OntarioTech Science

Kourosh Davoudi
kourosh@uoit.ca

Advanced Cluster Analysis:
GMM
Spectral Clustering

CSCI 4150U: Data Mining

Outline

- GMM and The EM Algorithm
- Spectral Clustering

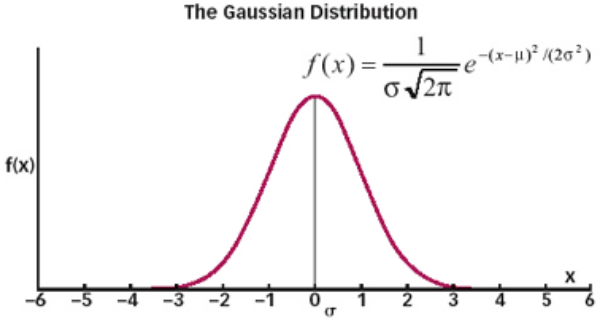
The Gaussian Distribution

Multivariate Gaussian

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

mean

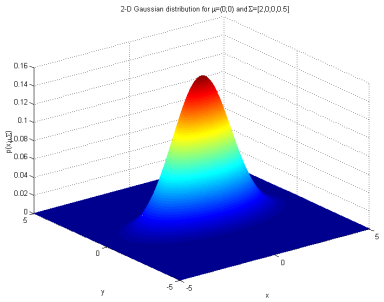
covariance



Maximum likelihood estimation

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(x_i-\hat{\mu})(x_i-\hat{\mu})^T$$

