CS57300
PURDUE UNIVERSITY

APRIL 4, 2019

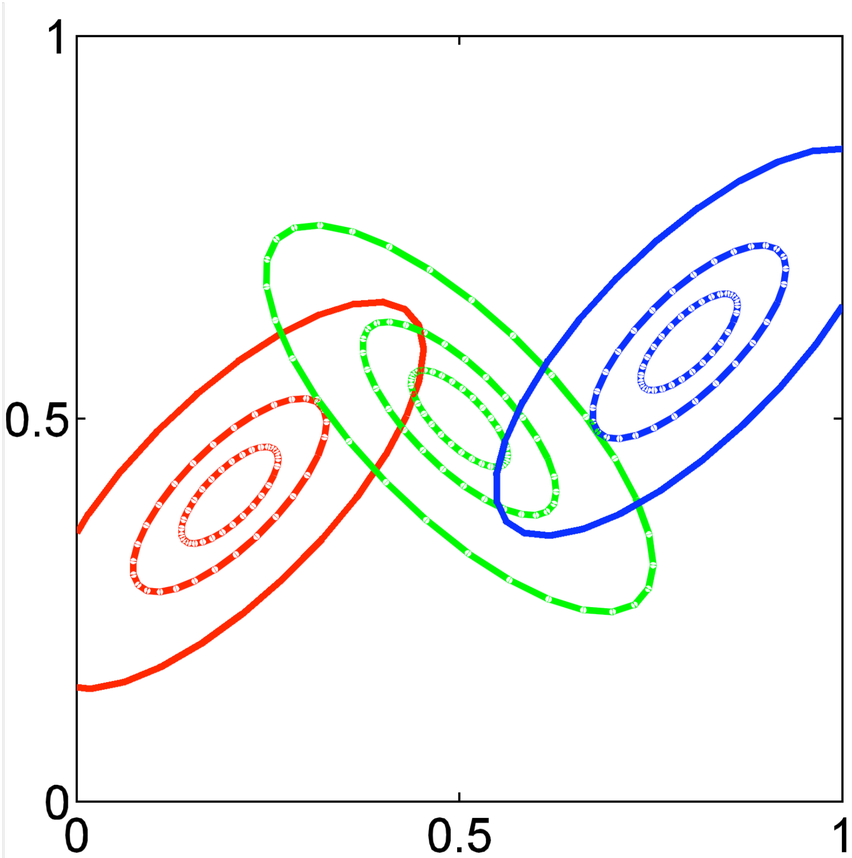
DATA MINING

MODEL-BASED CLUSTERING

PROBABILISTIC MODEL-BASED CLUSTERING

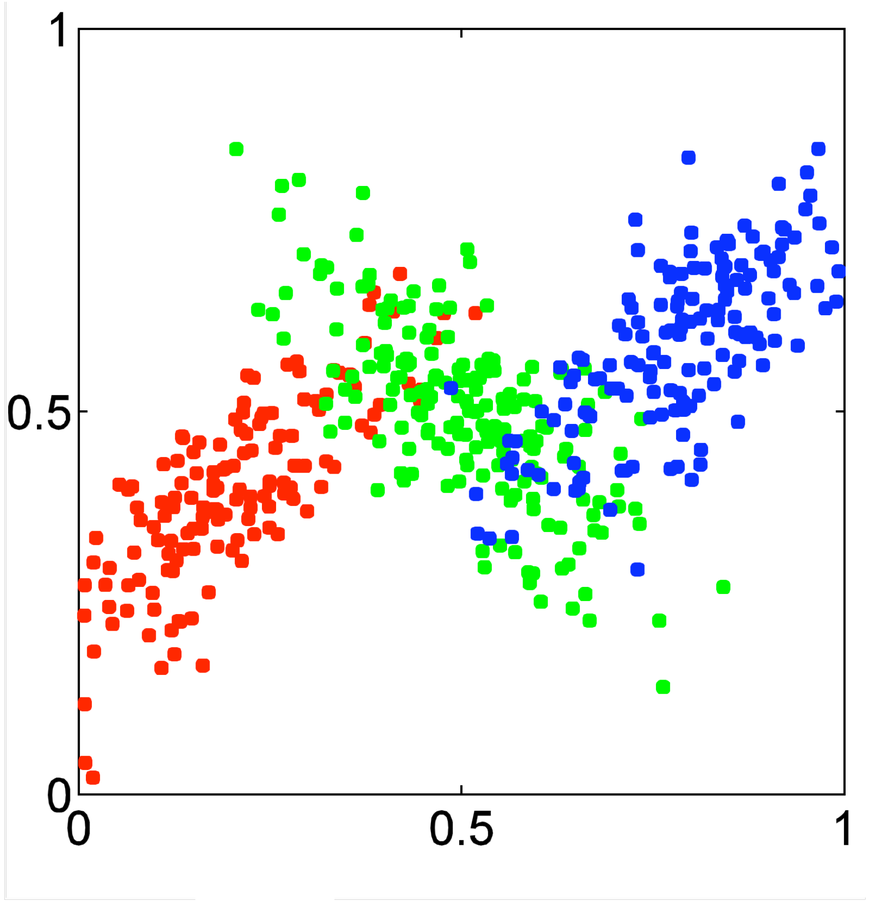
- Assumes a probabilistic model for each underlying cluster (component)
- Mixture model describes data as being generated from a weighted combination of component distributions (e.g., Gaussian)
- Generative process for data:
  - For each data point:
    - Select component *i* randomly based on component weights
    - Generate data point by sampling randomly from component *i*

