Gaussian Mixture Model

By:-

Dr. Jagannath H. Nirmal

Professor & Head

Department of Electronics Engineering

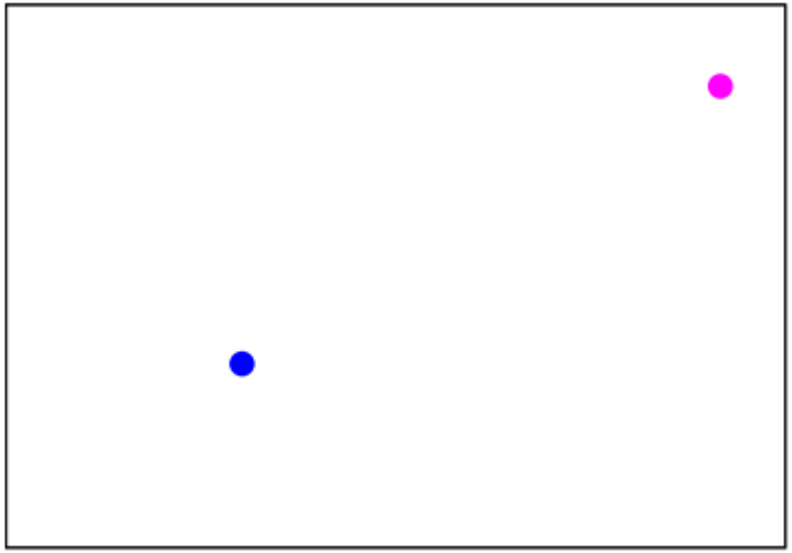
K.J. Somaiya College of Engineering, Mumbai

Gauss

Johann Carl Friedrich Gauss (1777-1855)
"Greatest Mathematician since Antiquity"

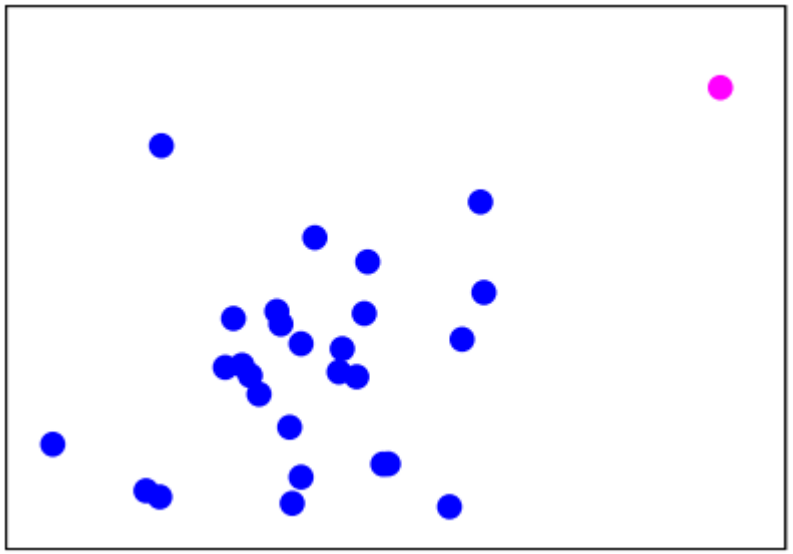
The Scenario

- Compute distance between test frame and frame of template



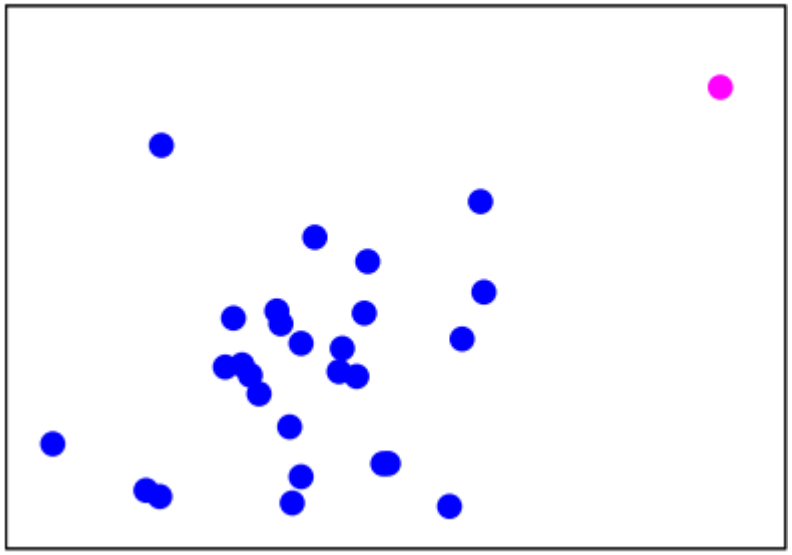
Problem Formulation

- What if instead of one training sample, have many?



Ideas

- Average training samples;
- compute Euclidean distance.
- Find best match over all training samples.
- Make probabilistic model of training samples.



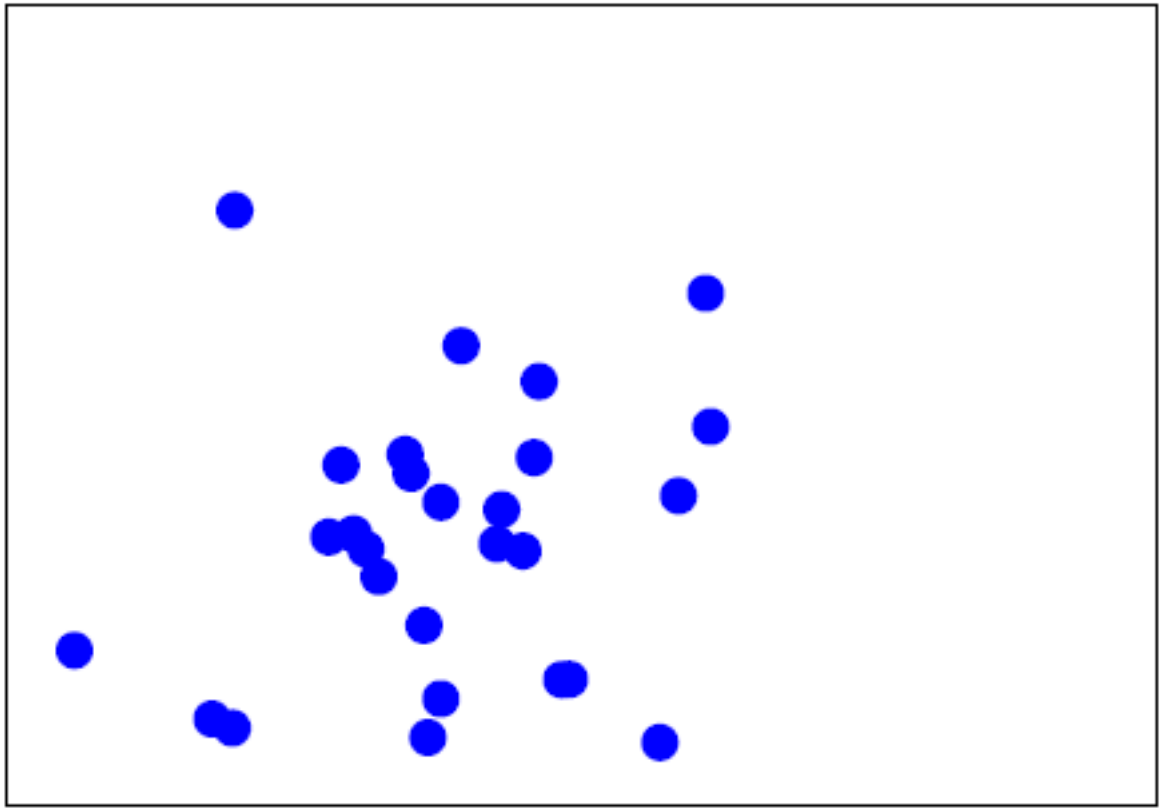
Where are we ?

- **Gaussian in one dimension**
- Gaussian in multiple dimension
- Estimating Gaussians from Data

Problem Formulation, Two Dimensions

- Estimate P(x1, x2), the "frequency" . . .

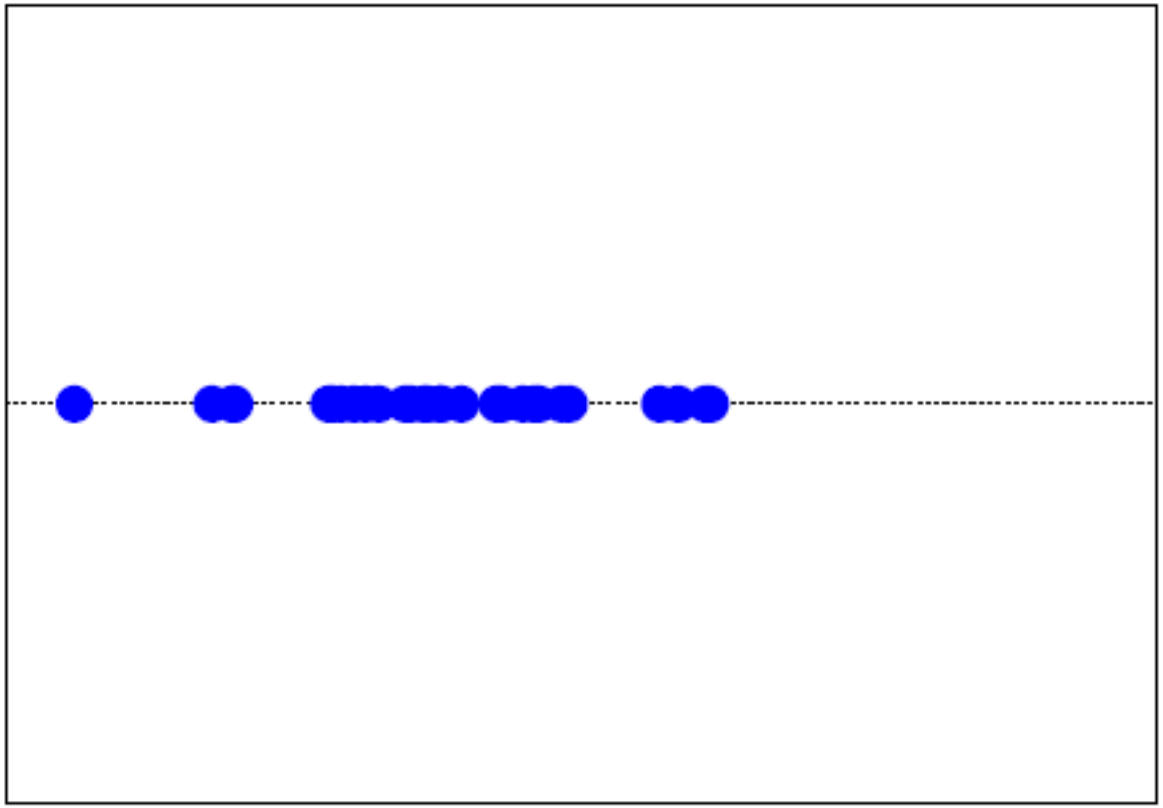
  That training sample occurs at location (x1, x2).



Let's Start With One Dimension

- Estimate P(x), the "frequency" . . .

  That training sample occurs at location x.



The Gaussian or Normal Distribution

$$P_{\mu,\sigma^2}(x) = \mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

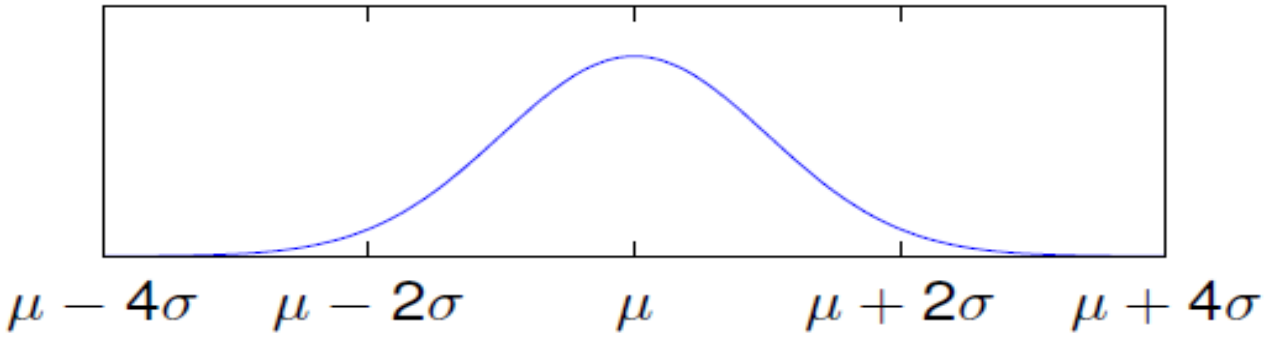
Parametric distribution with two parameters:

μ = mean (the center of the data).

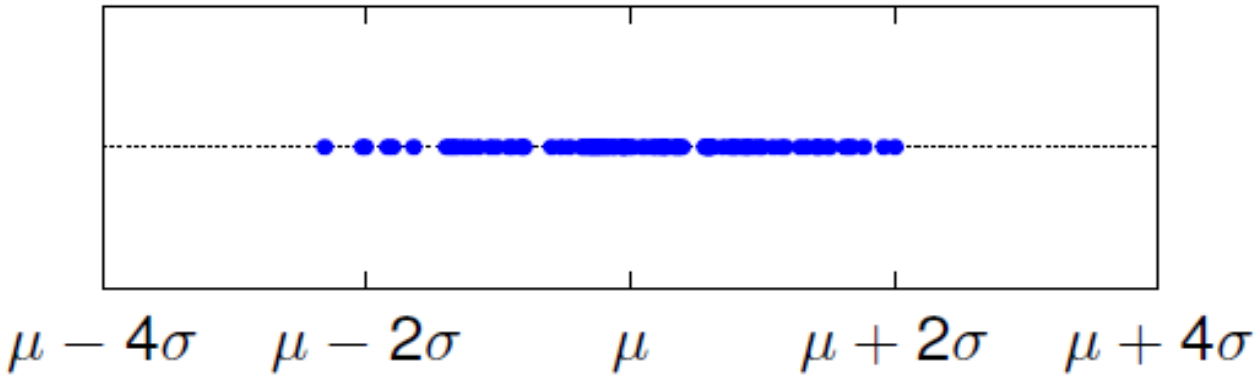
$\sigma 2$ = variance (how wide data is spread).

Visualization

- Density function:



- Sample from distribution:



Where are we ?

- Gaussian in one dimension
- Gaussian in multiple dimension
- Estimating Gaussians from Data

Gaussians in Two Dimensions

$$\mathcal{N}(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-r^2}} e^{-\frac{1}{2(1-r^2)}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} - \frac{2rx_1x_2}{\sigma_1\sigma_2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right)}$$

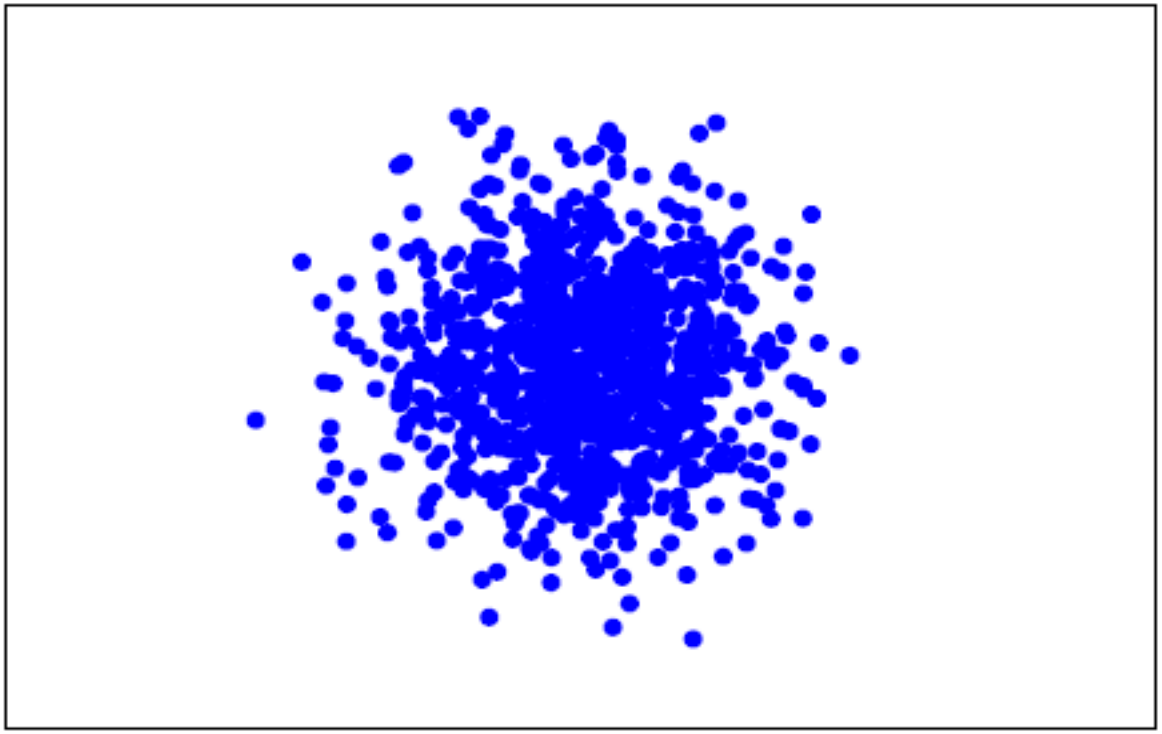
- If r = 0, simplifies to

$$\frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x_1-\mu_1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x_2-\mu_2)^2}{2\sigma_2^2}} = \mathcal{N}(\mu_1, \sigma_1^2)\mathcal{N}(\mu_2, \sigma_2^2)$$

i.e., like generating each dimension independently.

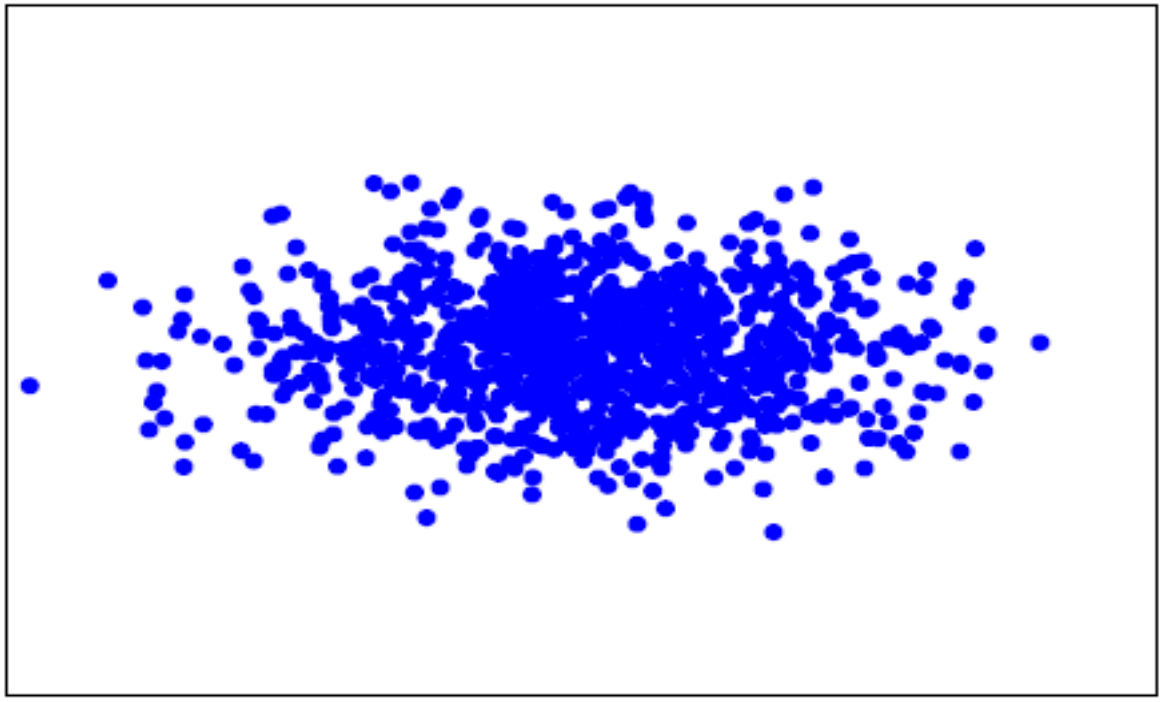
Example: $r = 0$, $\sigma_1 = \sigma_2$

- x1, x2 uncorrelated.
  - Knowing x1 tells you nothing about x2.