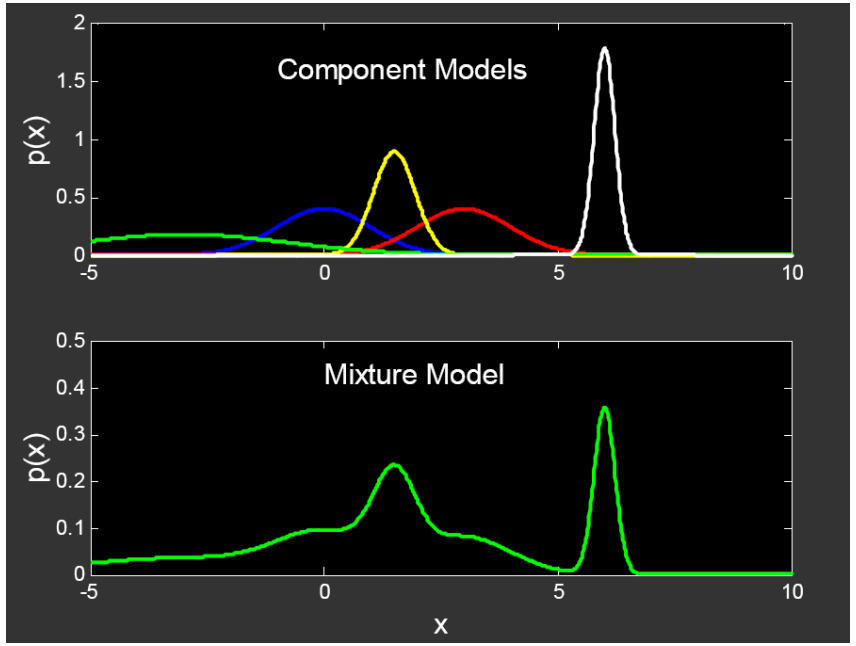


Gaussian Mixture

- Linear combination of Gaussians

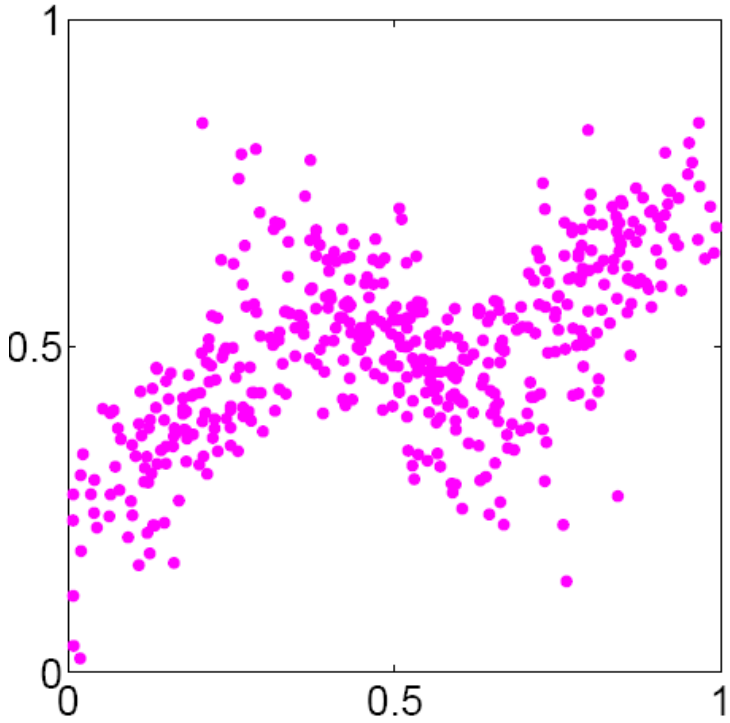
$$p(x) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x|\mu_k, \Sigma_k) \quad \text{where} \quad \sum_{k=1}^{K} \pi_k = 1, \quad 0 \leqslant \pi_k \leqslant 1$$