GAUSSIAN MIXTURE MODEL

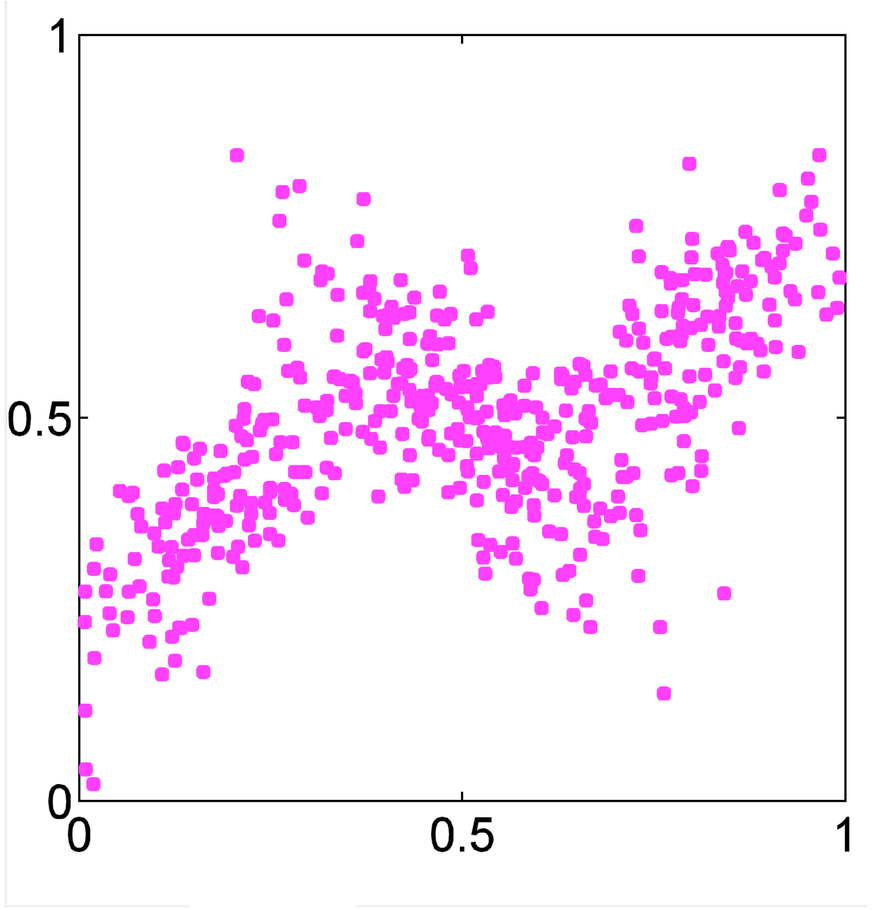


Mixture of three equi-probable Gaussians

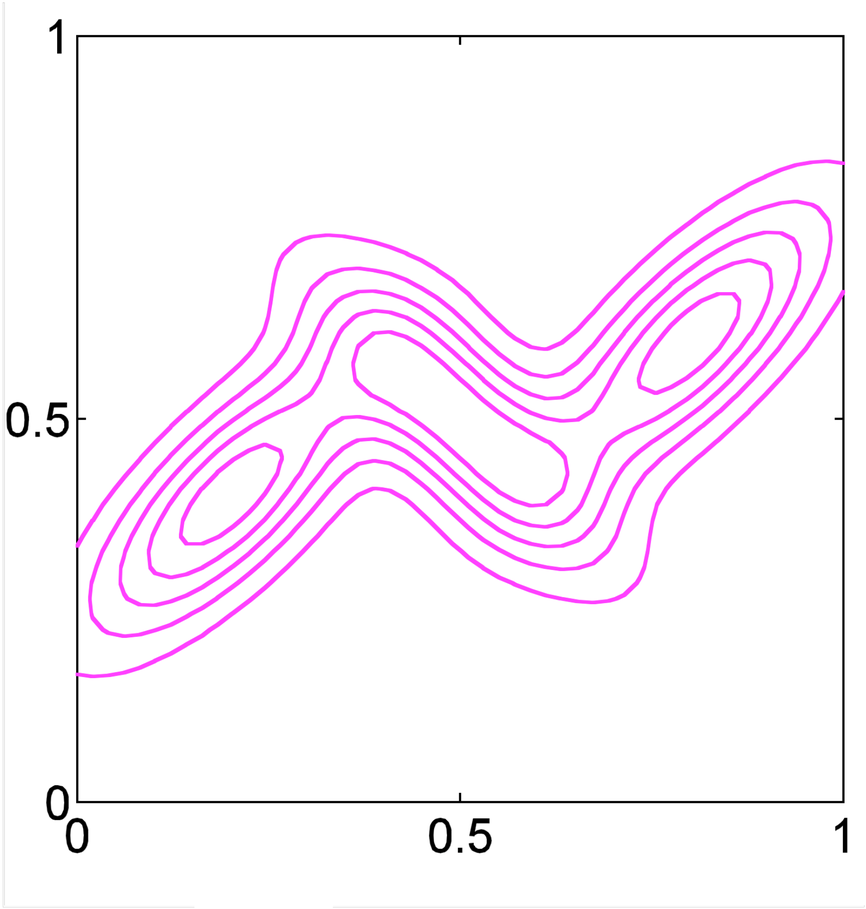
SAMPLE DATASET



UNLABELED DATASET



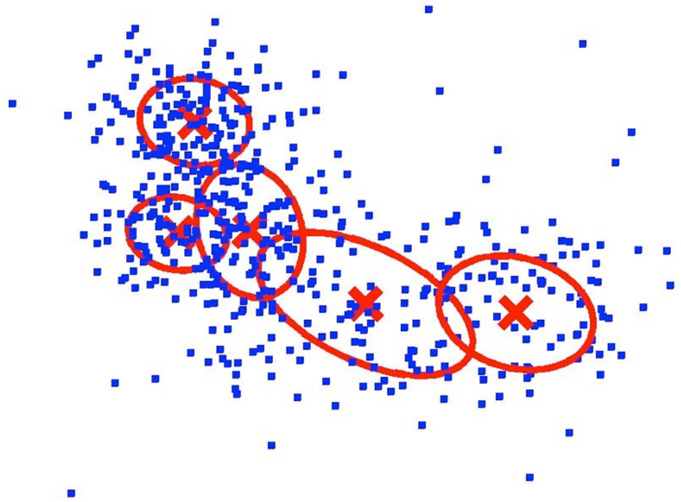
CONTOURS OF PROBABILITY DISTRIBUTION



Mixture of three Gaussians

PROBABILISTIC MIXTURE MODEL

- Instances represented as a weighted combination of *mixture* distributions



$$f(x) = \sum_{k=1}^{K} w_k f_k(x;\theta)$$

probability of observing x

likelihood of x being generated from cluster k

likelihood of point belonging to cluster k

GENERATIVE PROCESS (REVISITED)