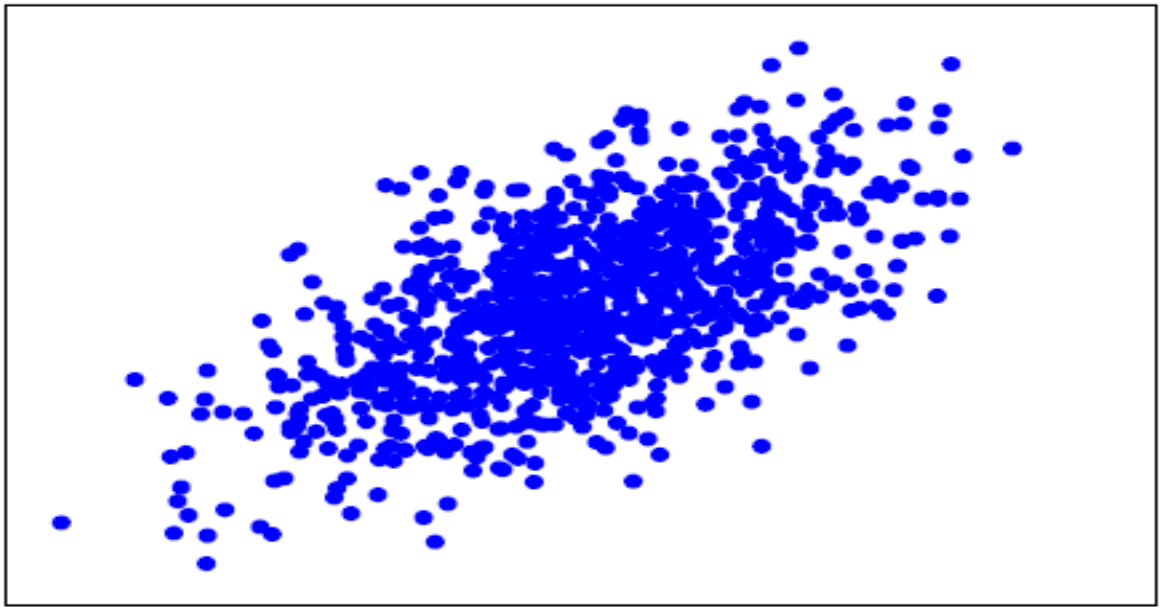
Example: r = 0, σ1≠ σ2

- x1, x2 can be uncorrelated and have unequal variance.



Example: r > 0, σ1 ≠ σ2

- x1, x2 correlated.
  - Knowing x1 tells you something about x2.



Generalizing to More Dimensions

- If we write following matrix:

$$\Sigma = \begin{vmatrix} \sigma_1^2 & r\sigma_1\sigma_2 \\ r\sigma_1\sigma_2 & \sigma_2^2 \end{vmatrix}$$

then another way to write two-dimensional Gaussian is:

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})}$$

where $\mathbf{x} = (x_1, x_2)$, $\boldsymbol{\mu} = (\mu_1, \mu_2)$.

- More generally, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can have arbitrary numbers of components.
  - *Multivariate* Gaussians.

Diagonal and Full Covariance Gaussians

- Let's say have 40d feature vector.
  - How many parameters in covariance matrix Σ?
  - The more parameters, . . .
  - The more data you need to estimate them.
- In ASR, usually assume Σ is diagonal $\Rightarrow$ d param.
  - This is why like having uncorrelated features!

Computing Gaussian Log Likelihoods

- Why *log* likelihoods?
- Full covariance:

$$\log P(\mathbf{x}) = -\frac{d}{2}\ln(2\pi) - \frac{1}{2}\ln|\Sigma| - \frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})$$

- Diagonal covariance:

$$\log P(\mathbf{x}) = -\frac{d}{2}\ln(2\pi) - \sum_{i=1}^{d}\ln\sigma_i - \frac{1}{2}\sum_{i=1}^{d}(x_i-\mu_i)^2/\sigma_i^2$$

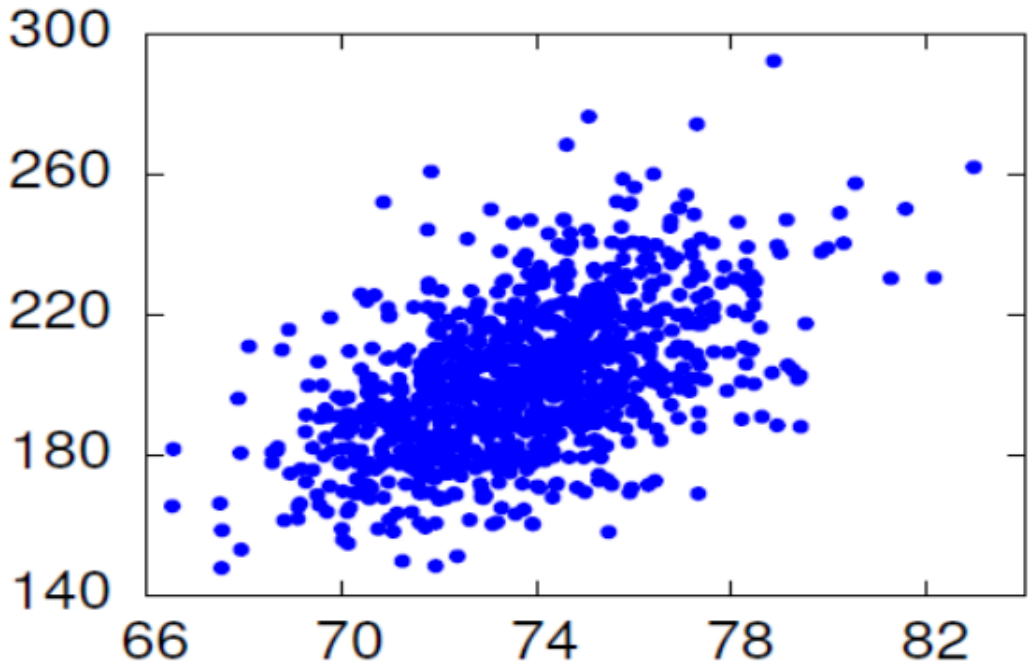
  - Again, note similarity to weighted Euclidean distance.
  - Terms on left independent of $\mathbf{x}$; precompute.
  - A few multiplies/adds per dimension.

Where are we ?

- Gaussian in one dimension
- Gaussian in multiple dimension
- Estimating Gaussians from Data

Estimating Gaussians

- Give training data, how to choose parameters μ, Σ?
- Find parameters so that resulting distribution . . .
  - "Matches" data as well as possible.
  - Sample data: height, weight of baseball players



Why Maximum-Likelihood Estimation ?

- Assume we have "correct" model form.
- Then, as the number of training samples increases . . .
  - ML estimates approach "true" parameter values (consistent)
  - ML estimators are the best! (efficient)
- ML estimation is easy for many types of models.
  - Count and normalize!

What is ML Estimate for Gaussians ?

- Much easier to work with log likelihood $L = \ln \mathcal{L}$:

$$L(x_1^N|\mu,\sigma) = -\frac{N}{2}\ln 2\pi\sigma^2 - \frac{1}{2}\sum_{i=1}^{N}\frac{(x_i-\mu)^2}{\sigma^2}$$

- Take partial derivatives w.r.t. μ, σ:

$$\frac{\partial L(x_1^N|\mu,\sigma)}{\partial \mu} = \sum_{i=1}^{N}\frac{(x_i-\mu)}{\sigma^2}$$

$$\frac{\partial L(x_1^N|\mu,\sigma)}{\partial \sigma^2} = -\frac{N}{2\sigma^2} + \sum_{i=1}^{N}\frac{(x_i-\mu)^2}{\sigma^4}$$

- Set equal to zero; solve for μ, σ^2.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2$$

What is ML Estimate for Gaussians?

- Multivariate case.

$$\mu = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$$

$$\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i - \mu)^T(\mathbf{x}_i - \mu)$$

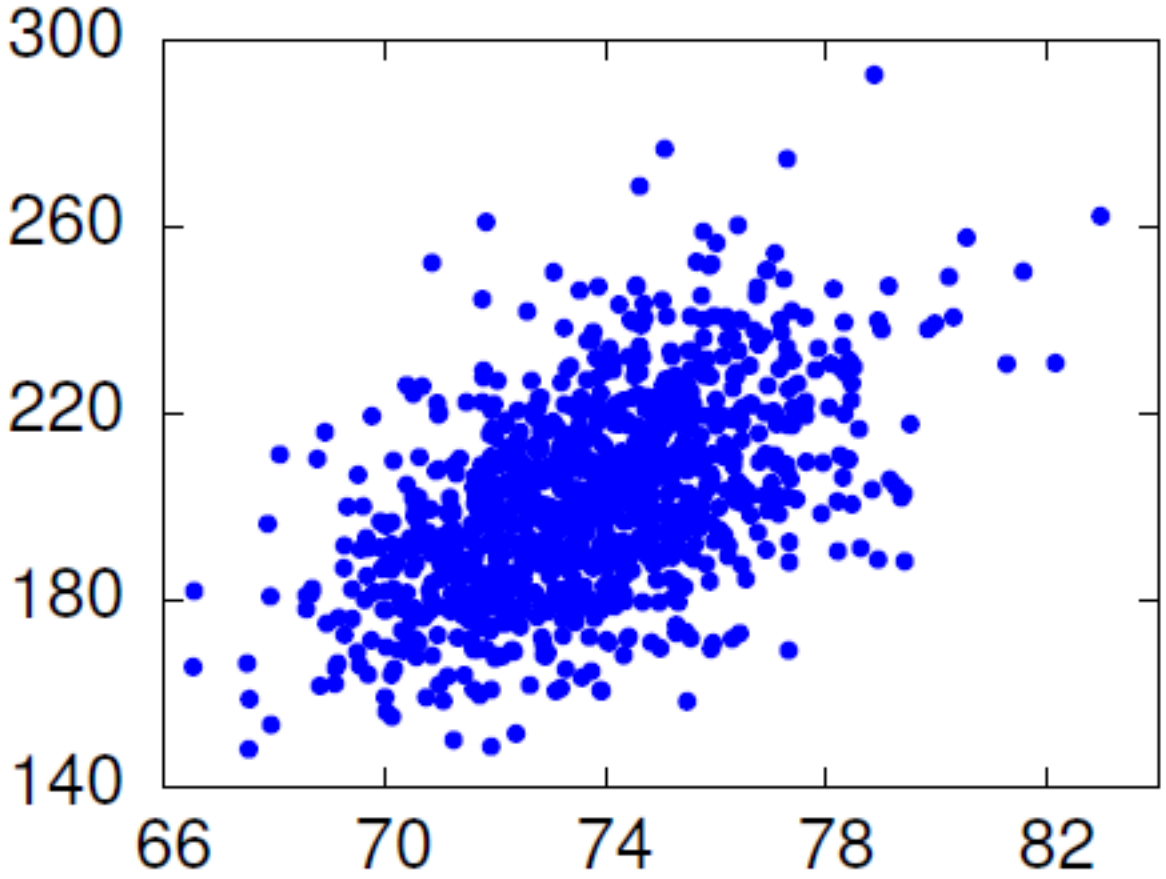
- What if diagonal covariance?
  - Estimate params for each dimension independently.

Example: ML Estimation

- Heights and weights of 1033 pro baseball players. Noise added to hide discretization effects.

height	weight
74.34	181.29
73.92	213.79
72.01	209.52
72.28	209.02
72.98	188.42
69.41	176.02
68.78	210.28
...	...
...	...

Example: ML Estimation



Example: Diagonal Covariance

$$\mu_1 = \frac{1}{1033}(74.34 + 73.92 + 72.01 + \cdots) = 73.71$$

$$\mu_2 = \frac{1}{1033}(181.29 + 213.79 + 209.52 + \cdots) = 201.69$$

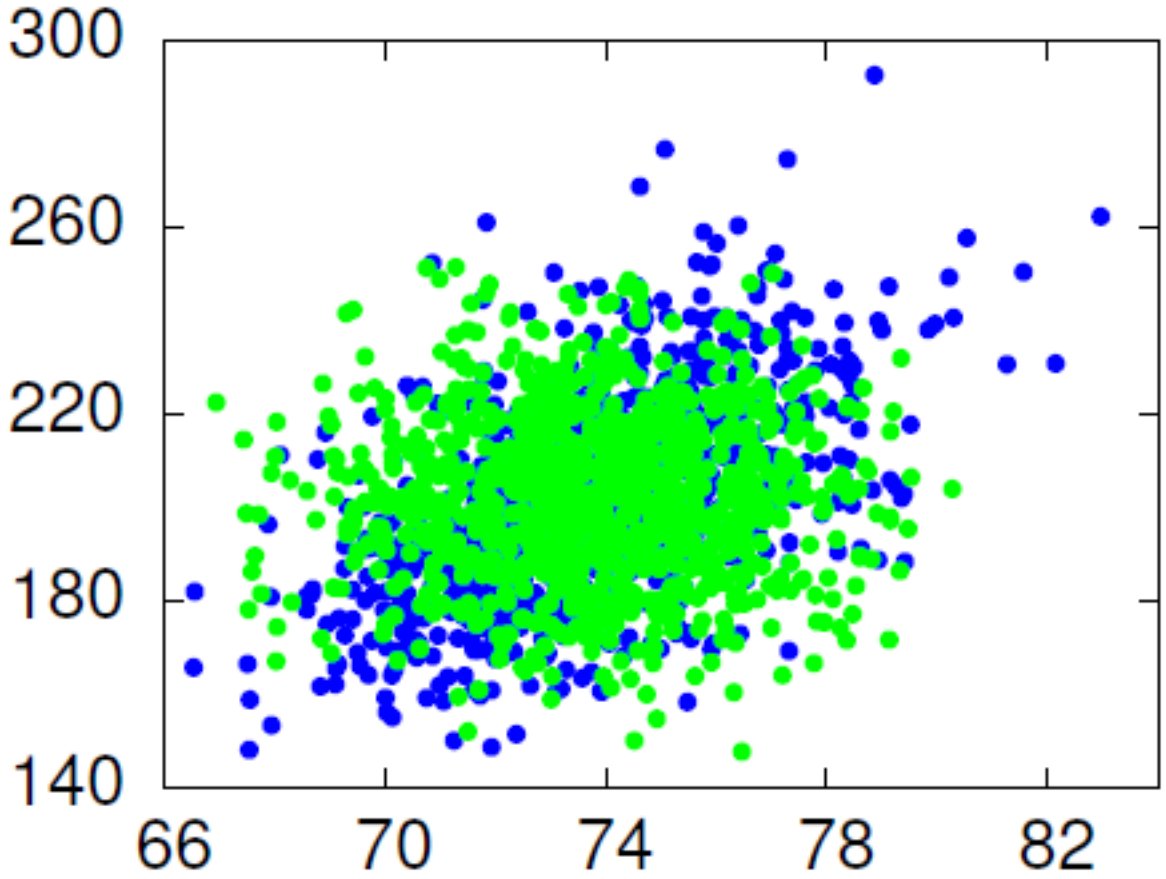
$$\sigma_1^2 = \frac{1}{1033}\left[(74.34 - 73.71)^2 + (73.92 - 73.71)^2 + \cdots\right]$$

$$= 5.43$$

$$\sigma_2^2 = \frac{1}{1033}\left[(181.29 - 201.69)^2 + (213.79 - 201.69)^2 + \cdots\right]$$

$$= 440.62$$

Example: Diagonal Covariance



Example: Full Covariance

- Mean; diagonal elements of covariance matrix the same.

$$\Sigma_{12} = \Sigma_{21}$$

$$= \frac{1}{1033}[(74.34 - 73.71) \times (181.29 - 201.69) +$$

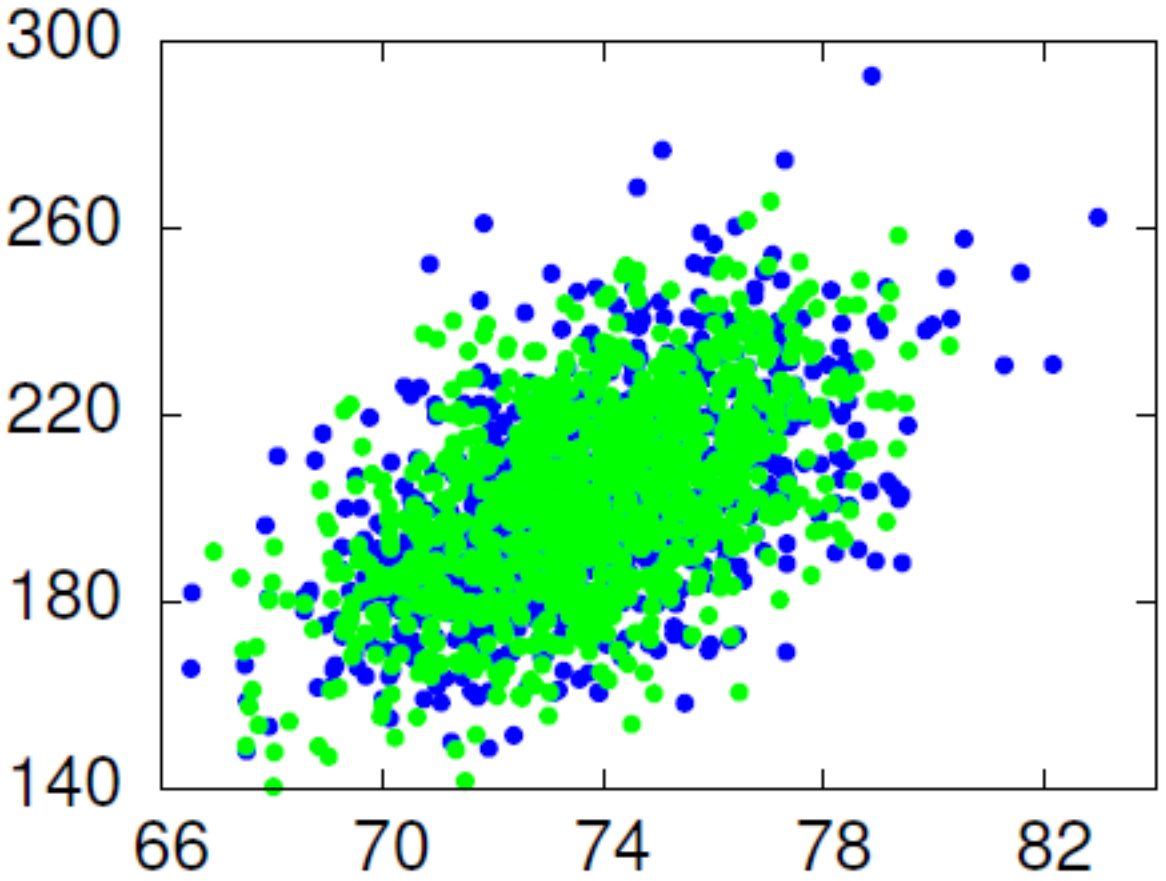
$$(73.92 - 73.71) \times (213.79 - 201.69) + \cdots)]$$

$$= 25.43$$

$$\mu = [\ 73.71 \quad 201.69\ ]$$

$$\Sigma = \begin{vmatrix} 5.43 & 25.43 \\ 25.43 & 440.62 \end{vmatrix}$$

Example: Full Covariance



Part II

Gaussian Mixture Models

Gaussian Distribution

- Gaussian or Normal Distribution, 1D

$$N(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{\frac{1}{2}(x - \mu)^2}{\sigma^2}\right]$$

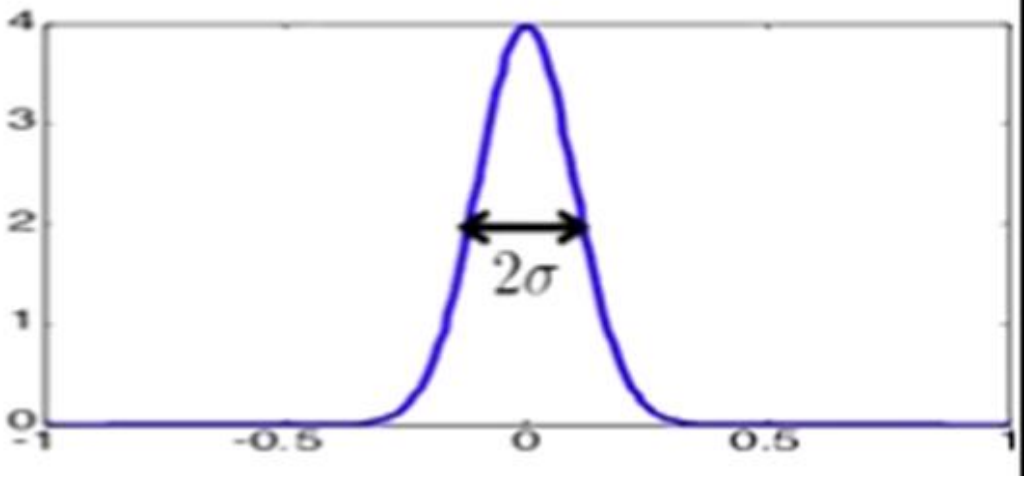
Parameters: mean μ, variance σ^2

(standard deviation)

Maximum Likelihood estimates

$$\mu = \frac{1}{N}\sum x^{(i)}$$

$$\sigma^2 = \frac{1}{N}\sum \left(x^{(i)} - \mu\right)^2$$



Multivariate Gaussian

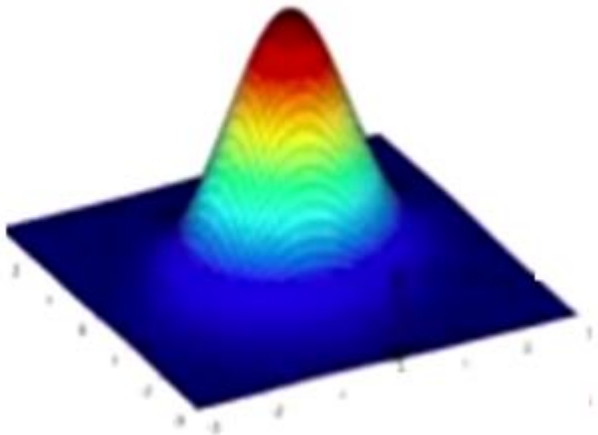
- Similar to Univariate Case

$$N(x: \mu, \sum) = \frac{1}{(2\pi)^{\frac{d}{2}}} |\sum|^{-\frac{1}{2}} exp\left\{-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)^T\right\}$$

