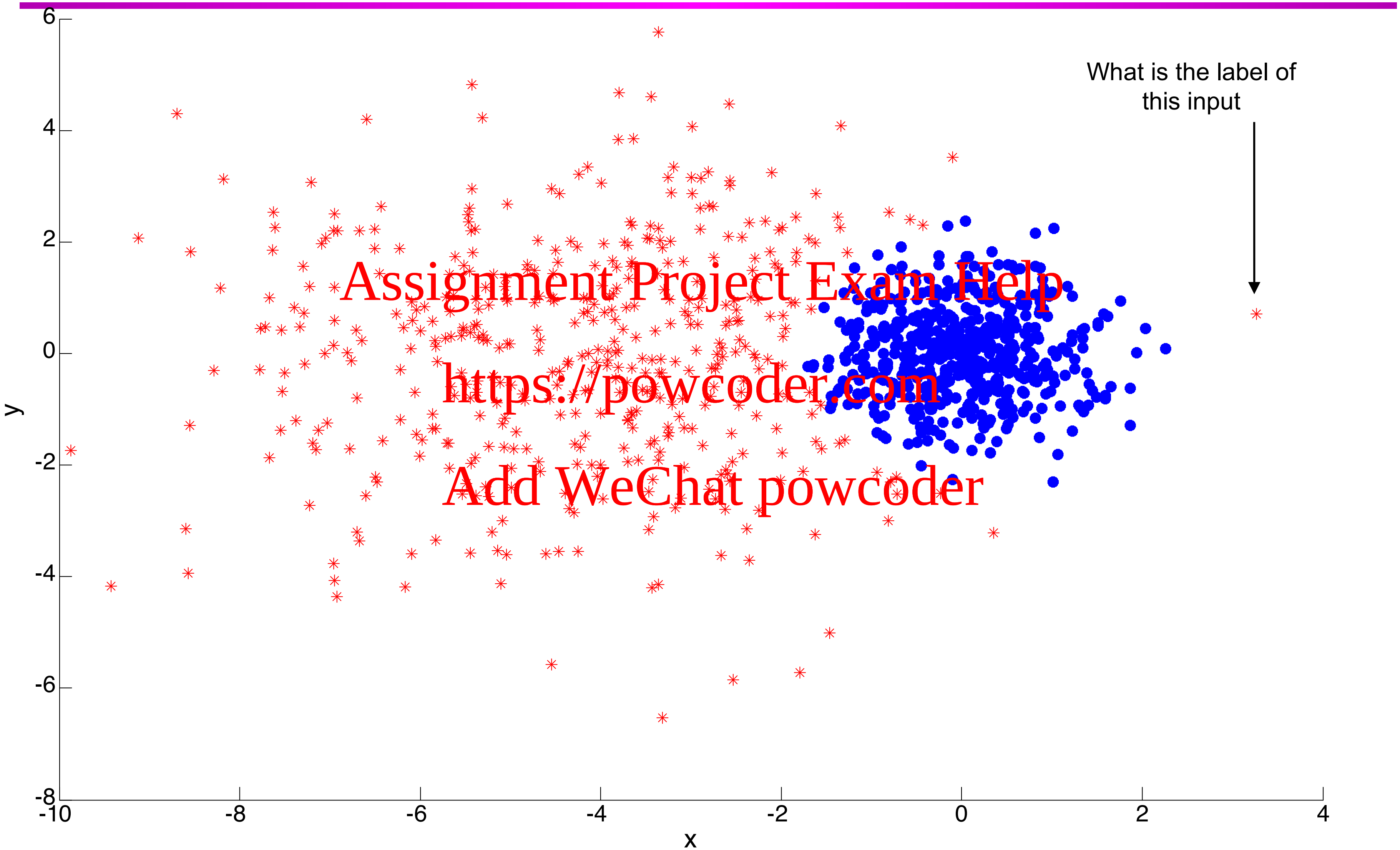


iterations

[illegible]	$\hat{\pi}$
1	0.485
5	0.493
10	0.523
15	0.544
20	0.546

Responsibility of each data point to two distributions

Probabilistic Clustering: Dense and Sparse Clusters



Problems with EM

- Convergence can be slow
- Only guarantees finding local maxima

- Makes some significant statistical assumptions