- Assume that the data are generated from a mixture of K multi-dimensional Gaussians, where each component has parameters: $N_k(\mu_k, \Sigma_k)$
- For each data point:
  - Pick component Gaussian randomly with probability *p*(*k*)
  - Draw point from that Gaussian randomly by sampling from: $N_k(\mu_k, \Sigma_k)$

$$p(x) = \sum_{k=1}^{K} p(k)p(x|k)$$

$$= \sum_{k=1}^{K} p(k)p\Big(x|x \sim N(\mu_k, \Sigma_k)\Big)$$

MULTIDIMENSIONAL GAUSSIAN

- A multi-dimensional Gaussians, for data with p dimensions is specified as follows

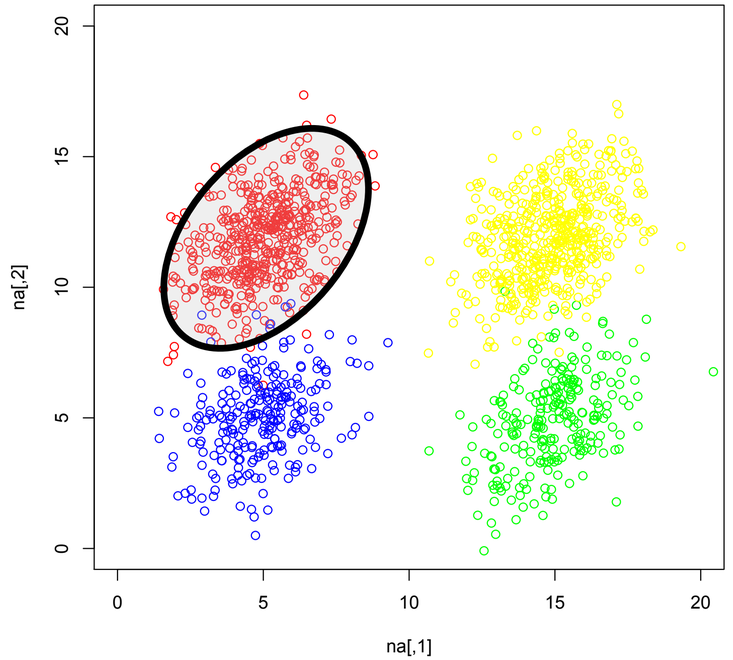
$$x \sim \mathcal{N}(\mu, \Sigma)$$

where:

$$\mu = \Big(E[X_1], ..., E[X_p]\Big)$$

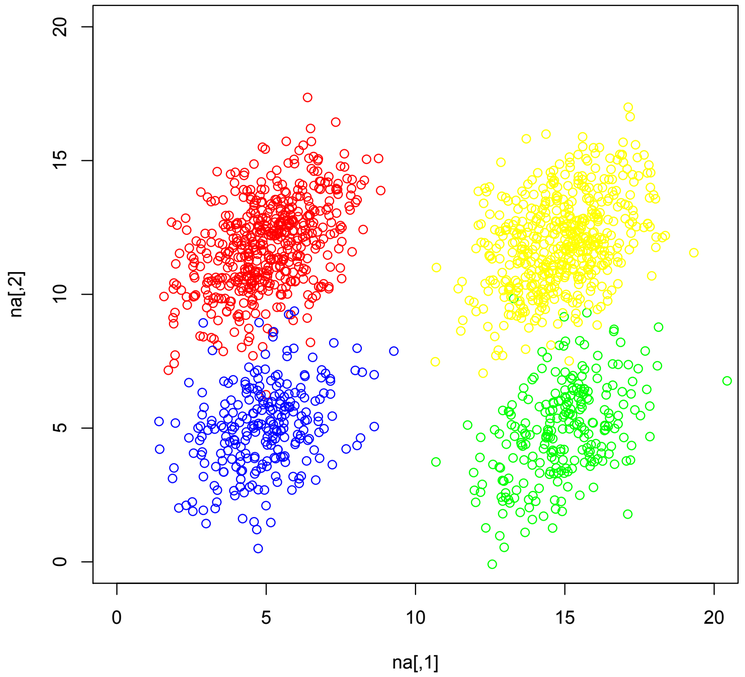
$$\Sigma = \begin{bmatrix} Var(X_1) & ... & Cov(X_1, X_p) \\ ... & ... & ... \\ Cov(X_1, X_p) & ... & Var(X_p) \end{bmatrix}$$

$$p(\mathbf{x}) = p(x_1, ..., x_p) = \frac{1}{\sqrt{(2\pi)^p |\Sigma|}} exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right)$$



EXAMPLE GENERATIVE PROCESS

```
sigma <- matrix(c(2,1,1,3),2,2)
na=mvrnorm(n=500, c(5,12), sigma)
nb=mvrnorm(n=250, c(5,5), sigma)
nc=mvrnorm(n=250, c(15,5), sigma)
nd=mvrnorm(n=500, c(15,12), sigma)
d=rbind(na,nb,nc,nd)
plot(na,xlim=c(0,20),ylim=c(0,20),col='red')
points(nb,col='blue')
points(nc,col='green')
points(nd,col='yellow')
```



Parameters

$$p(k) = [0.333, 0.167, 0.167, 0.333]$$

$$\mu_1 = [5, 15], \mu_2 = [5, 5], \mu_3 = [15, 5], \mu_2 = [15, 12]$$

$$\Sigma = \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_1, X_2) & Var(X_2) \end{bmatrix} \qquad \Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}$$

LEARNING THE MODEL FROM DATA

- We want to invert this process
- Given the dataset, find the parameters
  - Mixing coefficients $p(k)$
  - Component means and covariance matrix $N_k(\mu_k, \Sigma_k)$
- Once the parameters are learned, we can decide the cluster membership of a data point using the Bayes rule
  - $p(c \mid x) = \frac{p(c)p(x \mid c)}{p(x)}$

HOW TO LEARN GMMS?