μ = length – d row vector

$\sum$ d x d matrix

$|\sum|$ = matrix determinant



Multivariate Gaussians

- Defining μ and ∑

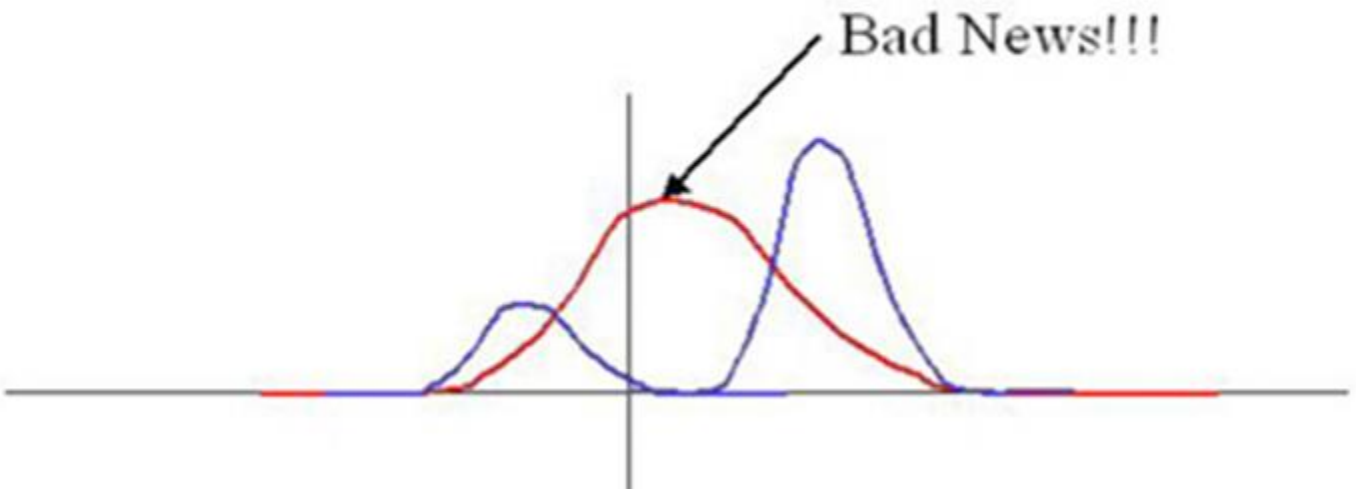
$$\mu = E(x)$$

$$\sum = E\left[(x - \mu)(x - \mu)^T\right]$$

- So the i-jth element of ∑ is:

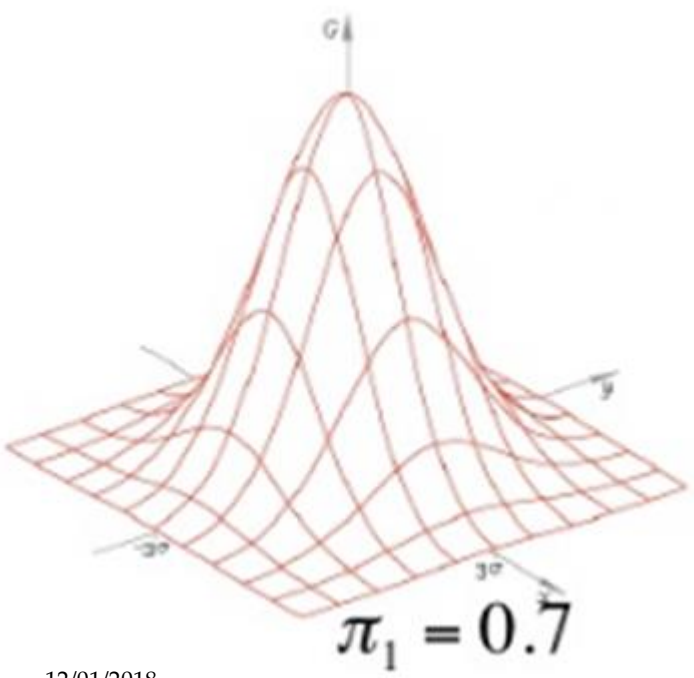
$$\sigma_{ij}^2 = E\left[(x_i - \mu_i)(x_i - \mu_j)\right]$$

- Single Gaussian may do a bad job of modeling distribution in any dimension



What's Gaussian Mixture?

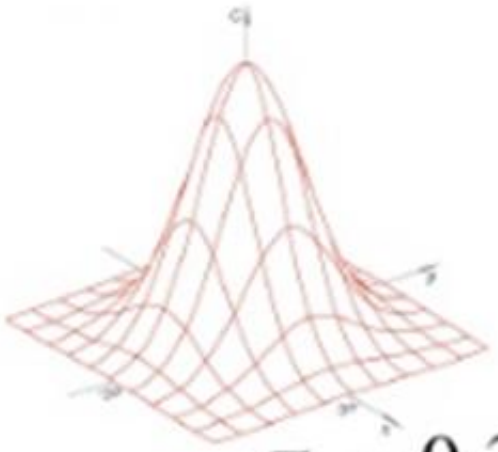
$$p(X) = \pi_1 N(X \mid \mu_1, \Sigma_1) + \pi_2 N(X \mid \mu_2, \Sigma_2)$$



$\pi_1 = 0.7$

Mixing Coefficient

$$\sum_{k=1}^{k} \pi_k = 1$$



$\pi_2 = 0.3$

Gaussian Mixture Models

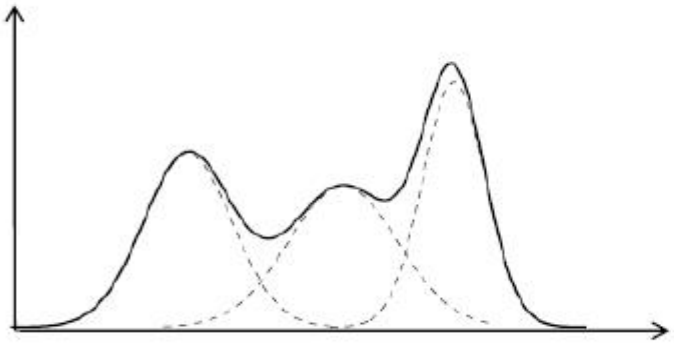
- GMM: the weighted sum of a number of Gaussian where the weights are determined by distribution, π

$$p(X) = \pi_0 N(x\,|\mu_0, \sum_0) + \pi_1 N(x|\mu_1, \sum_1) + \ldots + \pi_k N(x\,|\mu_K, \sum_K)$$

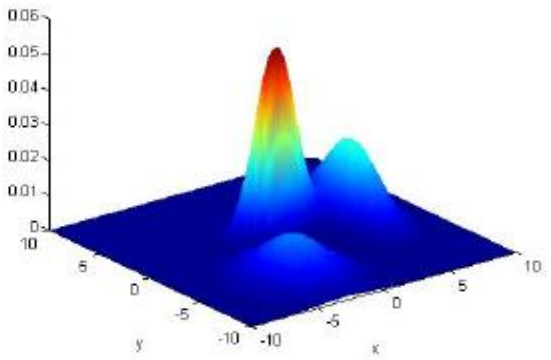
Where $\sum_{i=0}^{k} \pi_i = 1$ $\quad p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \sum_k)$

Examples:

d=1:



d=2:



What is a Gaussian Mixture Model

- **Problem**

  Given a set of data $X = \{x_1, x_2, \ldots\ldots\ldots\ldots x_N\}$ drawn from an unknown distribution (probably a GMM) estimates the parameters θ of the GMM model that fits the data

- **Solution**

  Maximize the likelihood $p(X\,|\theta)$ of the data with regard to the model parameters?

$$\theta^* = arg\max_{\theta} p(X\,|\theta) = \arg\max_{\theta} \prod_{i=1}^{N} p(x_i|\theta)$$

Maximum Likelihood Estimation

- **Consider log of Gaussian Distribution**

$$\ln p(x \mid \mu, \sum) = -\frac{1}{2}\, in(2\pi) - \frac{1}{2}\, ln|\sum| - \frac{1}{2}(x - \mu)^T \sum^{-1} (x - \mu)$$

- **Take the derivative and equate it to zero**

$$\frac{\partial \ln p(x \mid \mu, \sum)}{\partial \mu} = 0 \qquad \frac{\partial \ln p(x \mid \mu, \sum)}{\partial \sum} = 0$$

$$\mu_{ML} = \frac{1}{N} \sum_{a=1}^{N} X_n \qquad \sum_{ML} = \frac{1}{N} \sum_{a=1}^{N} (x_n - \mu_{ML})\,(x_n - \mu_{ML})^T$$

Maximum Likelihood Estimation

$$p(x) = \sum_{k=1}^{k} \pi_k N(x \mid \mu_k, \sum_K)$$

$$N(x|\sum_k) = \frac{1}{(2\pi)^{\frac{d}{2}} \|\sum\|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x - \mu)^T \sum^{-1} (x - \mu)\right\}$$

Which is called a Mixture of Gaussian or Gaussian Mixture Model

Each Gaussian density is called component of the mixtures and has its own mean μ_k and covariance $\sum_k$.

The parameters are called mixing coefficients ($\sum_k \pi_k = k$)

- The log likelihood function is given by

$$\ln p(X|\pi, \mu, \sum) = \sum_{n=1}^{N} \ln \left\{ \sum_{K=1}^{K} \pi_k N(X_n|\mu_k, \sum_k) \right\}$$

- Goal : Find parameter which maximize log likelihood
- Problem: Hard to compute maximum likelihood
- Solution: Use EM algorithm

The Expectation-Maximization algorithm

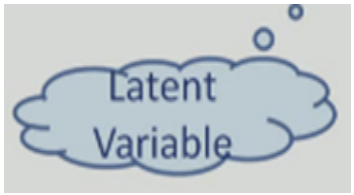
One of the most popular approaches to maximize the likelihood is to use the Expectation-Maximization (EM) algorithm

- Basic Ideas of EM algorithm :
  - Introduce the hidden variable such that its knowledge would simply the maximization of the likelihood
  - At each iteration :
    - E-Step Estimate the distribution of the hidden variable given the data and the current value of parameters.
    - M-Step Maximize the joint distribution of the data and the hidden variable

Latent Variable: posterior prob.

- We can think of the mixing coefficients as prior probabilities for the components
- For a given values of 'X', we can evaluate the corresponding posterior probabilities called responsibilities.
- From Baye's Rule

$$\gamma_k(x) = p(k|x) = \frac{p(k)p(x|k)}{p(x)}$$



$$= \frac{\pi_k N(x|\mu_k, \sum_k)}{\sum_j^k \pi_j N(x|\mu_j, \sum_j)} \qquad where\ \pi_k = \frac{N_k}{N}$$

Interpret N_k as the effective number of points assigned to cluster K

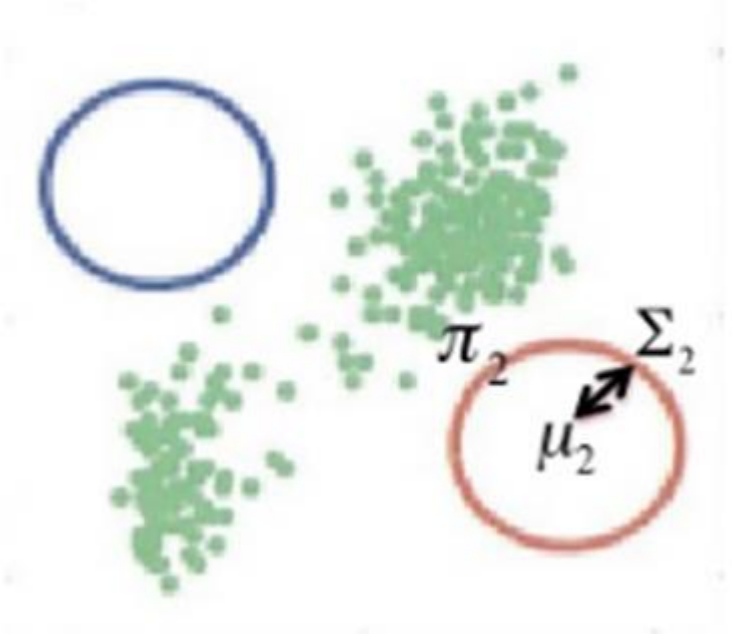
EM for Gaussian Mixtures

- Initialize μ_k, $\sum_k$, π_k one for each Gaussian K
- Tip! Use K-means result to initialize :

$$\mu_k \longleftarrow \quad \mu_k$$

$$\sum_k \longleftarrow \quad \text{conv(cluster (k))}$$

$$\Pi_k \longleftarrow \frac{Number\ of\ points\ in\ k}{Total\ number\ of\ points}$$

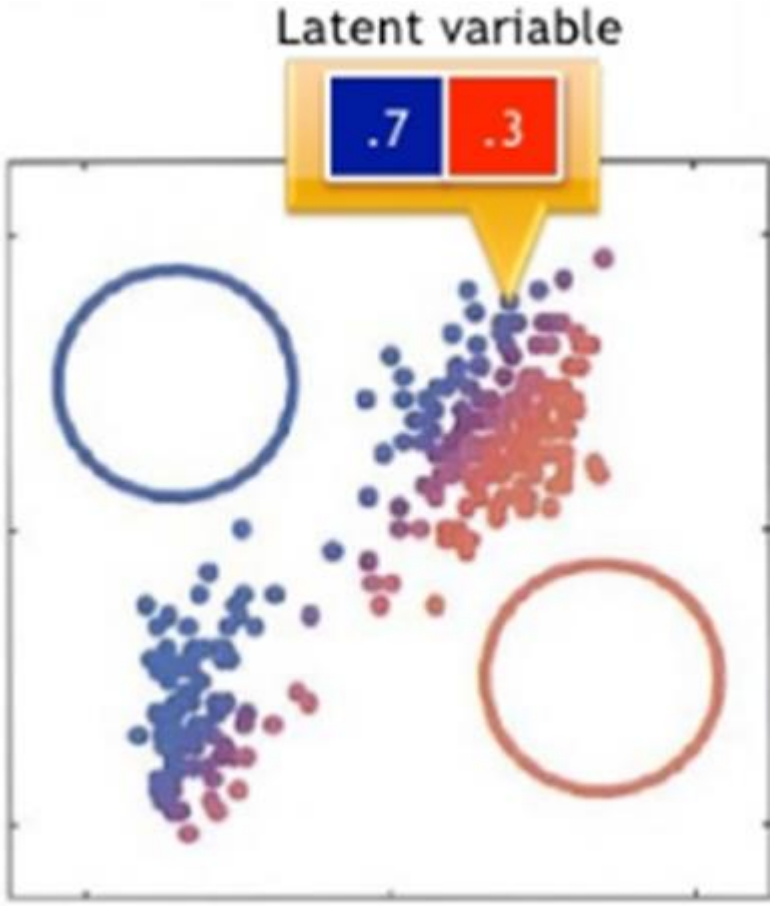


EM for Gaussian Mixtures

- **E Step**: For each point X_n, determine its assignment score to each Gaussian k:

$$\gamma(Z_{nk}) = \frac{\pi_k N(X_n|\mu_k, \sum_k)}{\sum_{j=1}^{k} \pi_j N(X_n|\mu_j, \sum_j)}$$

$\gamma(Z_{nk})$ is called a "responsibility": how much is its Gaussian k responsible for this point X_n ?



EM for Gaussian Mixtures

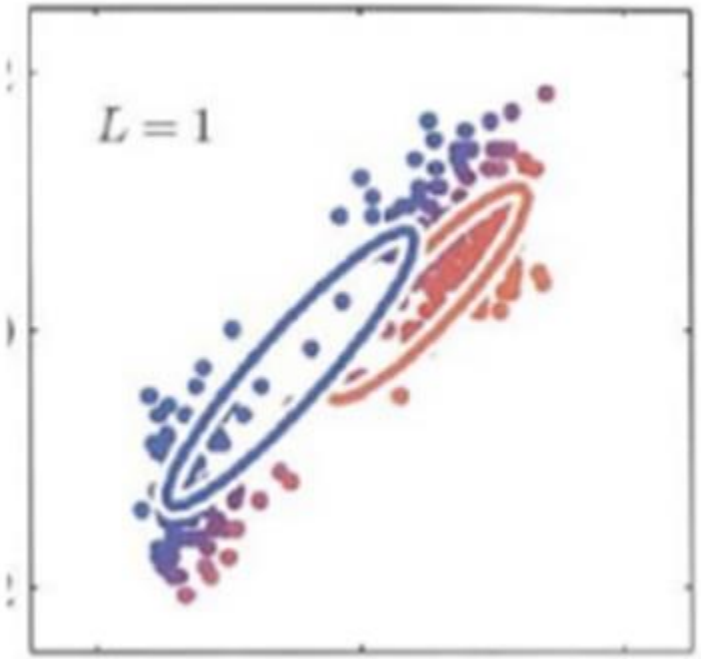
- **M step** : For each Gaussian k, update parameters using new $\gamma(Z_{nk})$

- **Mean of Gaussian k**

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n-1}^{N} \gamma(Z_{nk})\, X_n$$

$$N_k = \sum_{n=1}^{N} \gamma(Z_{nk})$$

Find mean that 'Fits' the assignment score best

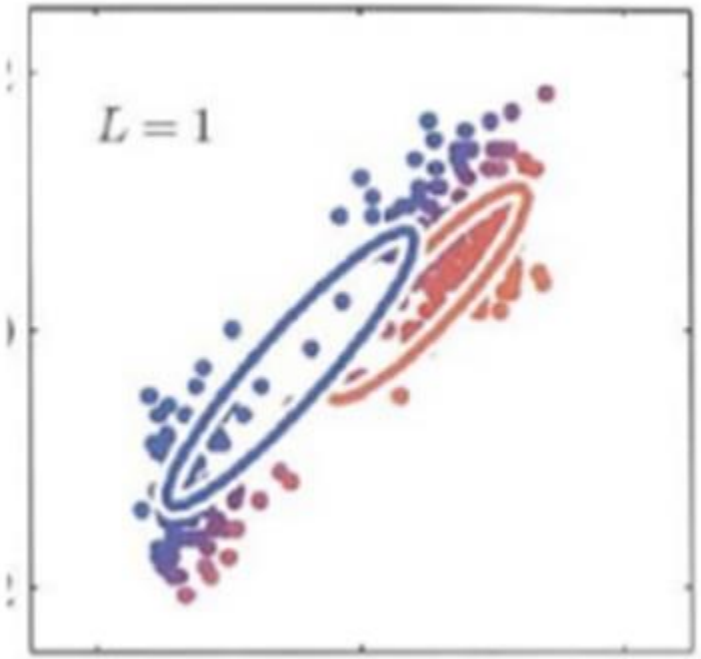


EM for Gaussian Mixture

- M step: For each Gaussian k, update parameters using new $\gamma(Z_{nk})$
- Covariance matrix of Gaussian k

$$\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk}) * (X_n - \mu_k^{new})(X_n - \mu_k^{new})^T$$

Just calculated this!



EM for Gaussian Mixture

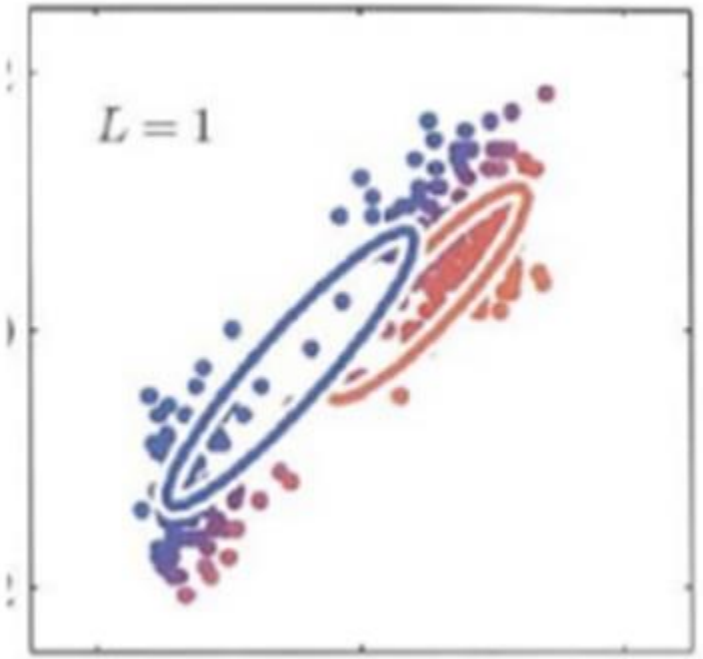
- **M step:** For each Gaussian k, update parameters using new $\gamma(Z_{nk})$

- Mixing Coefficients for Gaussian k

$$N_k = \sum_{n=1}^{N} \gamma(Z_{nk})$$

$$\pi_k^{new} = \frac{N_k}{N}$$

Total # of points



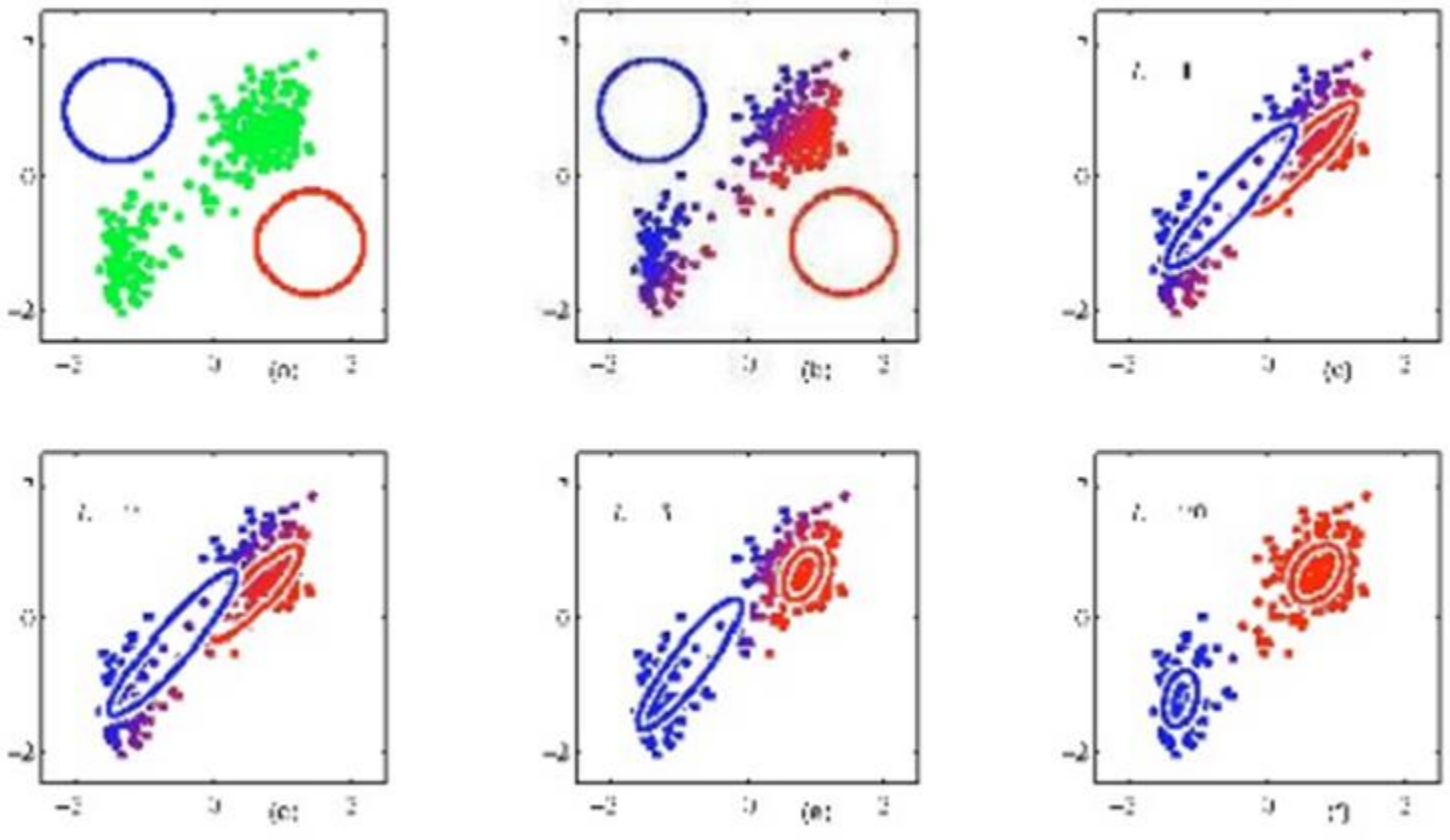
EM for Gaussian Mixtures

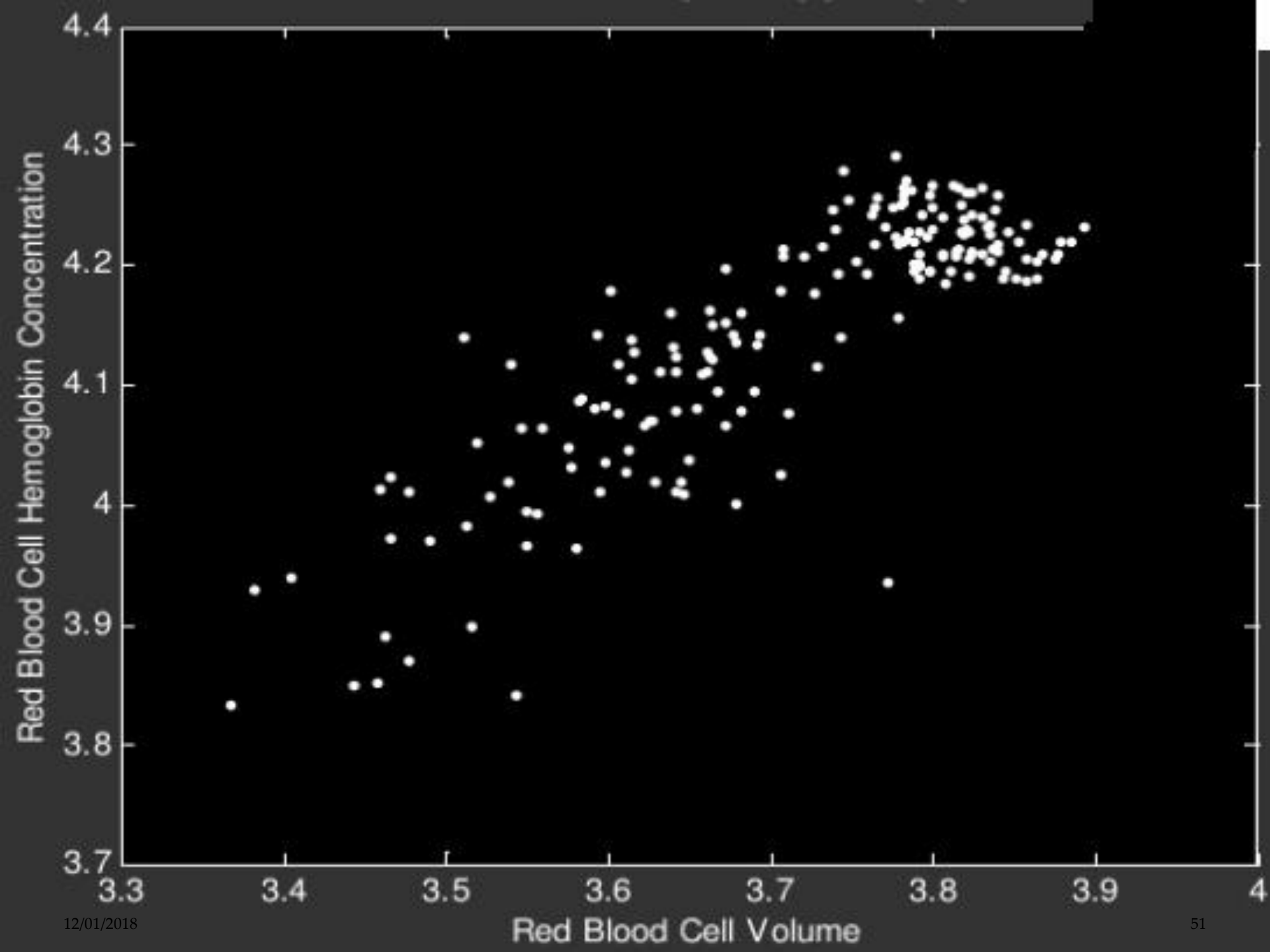
- Evaluate log likelihood. If likelihood or parameters coverage, stop. Else go to Step 2 (E step).

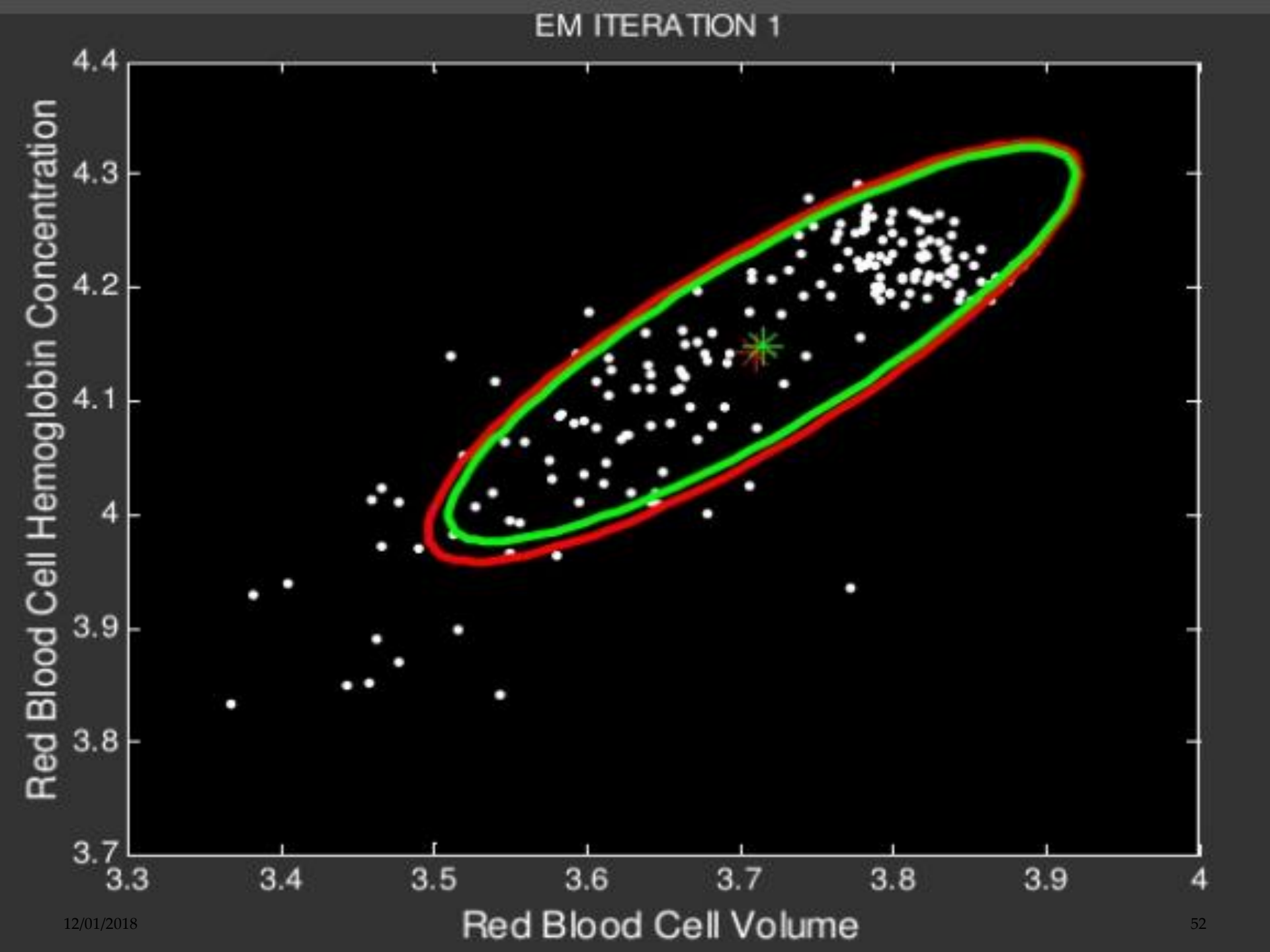
$$\ln p(X\,|\mu, \sum, \pi) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(X_n|\mu_k, \sum_k) \right\}$$

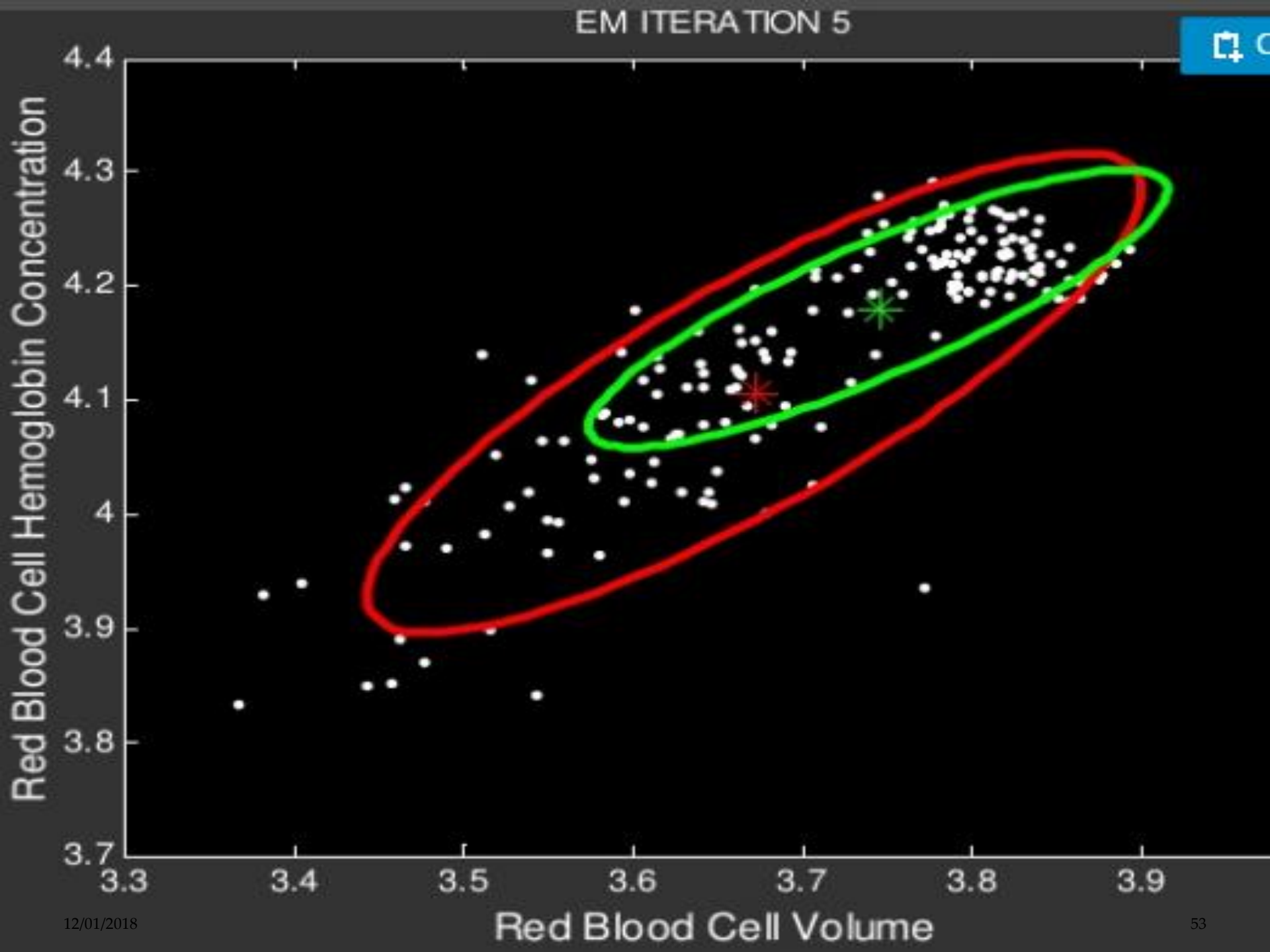
- Likelihood is the probability that the data X was generated by the parameters you found i.e. Correctness!

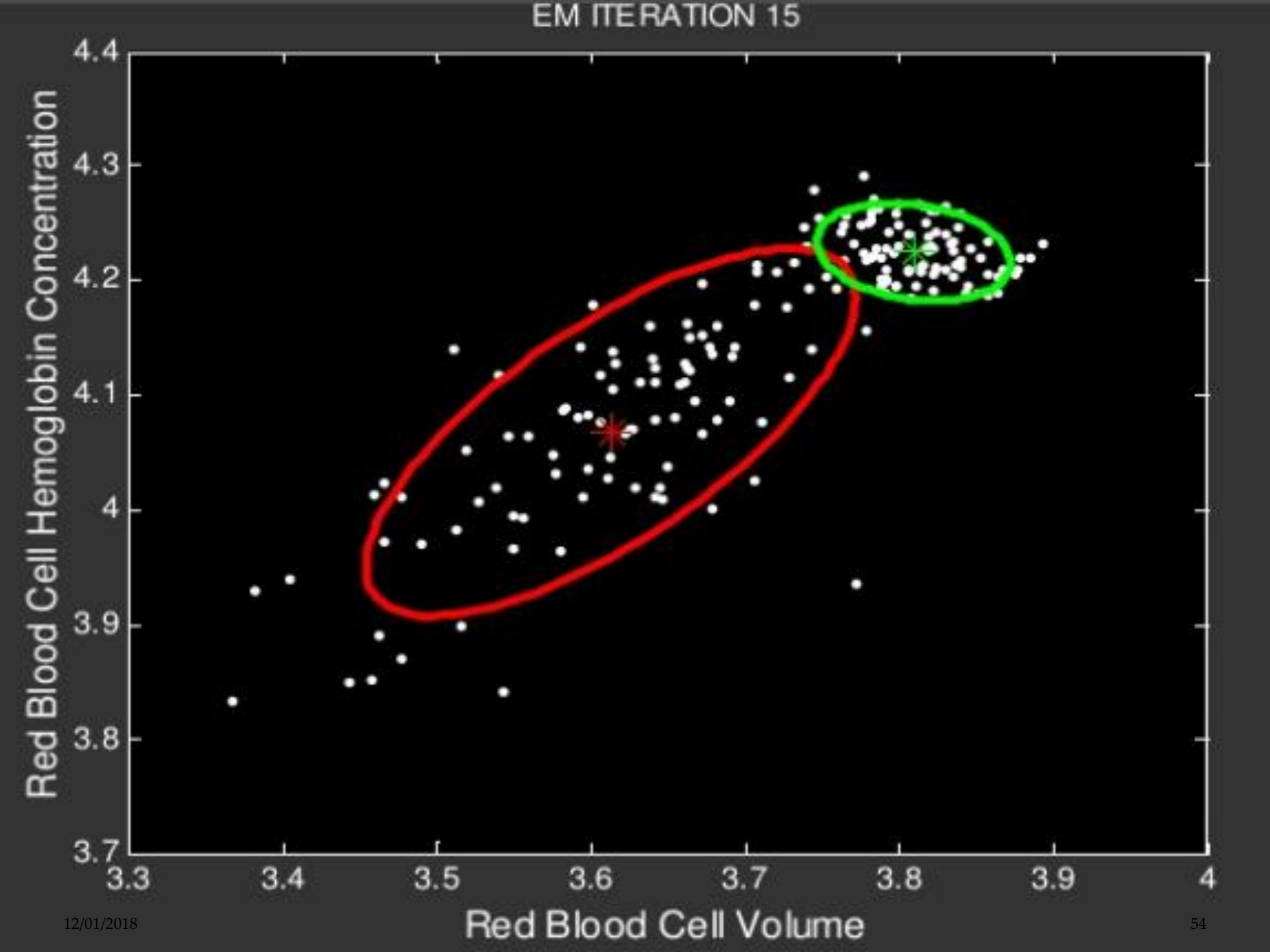
Visual Example of EM

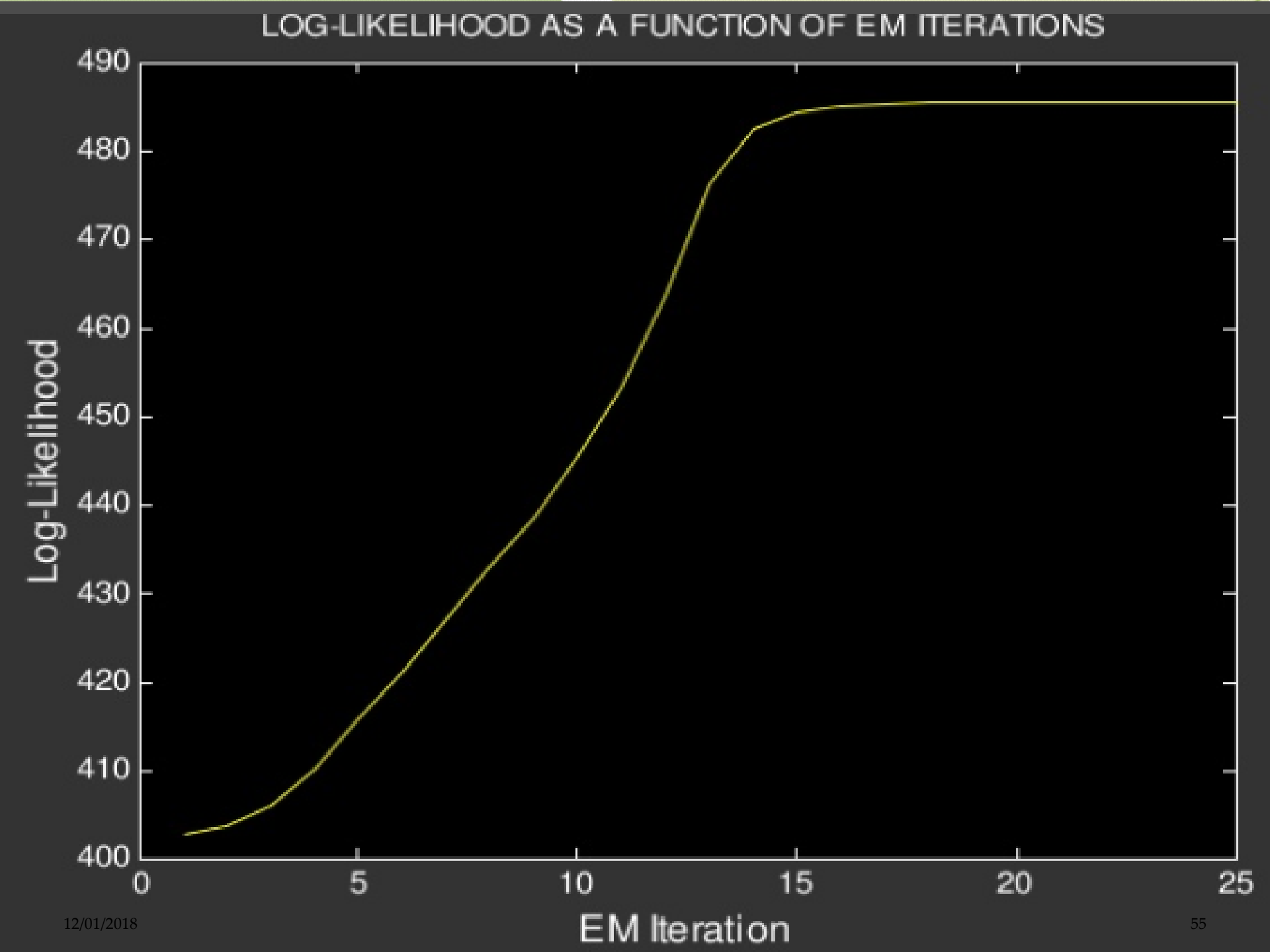



Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4
Red Blood Cell Volume


EM ITERATION 1
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


EM ITERATION 5
Red Blood Cell Hemoglobin Concentration
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
Red Blood Cell Volume