SCORE FUNCTION FOR GMM

- **Log likelihood** takes the following form (for model M={w,μ,Σ}):

$$
\begin{aligned}
log\, p(D|w,\mu,\Sigma) &= \sum_{n=1}^{N} log\, p(x_n|M) \\
&= \sum_{n=1}^{N} log \left[\sum_{k=1}^{K} p(x_n|k,M) P(k|M) \right] \\
&= \sum_{n=1}^{N} log \left[\sum_{k=1}^{K} w_k N(x_n|\mu_k,\Sigma_k) \right]
\end{aligned}
$$

- Note the sum over components is inside the log
- There is no closed form solution for the MLE

HIDDEN CLUSTER MEMBERSHIP VARIABLES

- Consider k cluster indicator variables for example x_n: $\mathbf{z_n} = [z_{n1}, ..., z_{nk}]$ which equals 1 for the cluster that x_n is a member of, and 0 otherwise
- If we knew the values of the hidden cluster membership variables (z) we could easily maximize the complete data log-likelihood, which has a closed form solution:

$$
\begin{aligned}
log\, p(D, \mathbf{z}|w, \mu, \Sigma) &= \sum_{n=1}^{N} log\Big[\sum_{k=1}^{K} z_{nk} \cdot w_k N(x_n|\mu_k, \Sigma_k)\Big] \\
&= \sum_{n=1}^{N} log\,\Big[w_{k'} N(x_n|\mu_{k'}, \Sigma_{k'})\Big] \quad \text{where } z_{nk'} \neq 0 \\
&= \sum_{n=1}^{N} log\, w_{k'} + \qquad log\, N(x_n|\mu_{k'}, \Sigma_{k'}) \quad \text{where } z_{nk'} \neq 0
\end{aligned}
$$

- Unfortunately we don't know the values for the hidden variables!
- But, for given set of parameters we can compute the ***expected values*** of the hidden variables (cluster memberships)

POSTERIOR PROBABILITIES OF CLUSTER MEMBERSHIP

- We can think of the mixing coefficients as **prior** probabilities for cluster membership
- Then for a given example x_n, we can evaluate the corresponding **posterior** probabilities of **cluster membership** with Bayes theorem:

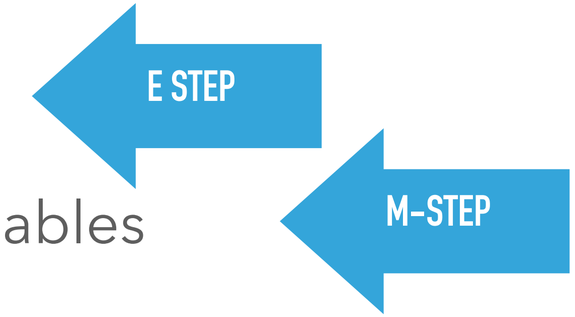
$$\gamma_k(x_n) \equiv p(z_{nk} = 1|x_n) = \frac{p(x_n|z_{nk} = 1)p(z_{nk} = 1)}{p(x_n)}$$

$$= \frac{w_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} w_j N(x_n|\mu_j, \Sigma_j)}$$

cluster membership for x

EXPECTATION-MAXIMIZATION (EM) ALGORITHM

- Popular algorithm for parameter estimation in data with hidden/unobserved values
  - Hidden variables=cluster membership
- Basic idea
  - Initialize hidden variables and parameters
  - Predict values for hidden variables given current parameters
  - Estimate parameters given current prediction for hidden variables
  - Repeat



EM FOR GMM

- Suppose we make a guess for the parameters values
- Use these to evaluate posterior probs for cluster memberships (using Bayes rule)

$$\Gamma(x_n) = [\gamma_1(x_n), ..., \gamma_K(x_n)]$$

E-STEP

- Now compute the log-likelihood using predicted cluster memberships

$$log\ p(x, z|\theta) = \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma_i(x_n) \left[log\ w_k + log\ N(x_n|\mu_k, \Sigma_k) \right]$$