EM ITERATION 15
Red Blood Cell Hemoglobin Concentration
Red Blood Cell Volume
4.4
4.3
4.2
4.1
4
3.9
3.8
3.7
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4


LOG-LIKELIHOOD AS A FUNCTION OF EM ITERATIONS
Log-Likelihood
EM Iteration
490
480
470
460
450
440
430
420
410
400
0
5
10
15
20
25

Choice of Covariance Matrix

- Nodal Covariance

  One covariance matrix per Gaussian Component

- Grand Covariance

  one covariance matrix for all Gaussian component

- Global Covariance

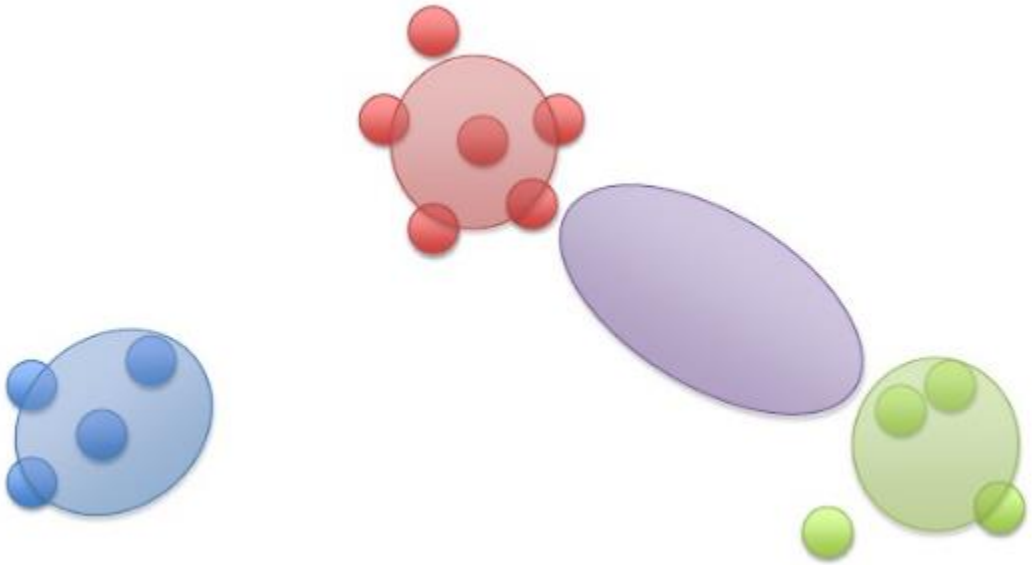
  single covariance matrix shared by all speaker component

Potential Problems

- Incorrect number of Mixtures Components
- Singularities
  - EM is an iterative algorithm which is very sensitive to initial conditions

  .
  - Usually, we use k means to get good initialization

Incorrect Number of Gaussians



Singularities

- When a mixture component collapses on a given point, the mean becomes the point and the variance goes to zero.

- Consider the likelihood function as the covariance goes to zero

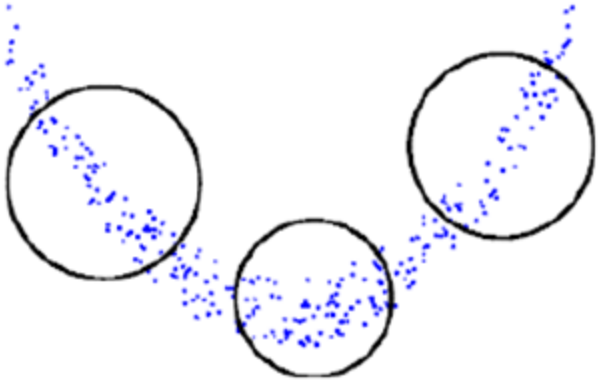
$$N(X_n|X_n, \sigma^2 I) = \frac{1}{\sqrt{2\pi}}\frac{1}{\sigma_j}$$

- The likelihood approaches infinity

$$p(x) = \sum_{i=0}^{k} \pi_i N(x\,|\mu_k, \sum_k)$$

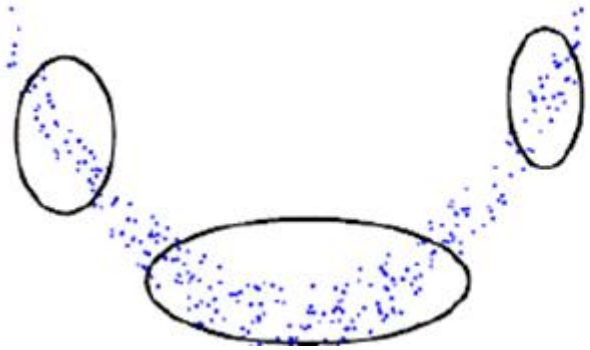
Simplification of the covariance matrix

- Case 1: Spherical Covariance matrix $\sum_j = diag(\sigma_i^2, \sigma_i^2, \ldots \ldots \sigma_i^2) = \sigma_j^2 I$



-Less precise.
-Very efficient to compute

- Diagonal covariance matrix $\sum_j = diag(\sigma_j^2, \sigma_j^2, \ldots \ldots \sigma_j^2)$



-More precise.
-Efficient to compute.

Simplification of the covariance matrices :

- Case full Covariance matrix

$$\Sigma_j = \begin{bmatrix} \sigma_{j1}^2 & \mathrm{cov}_j(x^1, x^2) & \cdots & \mathrm{cov}_j(x^1, x^d) \\ \mathrm{cov}_j(x^2, x^1) & \sigma_{j2}^2 & \cdots & \mathrm{cov}_j(x^2, x^d) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{cov}_j(x^d, x^1) & \mathrm{cov}_j(x^d, x^2) & \cdots & \sigma_{jd}^2 \end{bmatrix}$$

-Very precise.
-Less efficient to compute.

K-Means :- Gaussian Mixture

- K means is a classifier
- Mixture of Gaussian is a probability model
- We can use it as a 'Soft' classifier

Parameter to fit to data
Mean μ_k

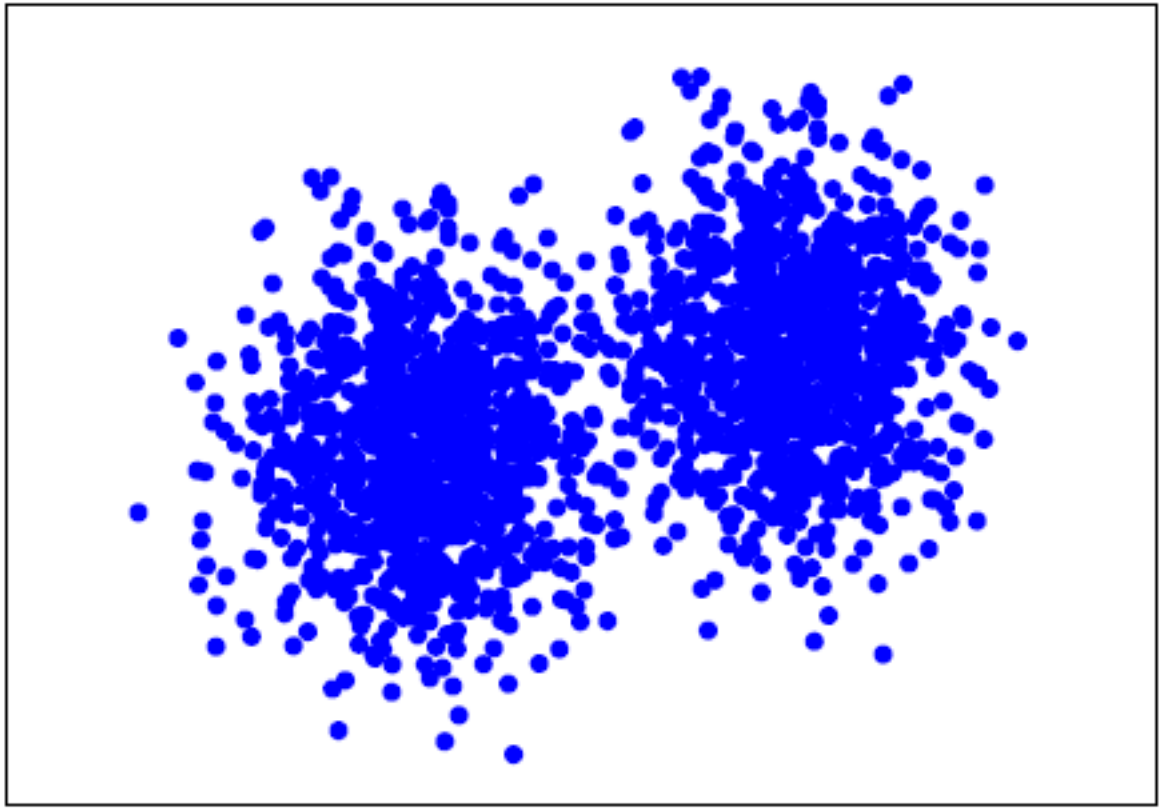
Parameters to fit to data
- Mean μ_k
- Covariance $\sum_k$
- Mixing Coefficients π_k

K Means Algorithm Reminder

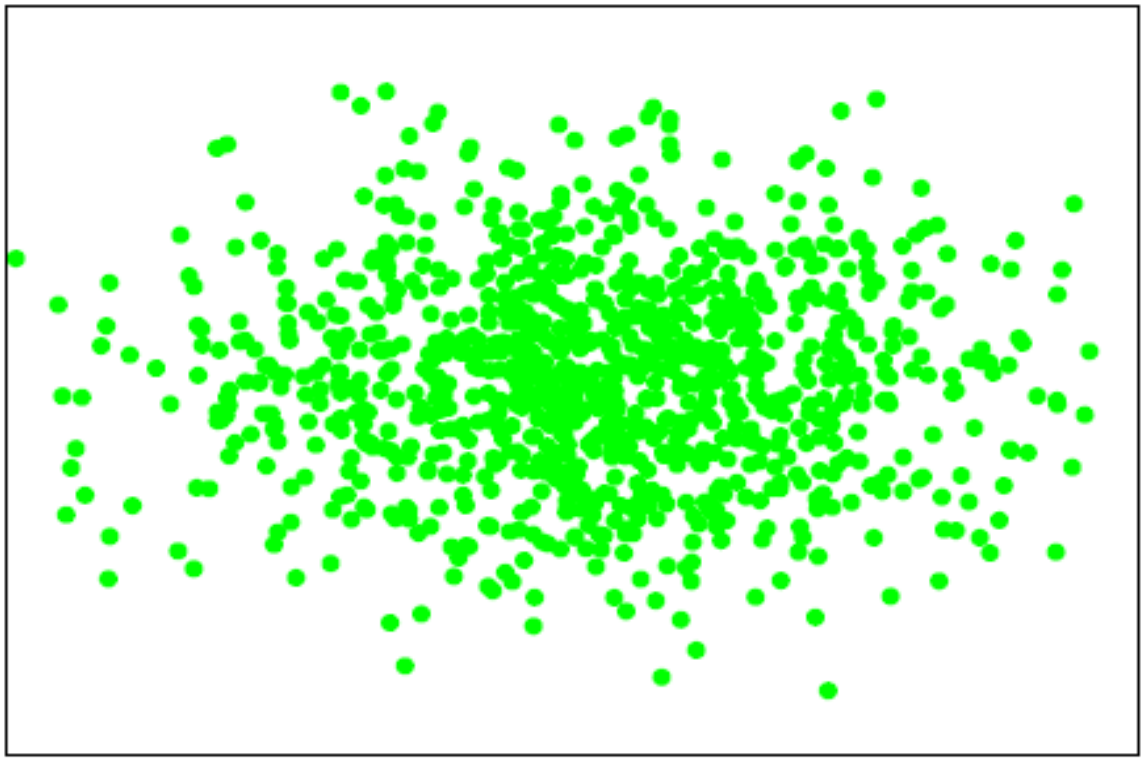
- Initialize means μ_k
  - E step: Assign each point to cluster
  - M step: Given clusters, refine mean $\mu_{k\ of}$ each cluster k

- Stop when change in means is small

Problems with Gaussian Assumption

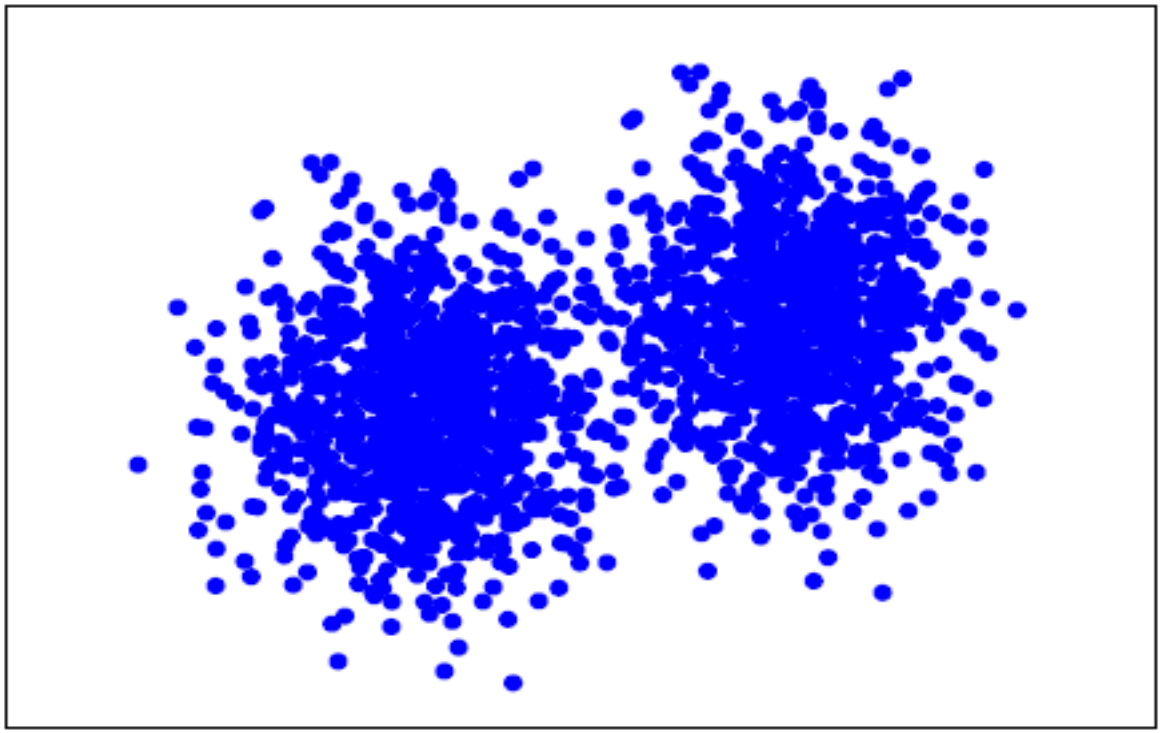


Problems with Gaussian Assumption



- Sample from MLE Gaussian trained on data on last slide.
- Not all data is Gaussian!

Problems with Gaussian Assumption

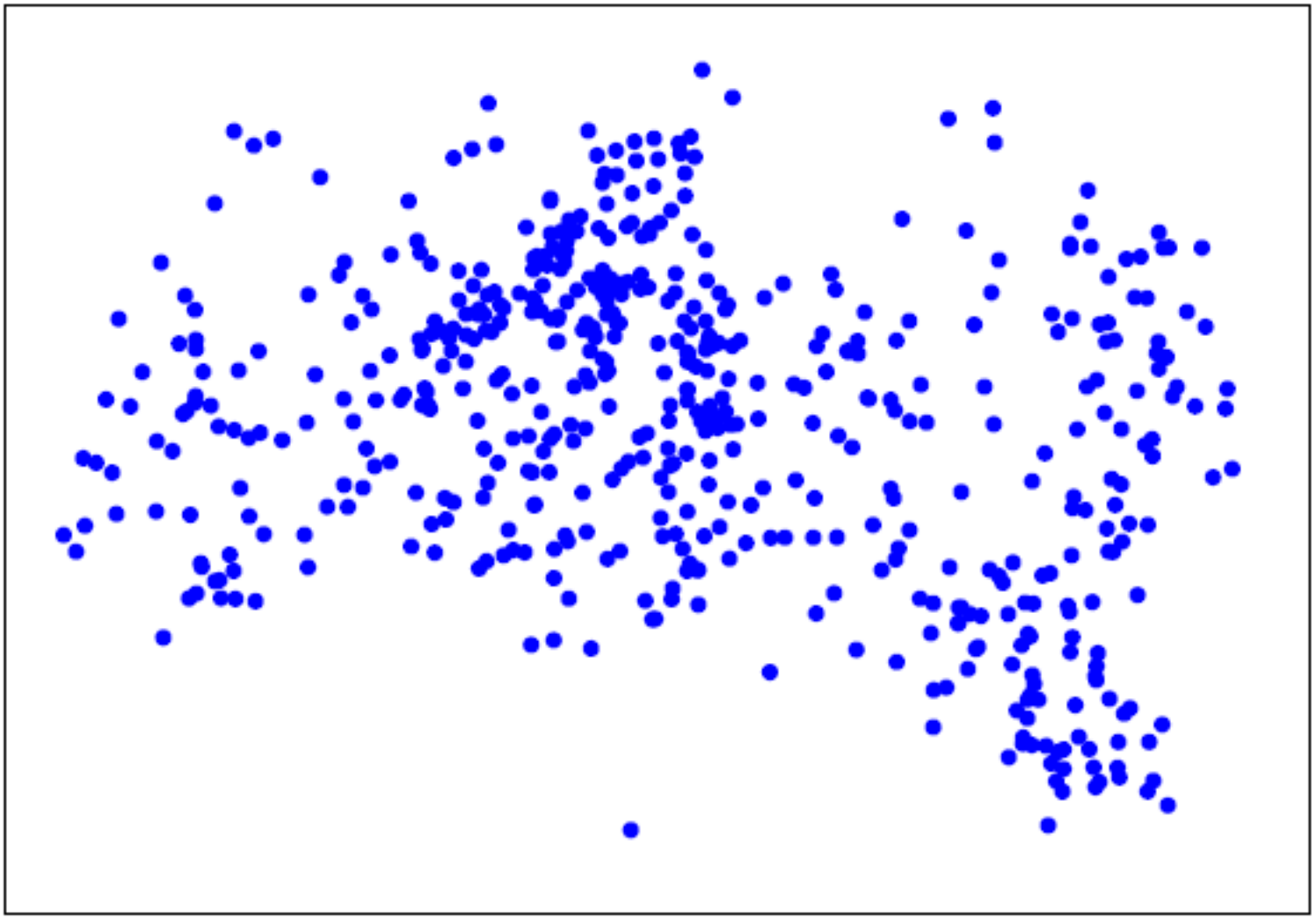


- What can we do? What about *two* Gaussians?

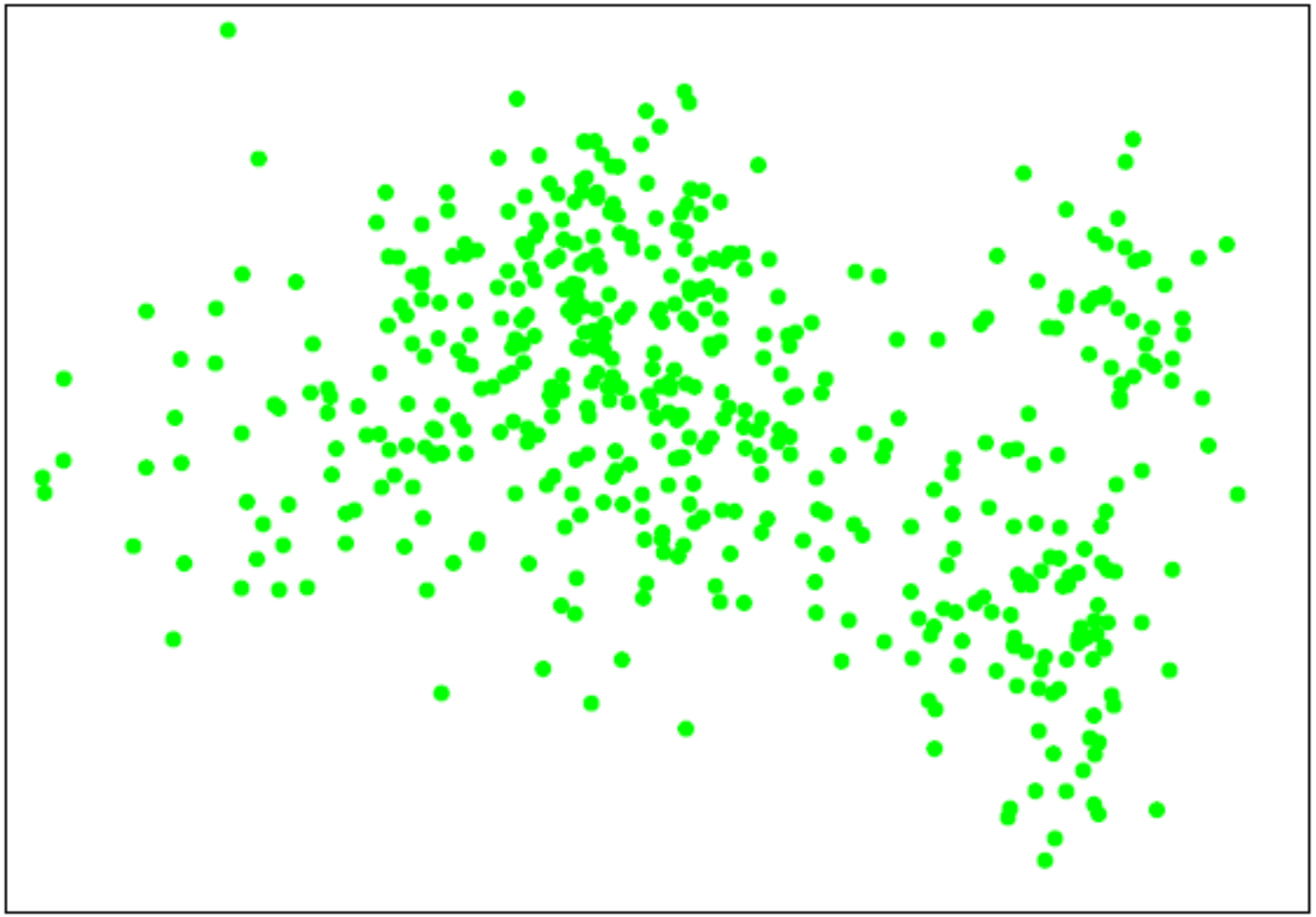
$$P(\mathbf{x}) = p_1 \times \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) + p_2 \times \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

where $p_1 + p_2 = 1$.

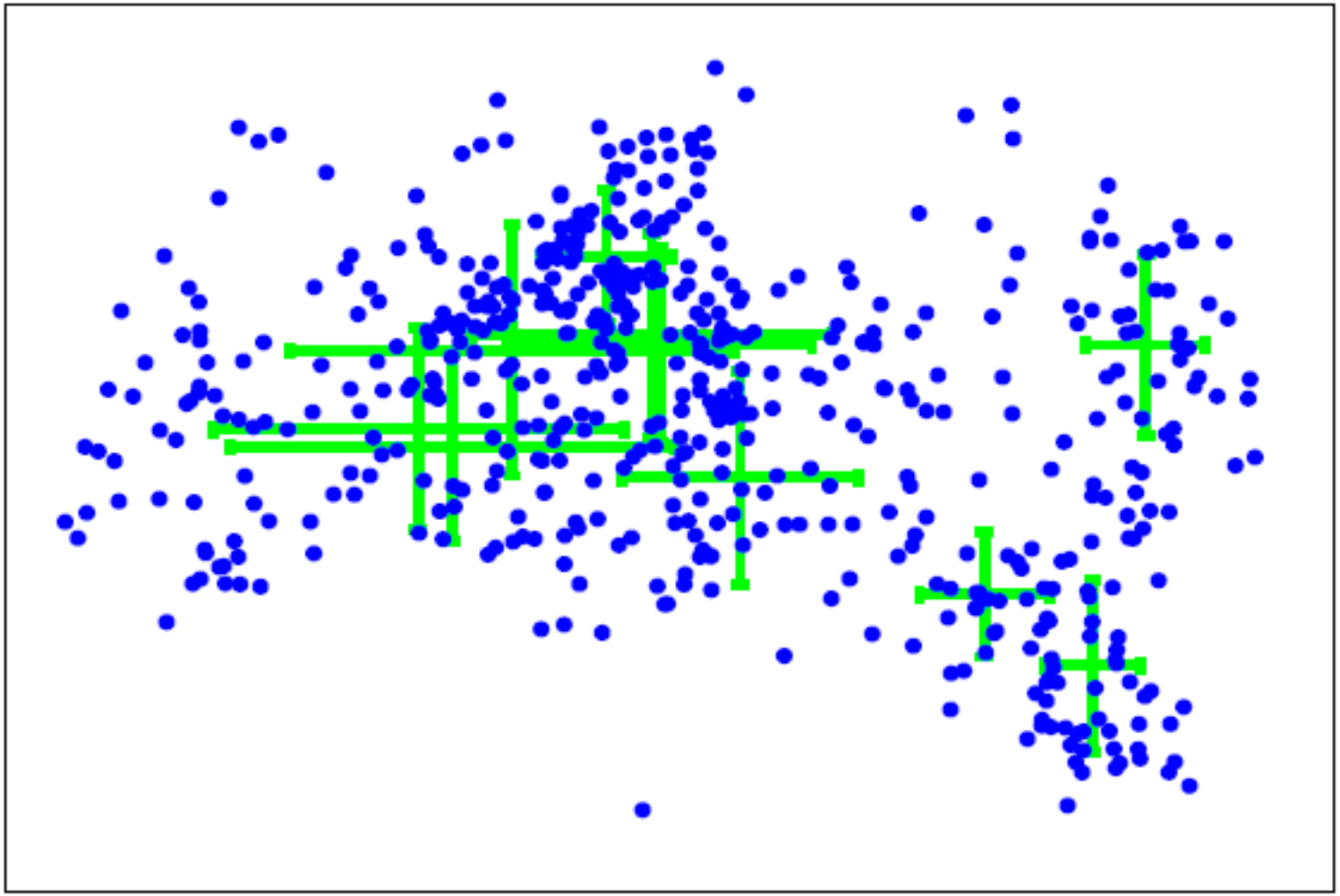
Example: Some Real Acoustic Data



Example: 10-component GMM (Sample)



Example: 10-component GMM (μ's, σ's)



ML Estimation For GMM's

- Given training data, how to estimate parameters . . . i.e., the μ_j , Σ_j , and mixture weights p_j . . .
- To maximize likelihood of data?
  - No closed-form solution. Can't just count and normalize.
- Instead, must use an optimization technique . . .
- To find good local optimum in likelihood.
  - Gradient search
  - Newton's method Tool of choice:
  - The Expectation-Maximization Algorithm

The Basic Idea, using more Formal Terminology

- Initialize parameter values somehow.
- For each iteration . . .
- Expectation step: compute posterior (count) of h for each xi .

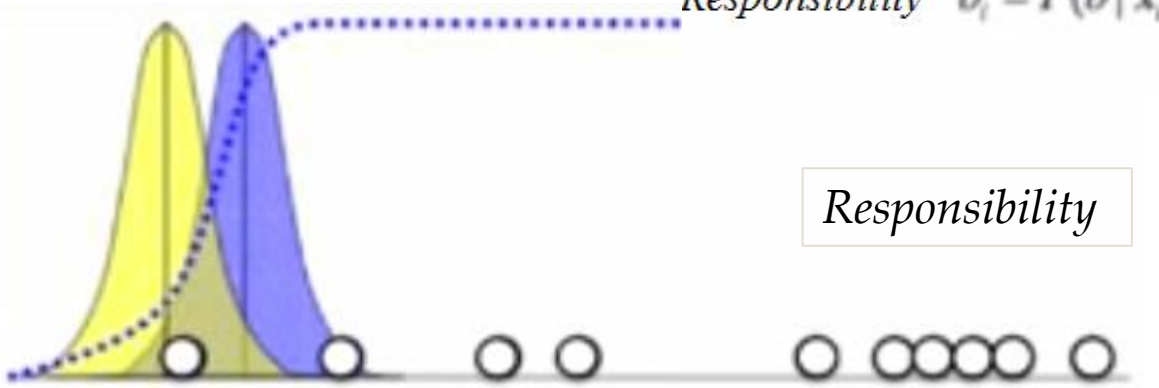
$$\tilde{P}(h|x_i) = \frac{P(h, x_i)}{\sum_h P(h, x_i)}$$

- Maximization step: update parameters.
- Instead of data xi with hidden h, pretend . . .
- Non-hidden data where . . .
- (Fractional) count of each (h, xi) is P˜(h|xi).

E-M Example

- 1 Dimensional example (10 Samples, 2 Gaussian, $\{\mu_1, \sigma^2 \mu_2\ \sigma_2^2\}$

$$P(x_i \mid b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left(-\frac{(x_i - \mu_b)^2}{2\sigma_b^2}\right)$$



$$\text{Responsibility} \quad b_i = P(b \mid x_i) = \frac{P(x_i \mid b)P(b)}{P(x_i \mid b)P(b) + P(x_i \mid a)P(a)}$$

Responsibility

$$p\,(a/x_i) = a_i = 1 - b_i$$

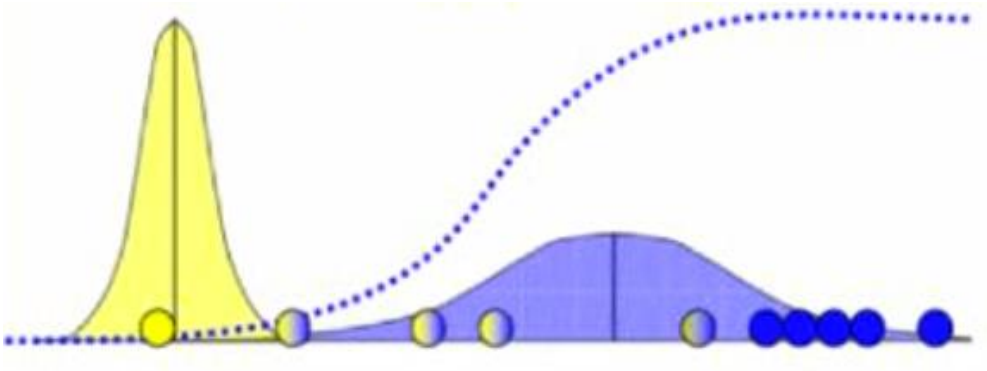
$$\mu_b = \frac{b_1 x_1 + b_2 x_2 + \ldots + b_n x_{n_2}}{b_1 + b_2 + \ldots + b_n}$$

$$\sigma_b^2 = \frac{b_1 (x_1 - \mu_1)^2 + \ldots + b_n (x_n - \mu_n)^2}{b_1 + b_2 + \ldots + b_n}$$

E M Example

$$\mu_a = \frac{a_1 x_1 + a_2 x_2 + \ldots + a_n x_{n_2}}{a_1 + a_2 + \ldots + a_n}$$

$$\sigma_a^2 = \frac{a_1 (x_1 - \mu_1)^2 + \ldots + a_n (x_n - \mu_n)^2}{a_1 + a_2 + \ldots + a_n}$$



$$\mu_k^{new} = \frac{1}{N_k} \sum_{n-1}^{N} \gamma(Z_{nk})\, X_n$$

$$\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(Z_{nk}) * (X_n - \mu_k^{new})(X_n - \mu_k^{new})^T$$

$$N_k = \sum_{n=1}^{N} \gamma(Z_{nk})$$

Example: Training a 2-component GMM

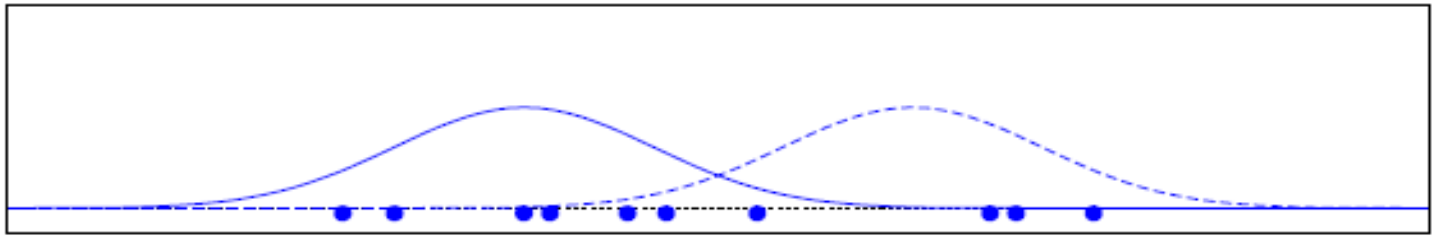
- Two-component univariate GMM; 10 data points.
- The data: $x_1, \ldots, x_{10}$

$$8.4, 7.6, 4.2, 2.6, 5.1, 4.0, 7.8, 3.0, 4.8, 5.8$$

- Initial parameter values:

p_1	μ_1	σ_1^2	p_2	μ_2	σ_2^2
0.5	4	1	0.5	7	1

- Training data; densities of initial Gaussians.



The E Step

x_i	$p_1 \cdot \mathcal{N}_1$	$p_2 \cdot \mathcal{N}_2$	$P(x_i)$	$\tilde{P}(1\|x_i)$	$\tilde{P}(2\|x_i)$
8.4	0.0000	0.0749	0.0749	0.000	1.000
7.6	0.0003	0.1666	0.1669	0.002	0.998
4.2	0.1955	0.0040	0.1995	0.980	0.020
2.6	0.0749	0.0000	0.0749	1.000	0.000
5.1	0.1089	0.0328	0.1417	0.769	0.231
4.0	0.1995	0.0022	0.2017	0.989	0.011
7.8	0.0001	0.1448	0.1450	0.001	0.999
3.0	0.1210	0.0001	0.1211	0.999	0.001
4.8	0.1448	0.0177	0.1626	0.891	0.109
5.8	0.0395	0.0971	0.1366	0.289	0.711

$$\tilde{P}(h|x_i) = \frac{P(h, x_i)}{\sum_h P(h, x_i)} = \frac{p_h \cdot \mathcal{N}_h}{P(x_i)} \qquad h \in \{1, 2\}$$

The M Step

- View: have *non-hidden* corpus for each component GMM.
  - For hth component, have $\tilde{P}(h|x_i)$ counts for event x_i.
- Estimating μ: fractional events.

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i \quad \Rightarrow \quad \mu_h = \frac{1}{\sum_i \tilde{P}(h|x_i)}\sum_{i=1}^{N} \tilde{P}(h|x_i)x_i$$

$$\mu_1 = \frac{1}{0.000 + 0.002 + 0.980 + \cdots} \times (0.000 \times 8.4 + 0.002 \times 7.6 + 0.980 \times 4.2 + \cdots)$$

$$= 3.98$$

- Similarly, can estimate σ_h^2 with fractional events.

The M Step (cont'd)

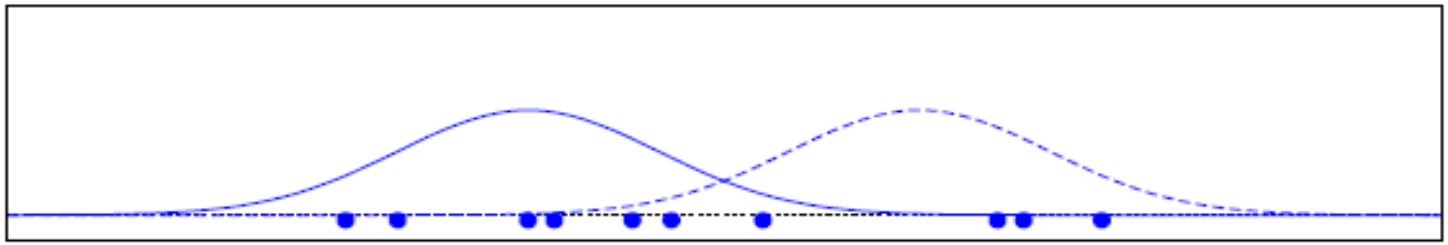
- What about the mixture weights p_h?
  - To find MLE, count and normalize!

$$p_1 = \frac{0.000 + 0.002 + 0.980 + \cdots}{10} = 0.59$$

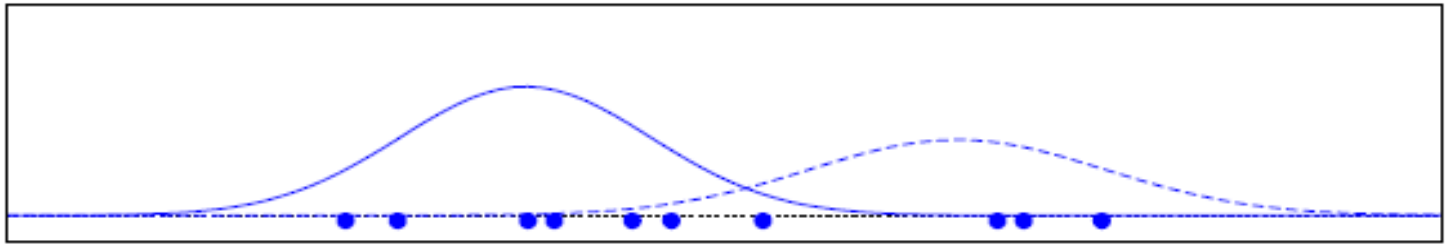
The End Result

iter	p_1	μ_1	σ_1^2	p_2	μ_2	σ_2^2
0	0.50	4.00	1.00	0.50	7.00	1.00
1	0.59	3.98	0.92	0.41	7.29	1.29
2	0.62	4.03	0.97	0.38	7.41	1.12
3	0.64	4.08	1.00	0.36	7.54	0.88
10	0.70	4.22	1.13	0.30	7.93	0.12