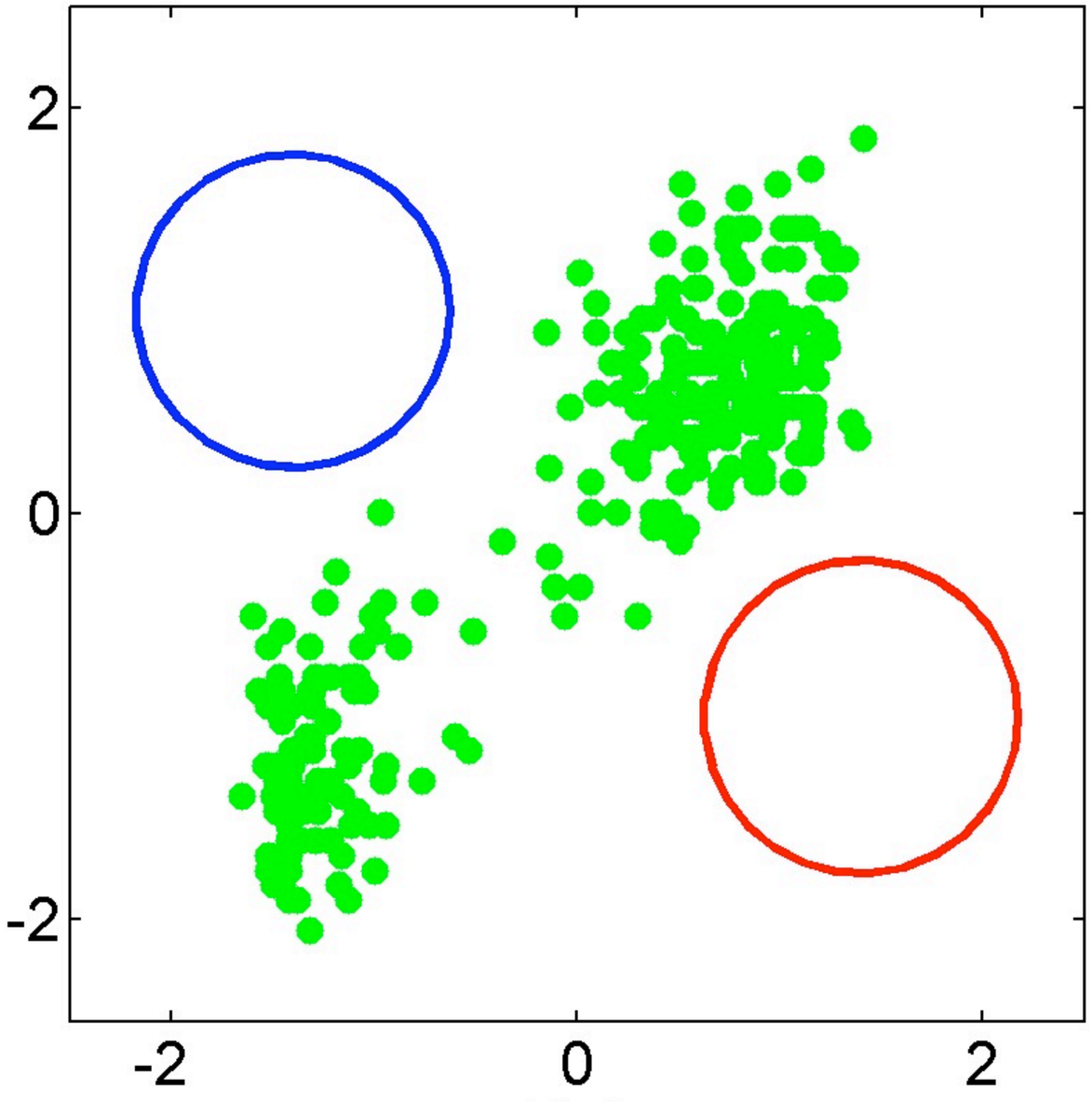
- Use expected complete likelihood to determine MLE for parameters (w_k,μ_k,Σ_k)

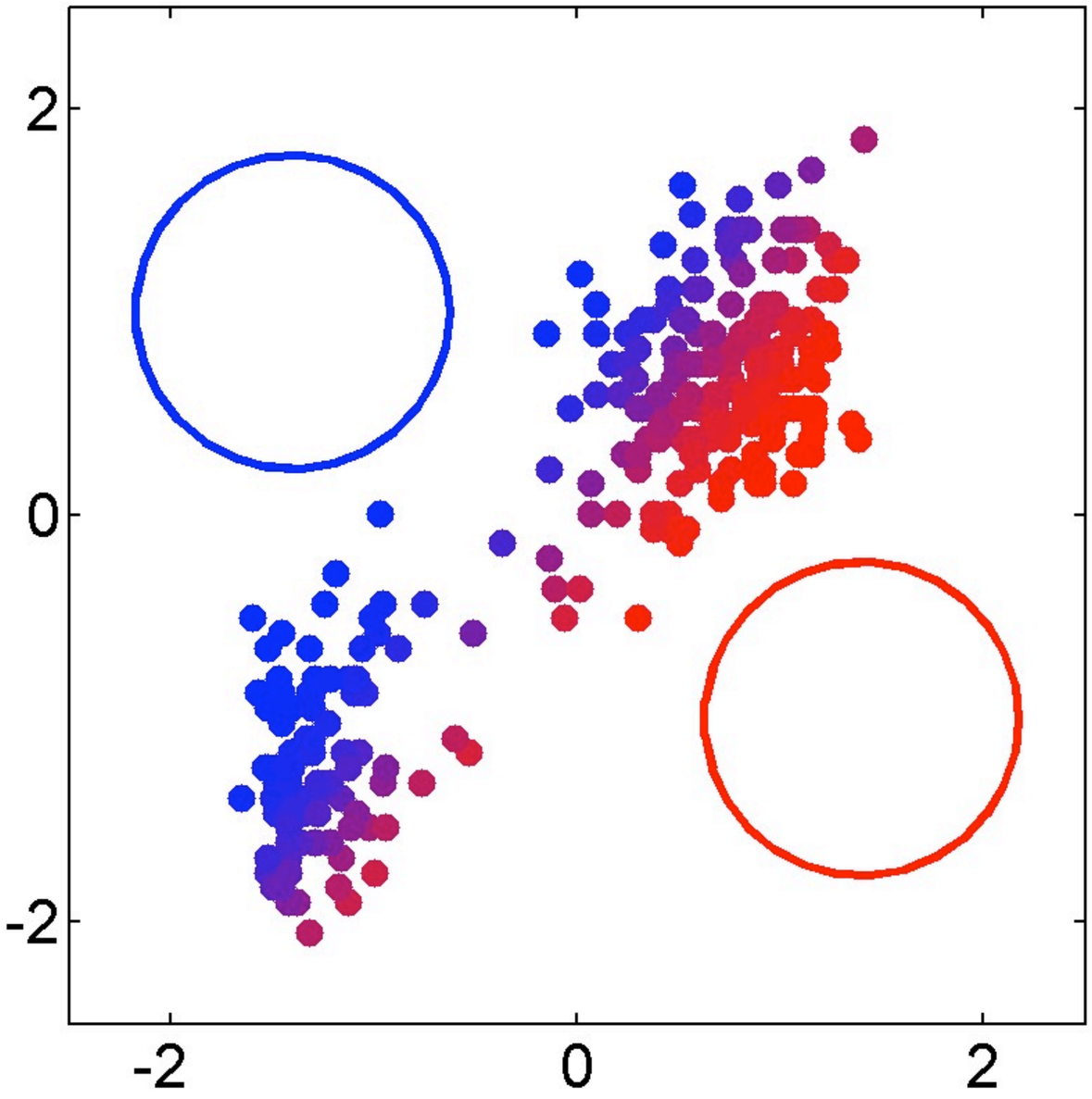
M-STEP

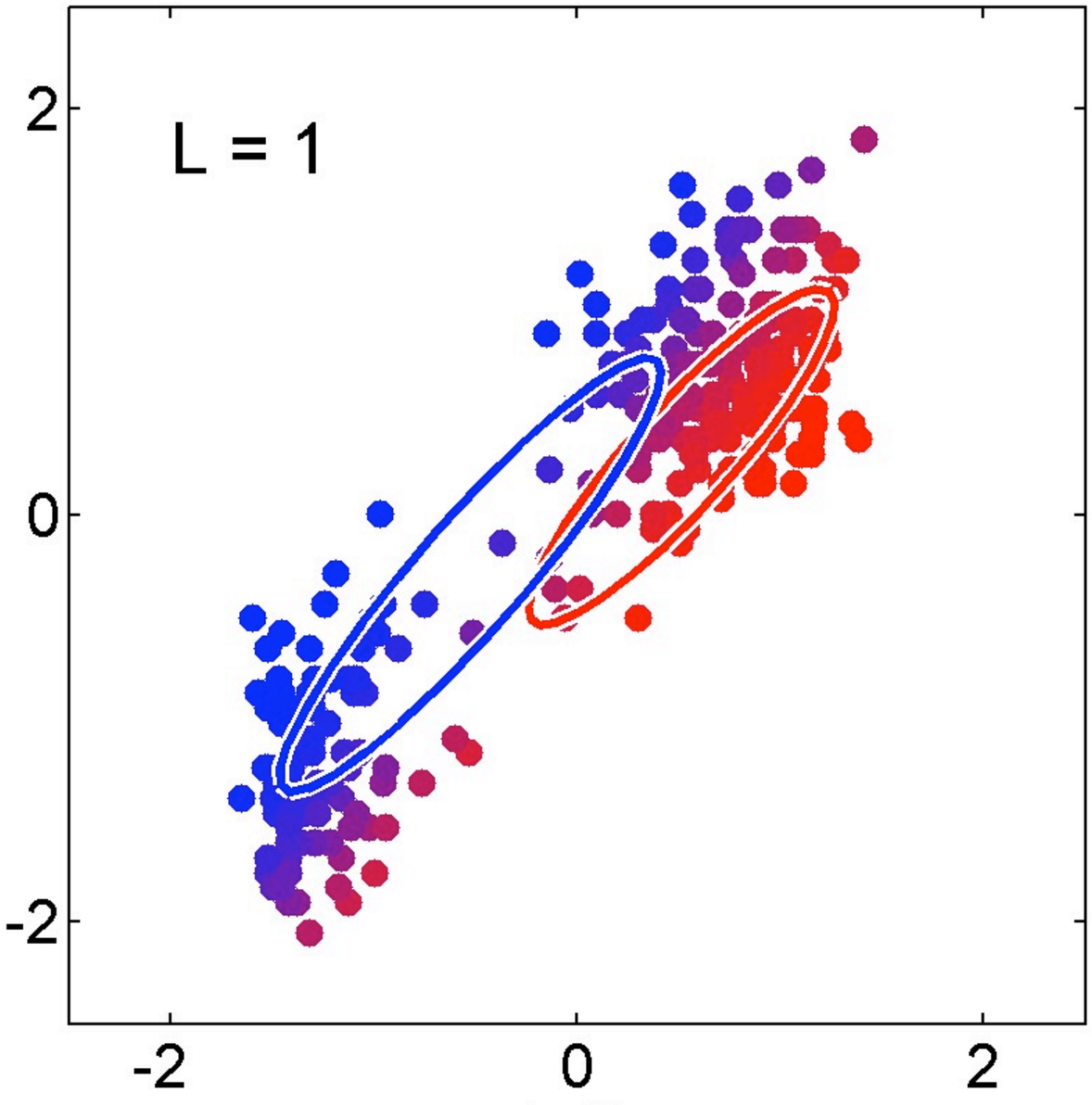
MORE ON EM

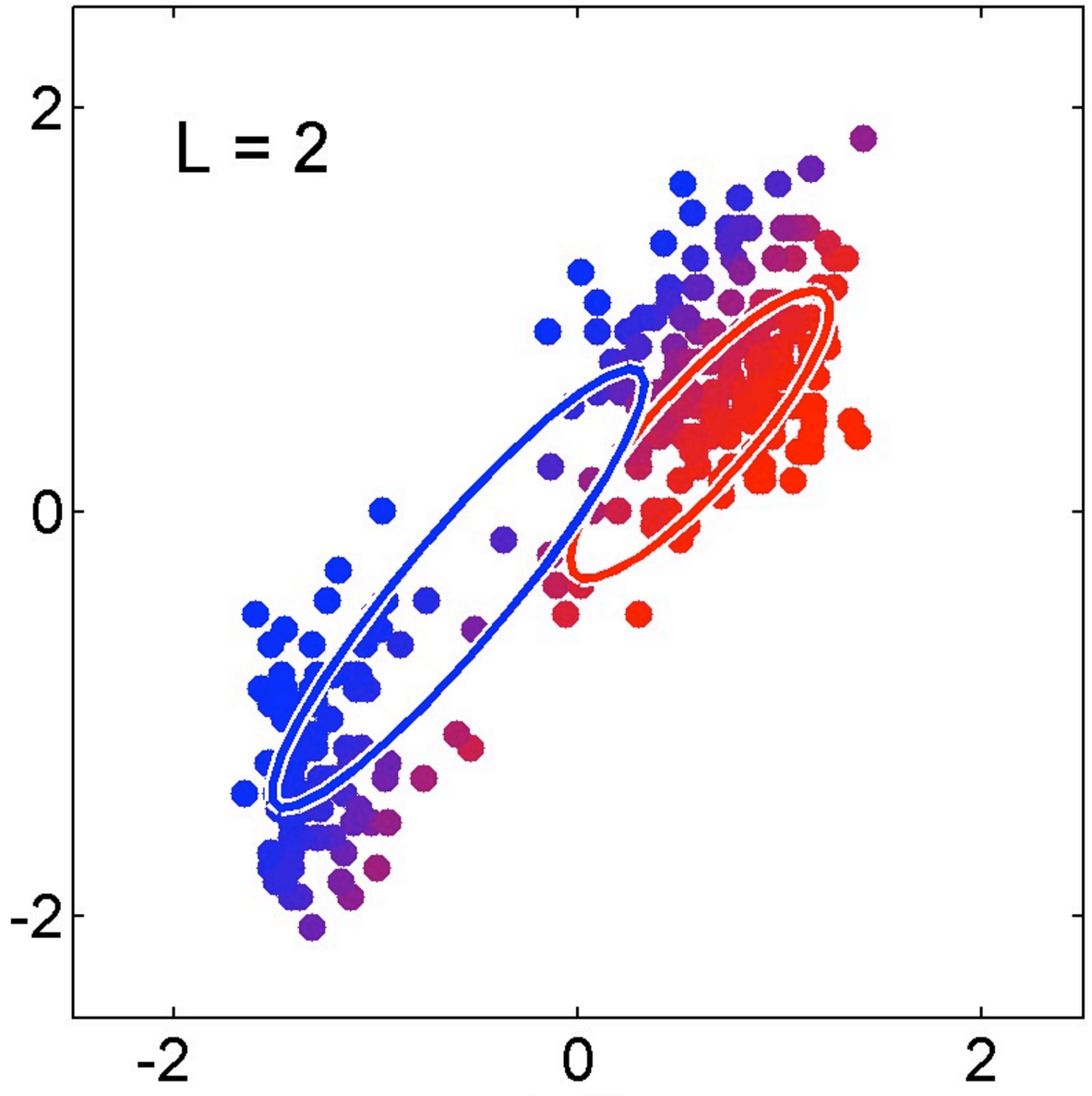
- Often both the E and the M step can be solved in closed form
- Neither the E step nor the M step can decrease the log-likelihood
- Algorithm is guaranteed to converge to a local maximum of the likelihood
- Must specify initialization and stopping criteria