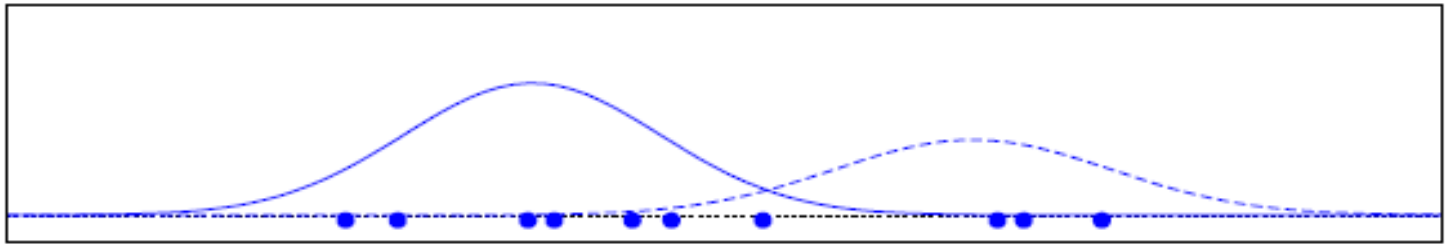
First Few Iterations of EM



iter 0

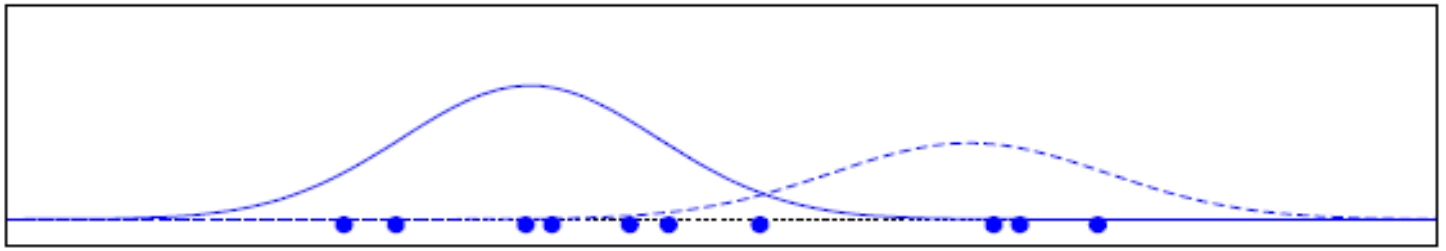


iter 1

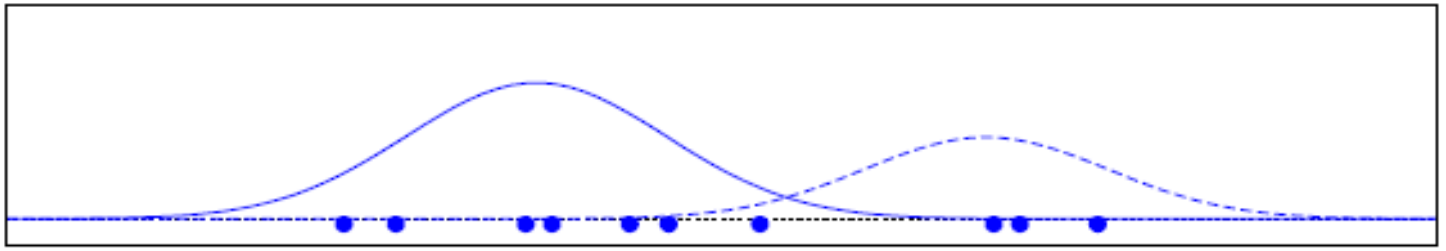


iter 2

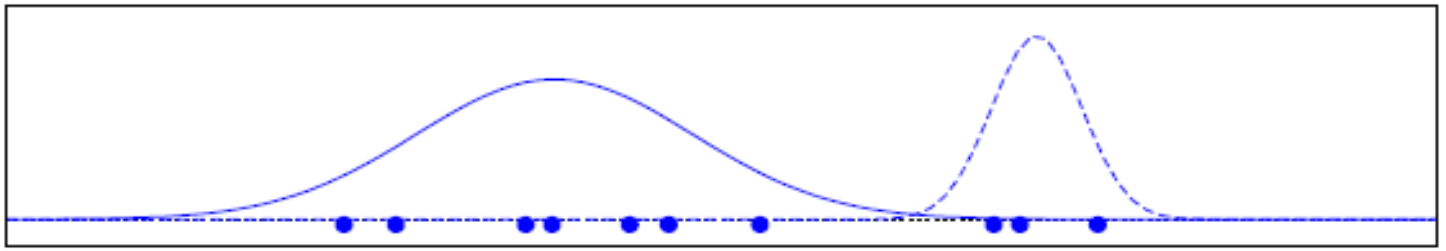
Later Iterations of EM



iter 2

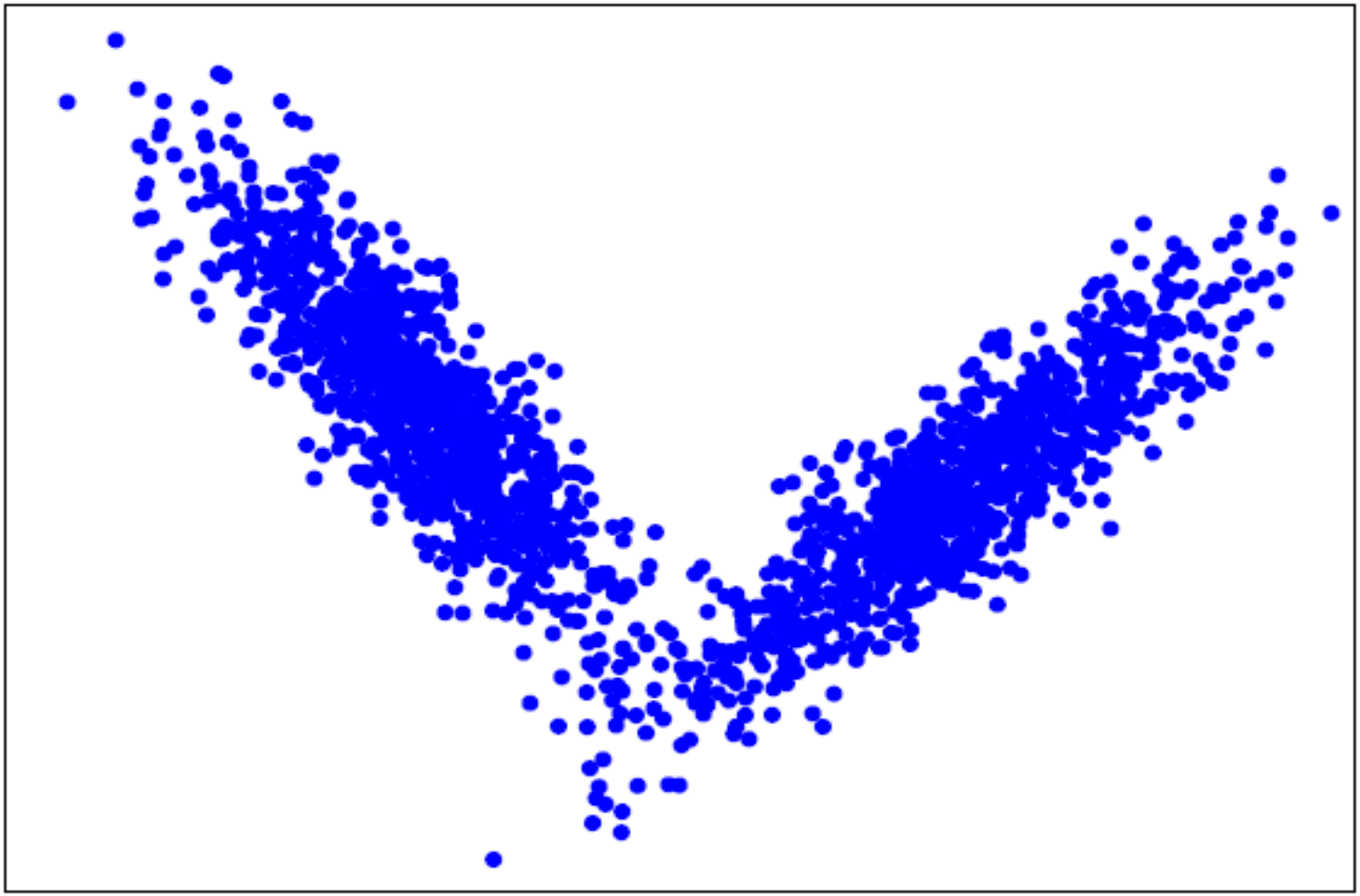


iter 3



iter 10

Another Example Data Set



Question: How Many Gaussians?

- Method 1 (most common): Guess!
- Method 2: Bayesian Information Criterion (BIC)[1].
  - Penalize likelihood by number of parameters.

$$\mathrm{BIC}(C_k) = \sum_{j=1}^{k} \{ -\frac{1}{2} n_j \log |\Sigma_j| \} - Nk(d + \frac{1}{2} d(d+1))$$

  - k = Gaussian components.
  - d = dimension of feature vector.
  - n_j = data points for Gaussian j; N = total data points.
- Discuss!

The Bayesian Information Criterion

- View GMM as way of coding data for transmission.
  - Cost of transmitting model ⇔ number of params.
  - Cost of transmitting data ⇔ log likelihood of data.
- Choose number of Gaussians to minimize cost.

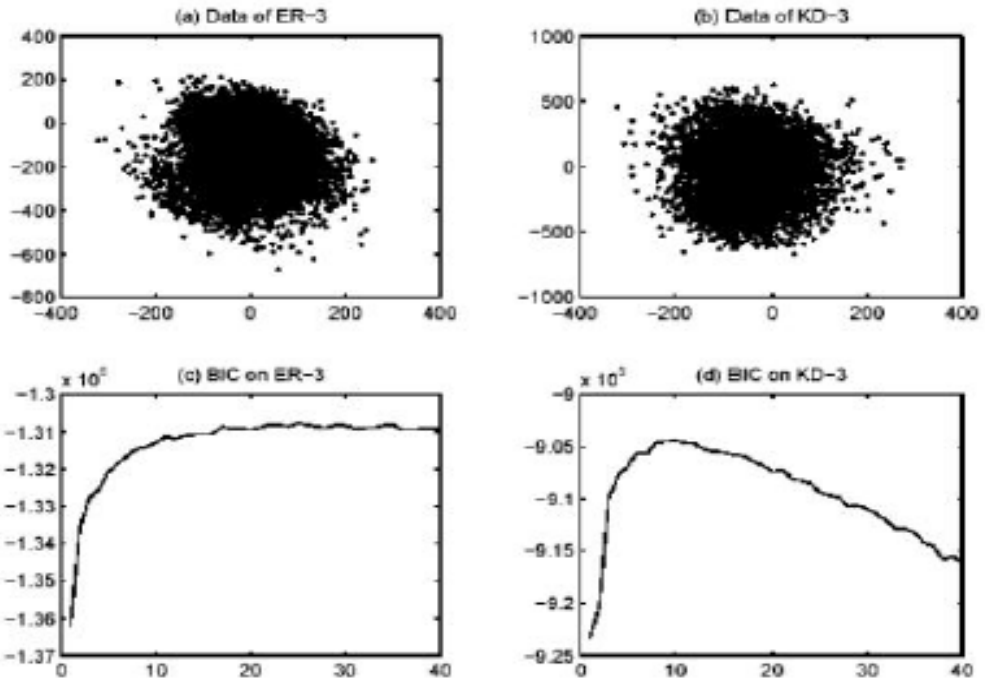


Figure 1. Different degrees of complexity in phone ER-3 and KD-3

Question: How To Initialize Parameters?

- Set mixture weights p_j to $1/k$ (for k Gaussians).
- Pick N data points at random and ...
  - Use them to seed initial values of μ_j.
- Set all σ's to arbitrary value ...
  - Or to global variance of data.
- Extension: generate multiple starting points.
  - Pick one with highest likelihood.

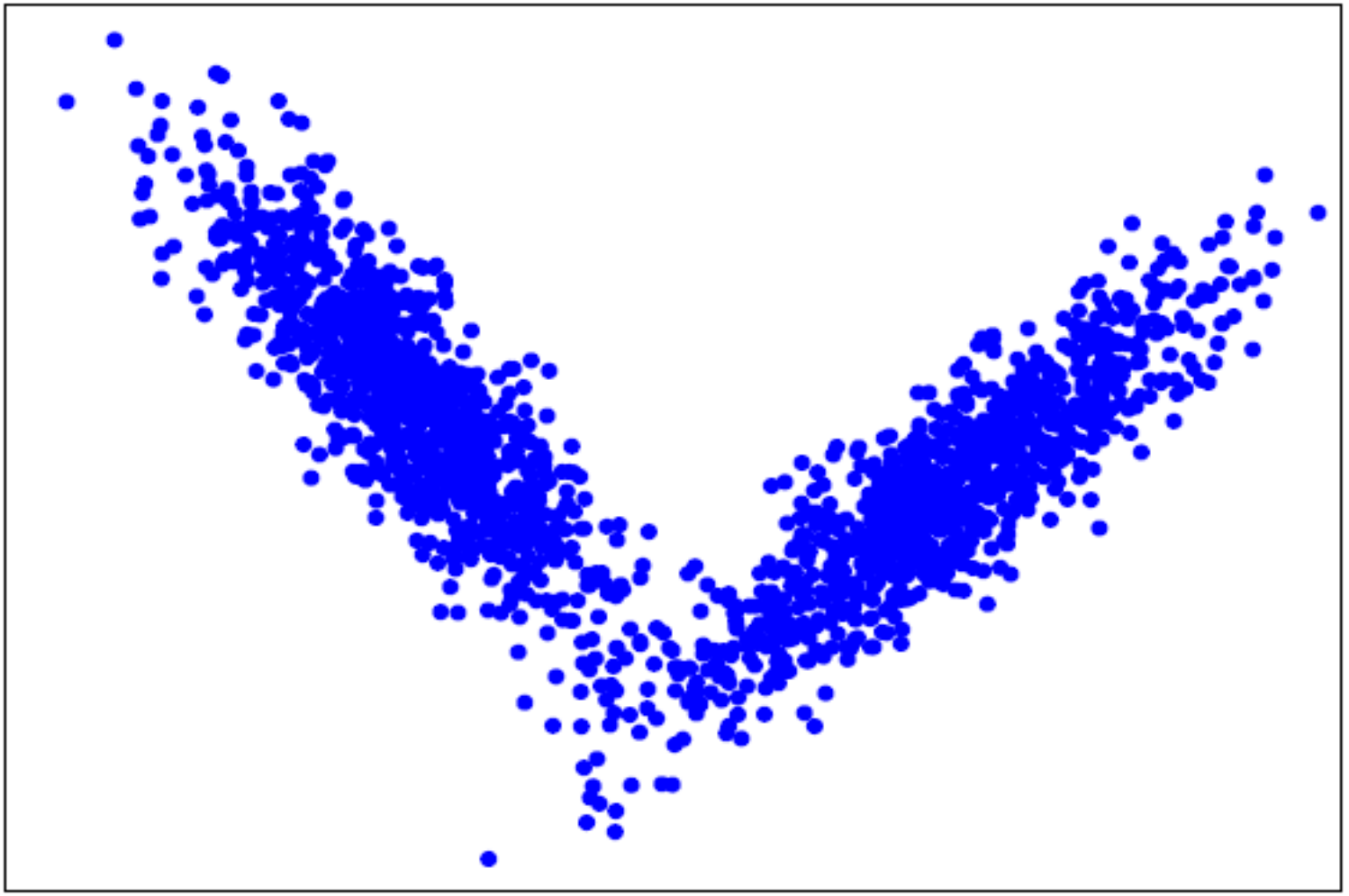
Another Way: Splitting

- Start with single Gaussian, MLE.
- Repeat until hit desired number of Gaussians:
  - Double number of Gaussians by perturbing means ...
  - Of existing Gaussians by $\pm\epsilon$.
  - Run several iterations of EM.

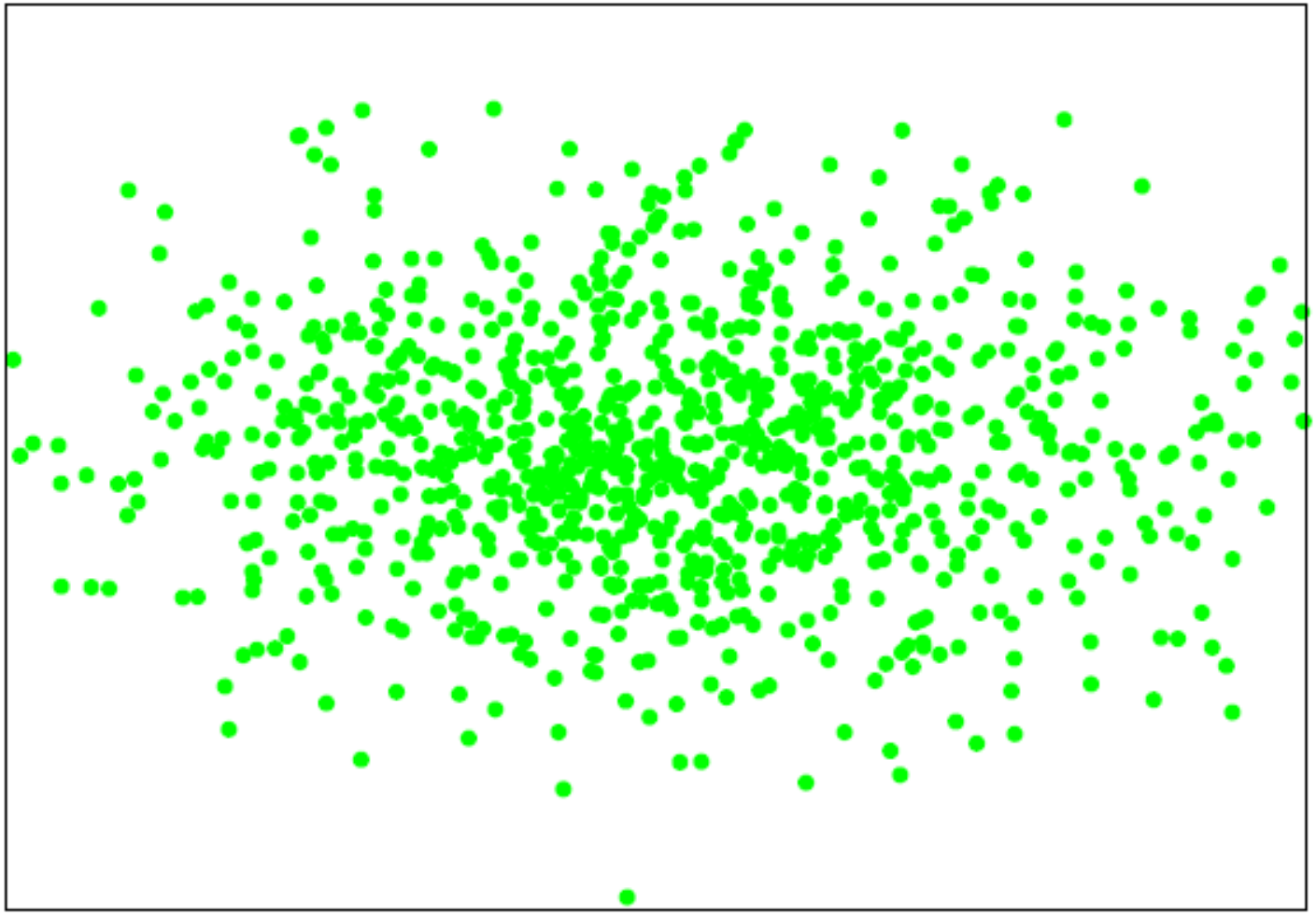
Question: How Long To Train?

- *i.e.*, how many iterations of EM?
- Guess.
- Look at performance on training data.
  - Stop when change in log likelihood per event . . .
  - Is below fixed threshold.
- Look at performance on held-out data.
  - Stop when performance no longer improves.