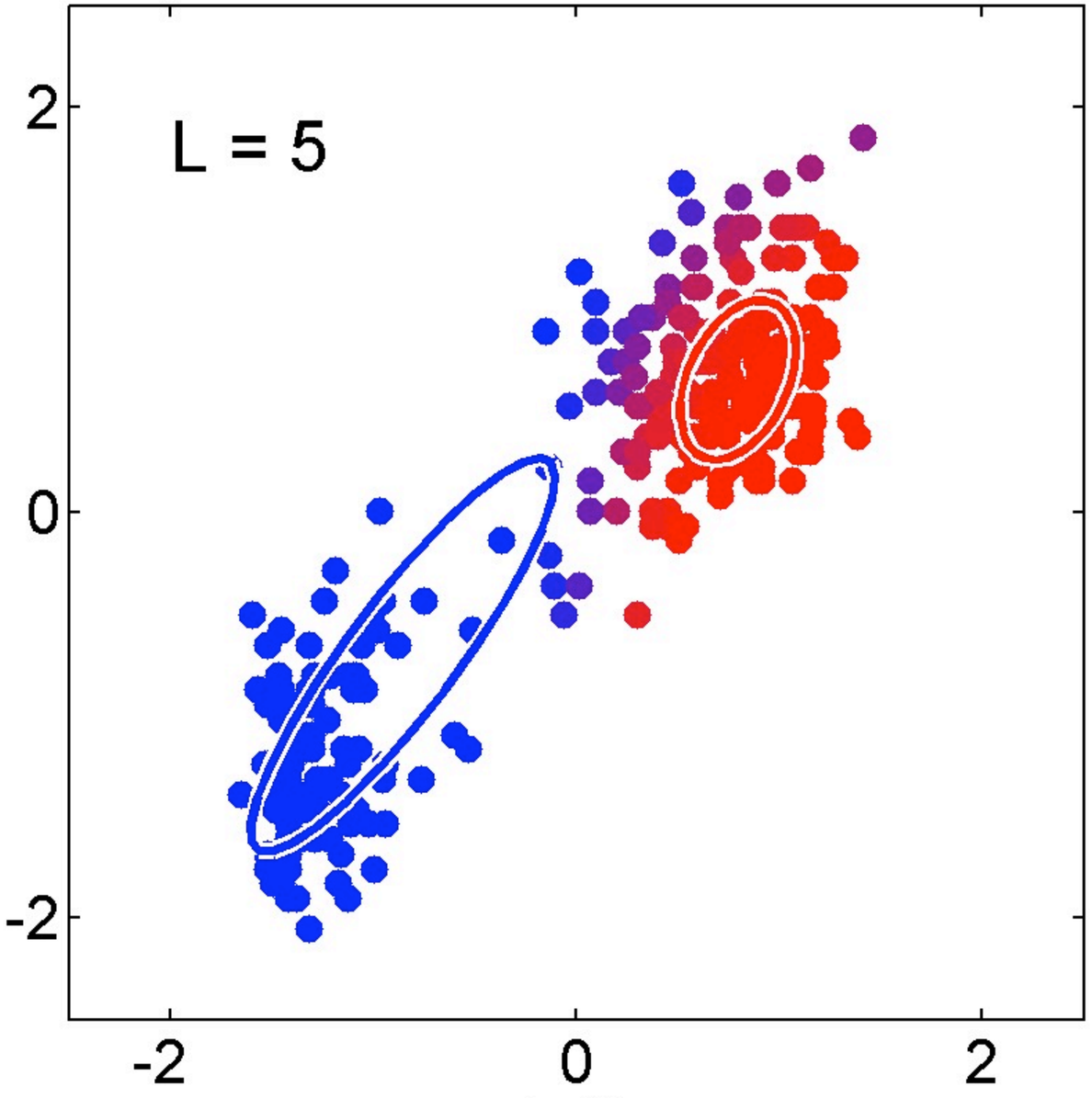
GMM EXAMPLE

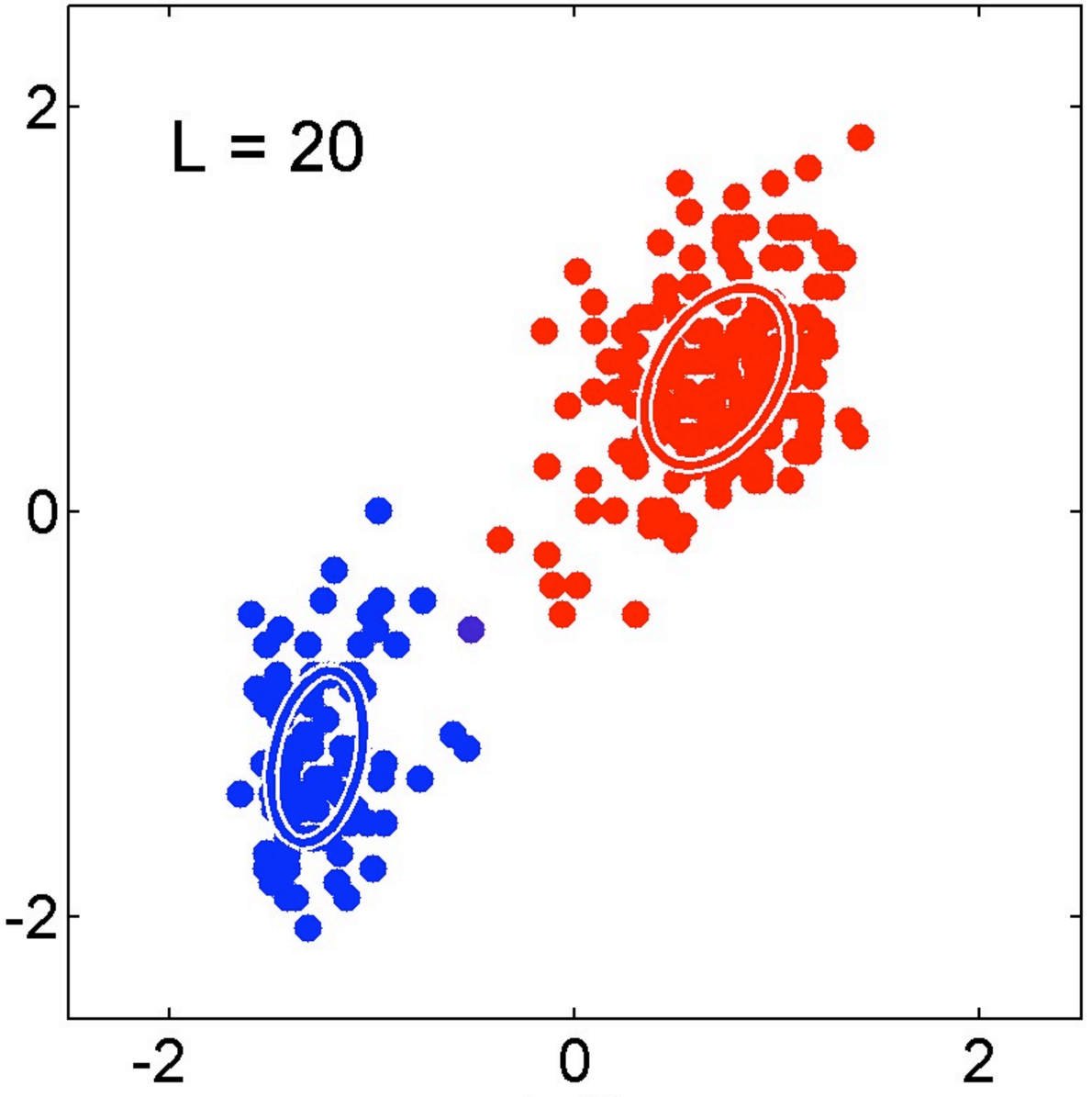

2
0
-2
-2
0
2


2
0
-2
-2
0
2


L = 1
2
0
-2
-2
0
2


L = 2
2
0
-2
-2
0
2


L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

PROBABILISTIC CLUSTERING

- Model provides full distributional description for each component
  - May be able to interpret differences in the distributions
- Soft clustering (compared to k-mean hard clustering)
  - Given the model, each point has a k-component vector of membership probabilities
- Key cost: assumption of parametric model

MIXTURE MODELS

- Knowledge representation?
  - **Parametric model**
    parameters = mixture coefficient and component parameters
- Score function?
  - **Likelihood**
- Search?
  - **Expectation maximization**
    iteratively find parameters that maximize likelihood and predicts cluster memberships
- Optimal? Exhaustive?