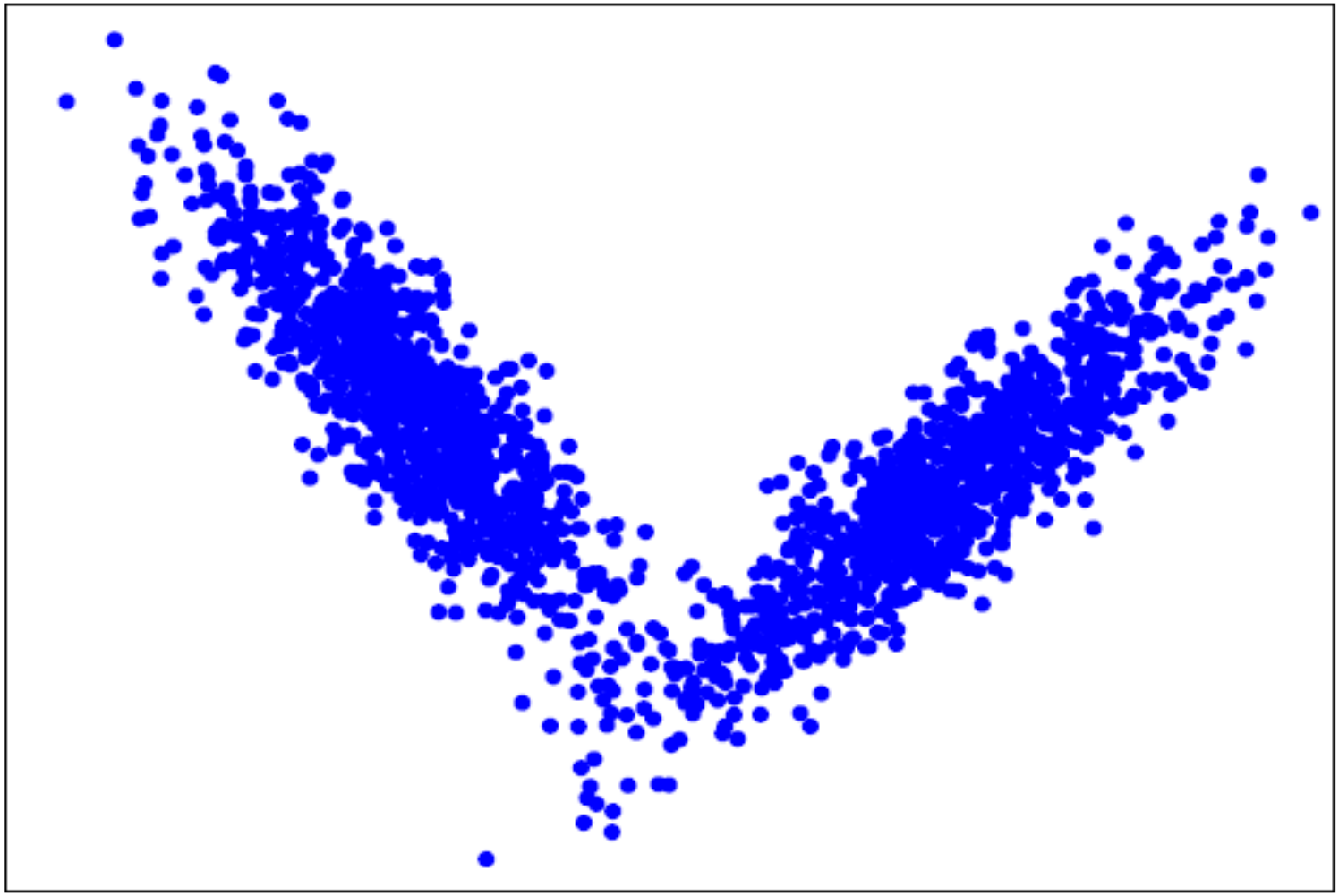
The Data Set



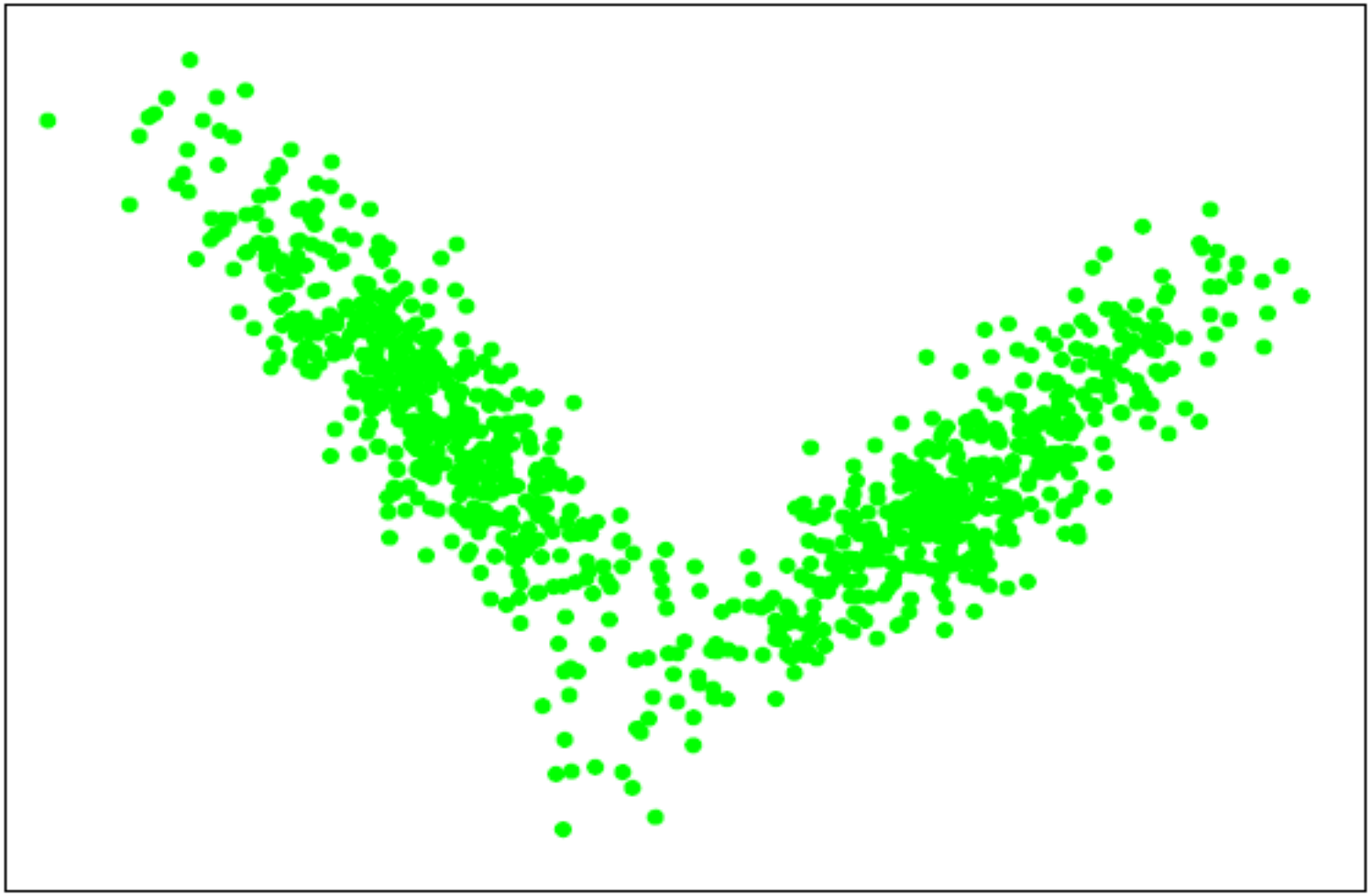
Sample From Best 1-Component GMM



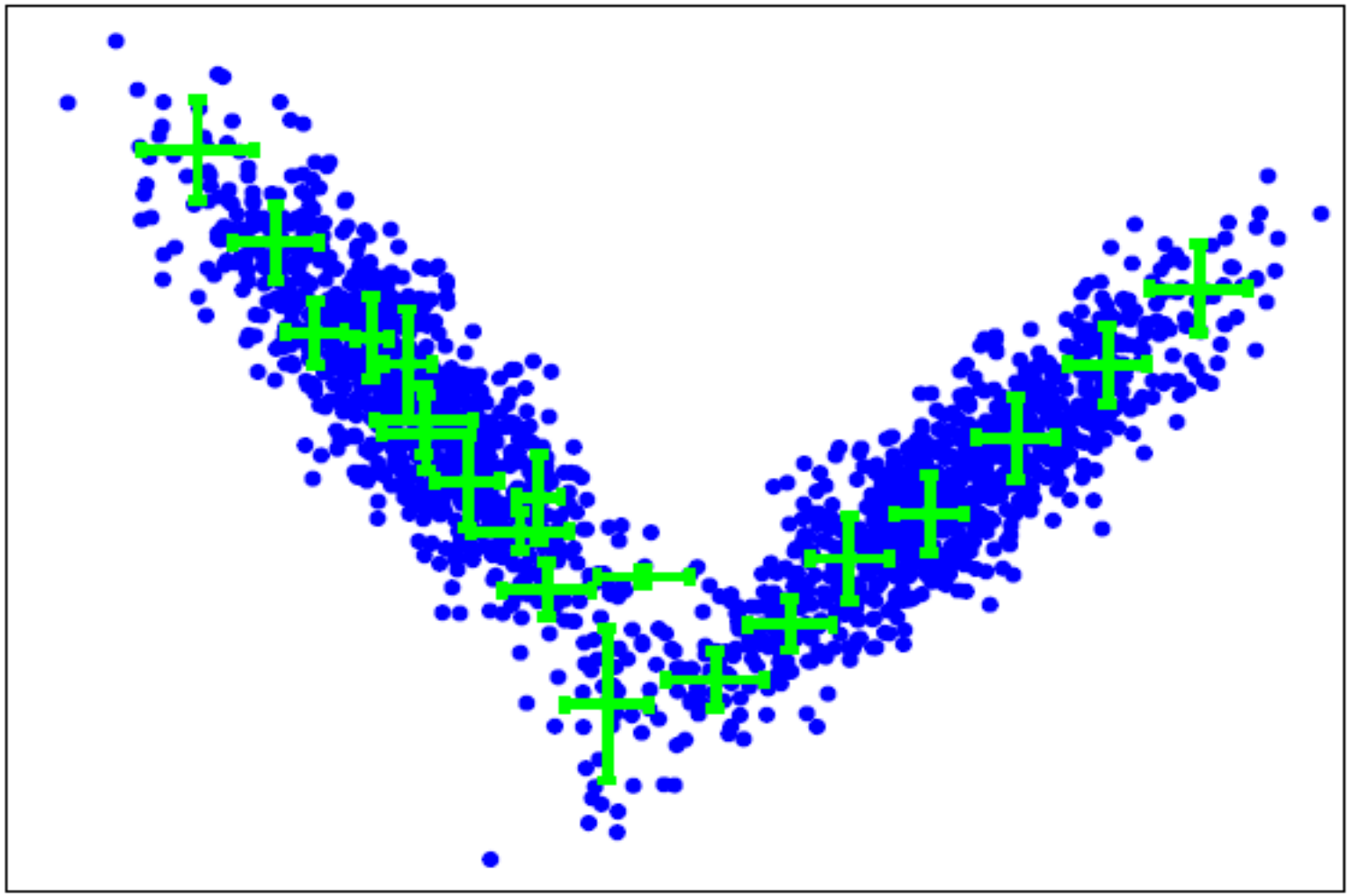
The Data Set, Again



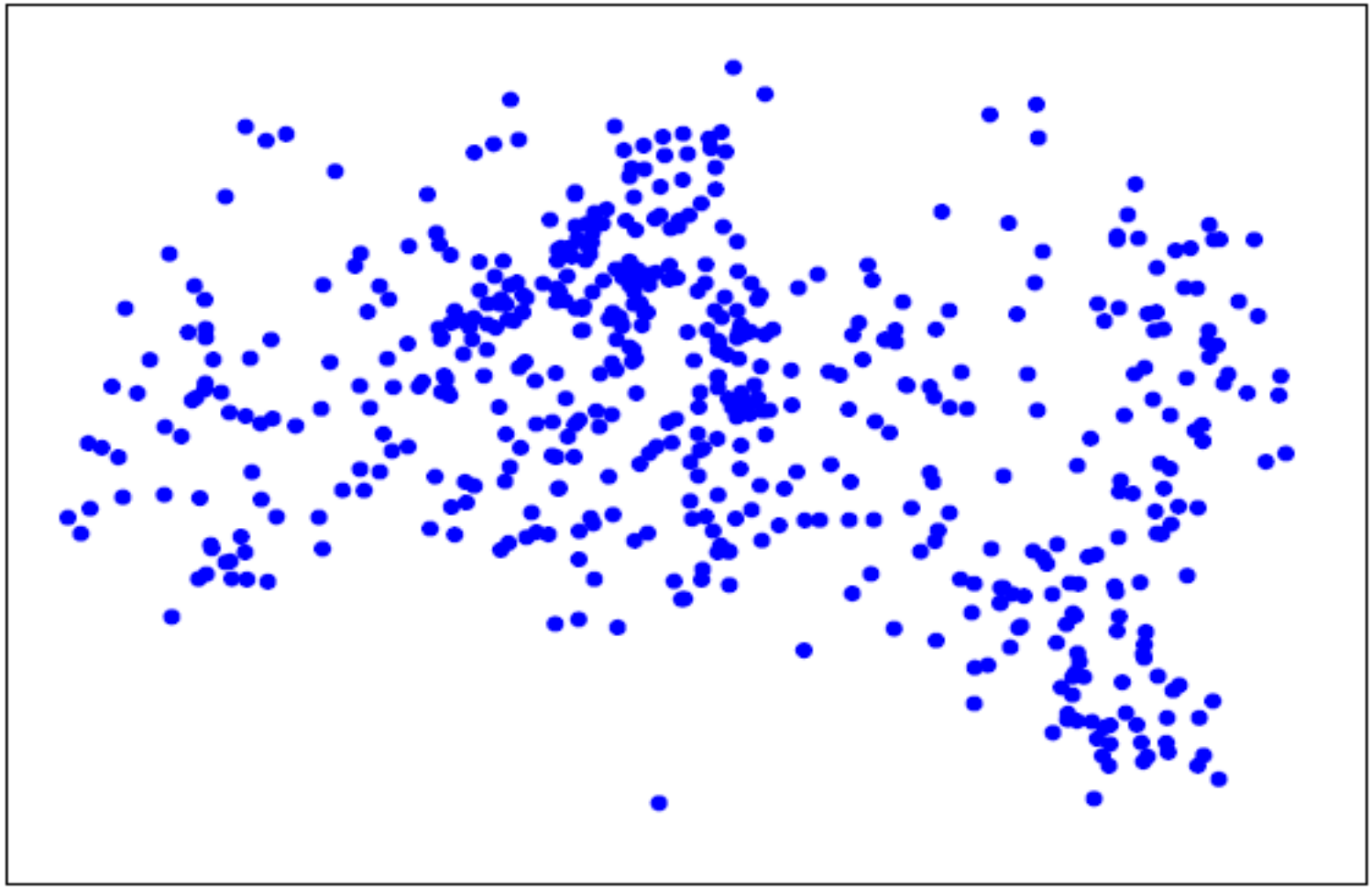
20-Component GMM Trained on Data



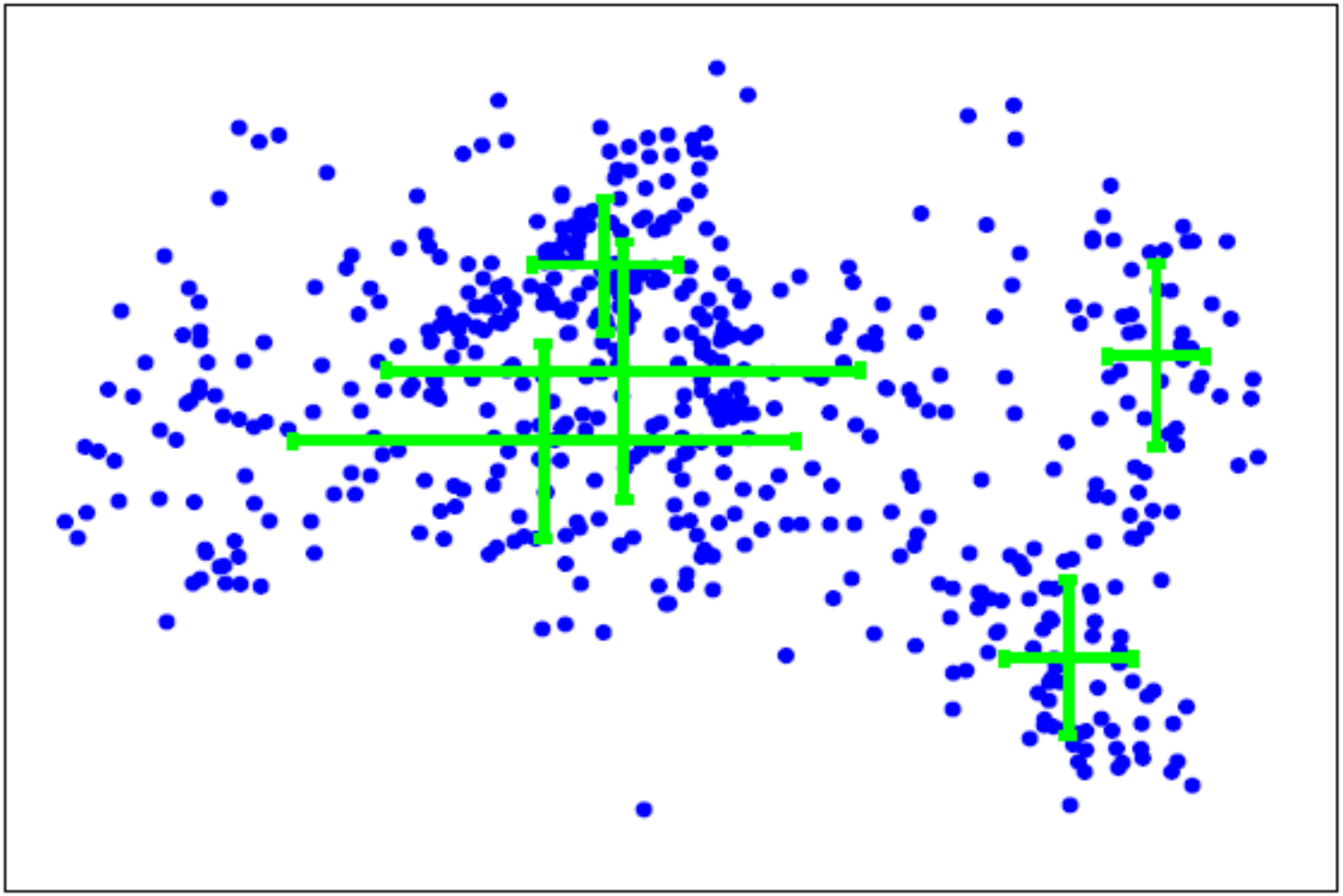
20-Component GMM μ's, σ's



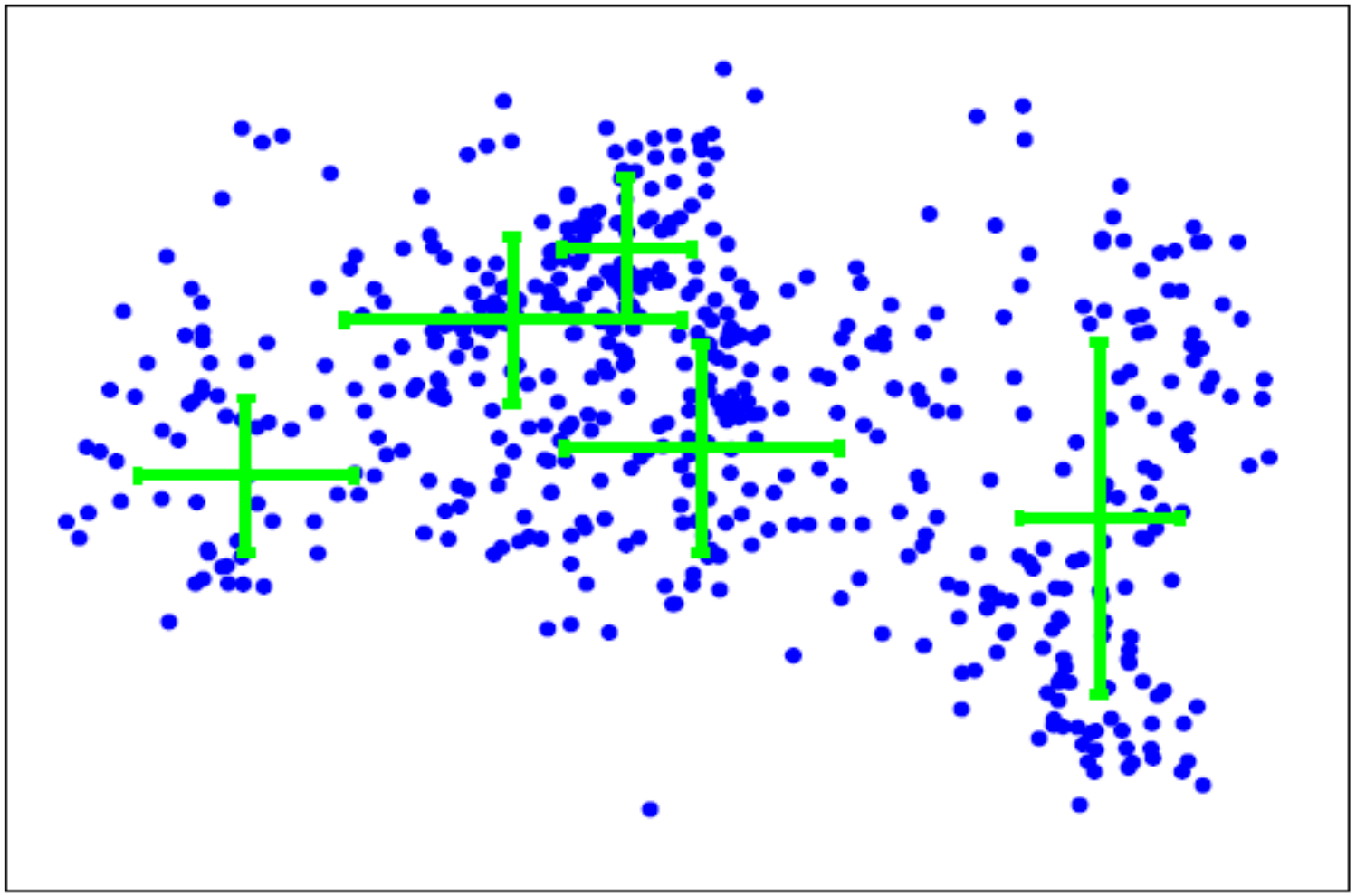
Acoustic Feature Data Set



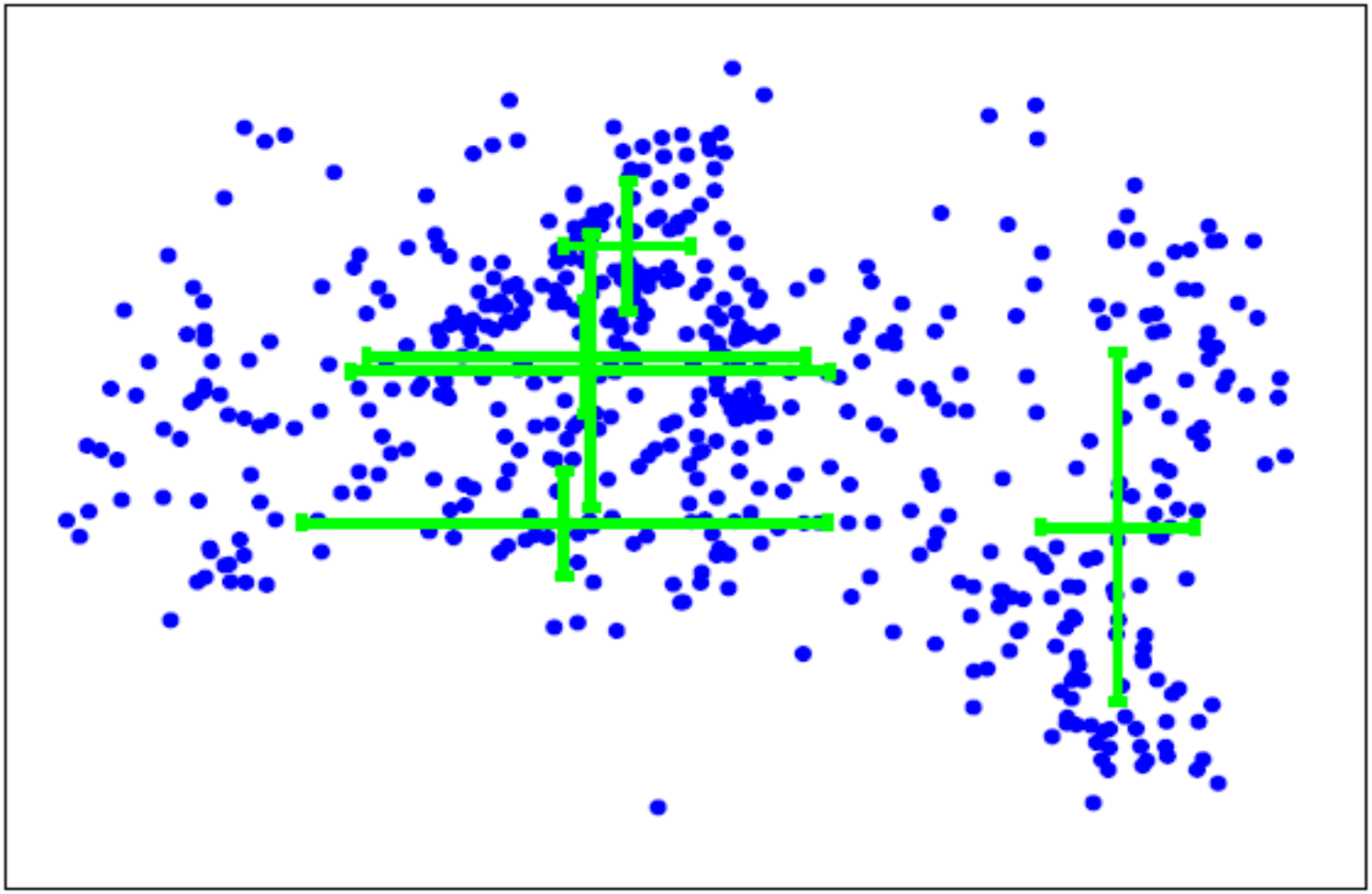
5-Component GMM; Starting Point A



5-Component GMM; Starting Point B



5-Component GMM; Starting Point C



Advantages and Disadvantages of Mixture Models

- Strength
  - Mixture models are more general than partitioning and fuzzy clustering
  - Clusters can be characterized by a small number of parameters
  - The results may satisfy the statistical assumptions of the generative models
- Weakness
  - Converge to local optimal (overcome: run multi-times w. random initialization)
  - Computationally expensive if the number of distributions is large, or the data set contains very few observed data points
  - Need large data sets
  - Hard to estimate the number of clusters

GMM Matlab Function

```
Fs = 8000; % Sampling Freq (Hz)
Duration = 10; % Duration (sec)
 y = audiorecord(Duration*Fs, Fs);
order = 12; nfft = 512; Fs = 8000;
pyulear(speech,order,nfft,Fs);
M=8 %Number of Gaussian component densities
model = gmdistribution.fit(MFCCtraindata, M);
[P, log _ like] = posterior(model,testdata);
% posterior probabilities of each of the M components
```