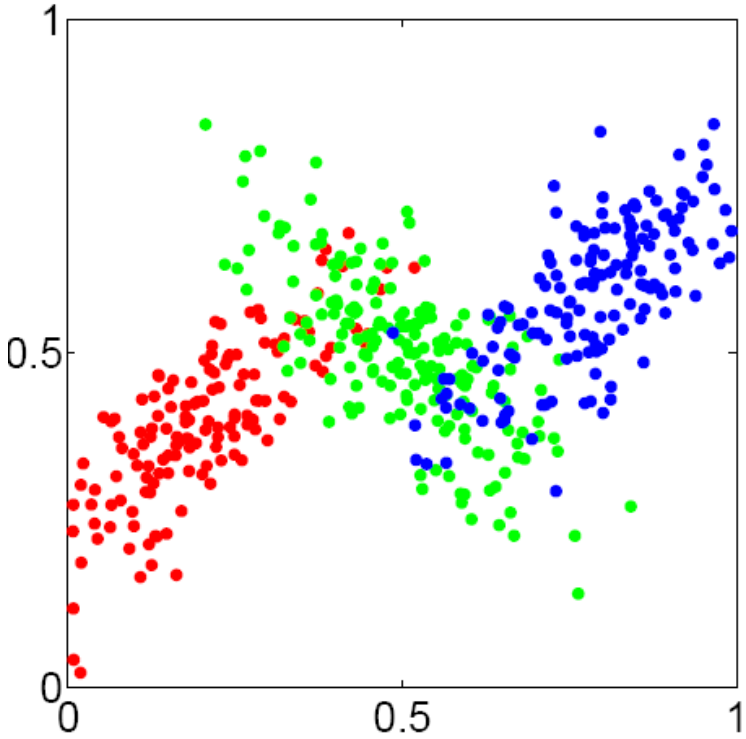
parameters to be estimated



Gaussian Mixture

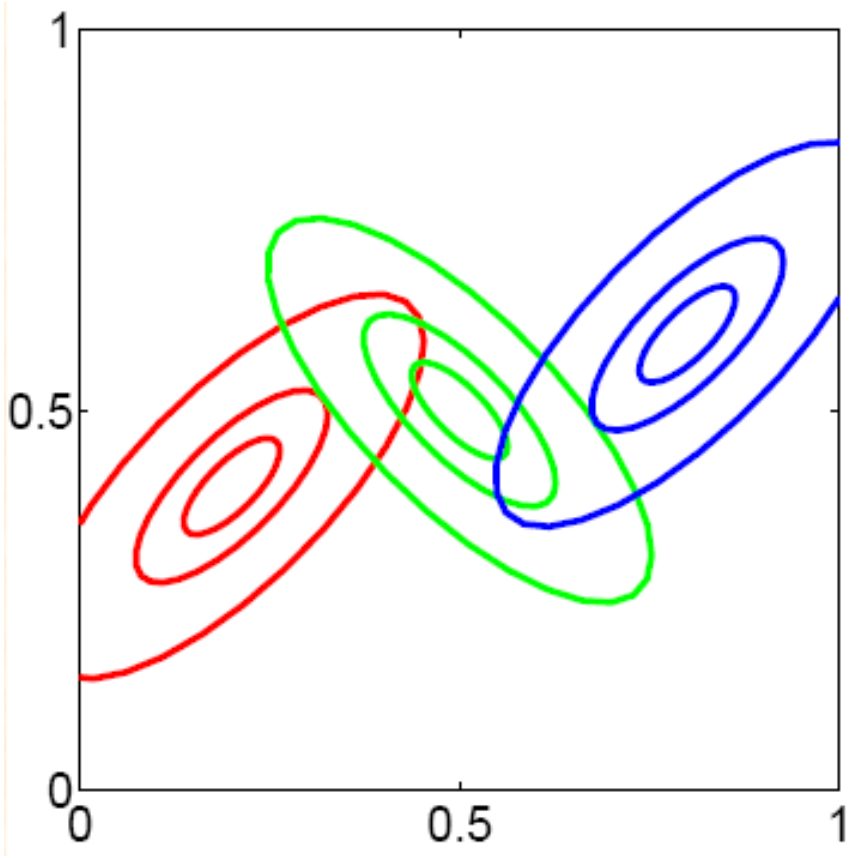
- Incomplete Data vs. Complete data





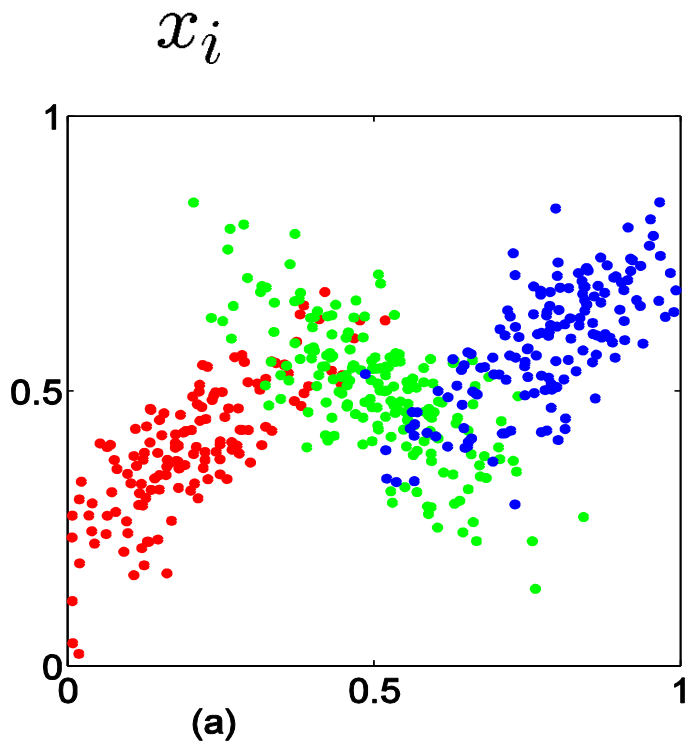
Gaussian Mixture

- Example: Mixture of 3 Gaussians



Gaussian Mixture

- To generate a data point:
  - first pick one of the components with probability π_k
  - then draw a sample x_i from that component distribution

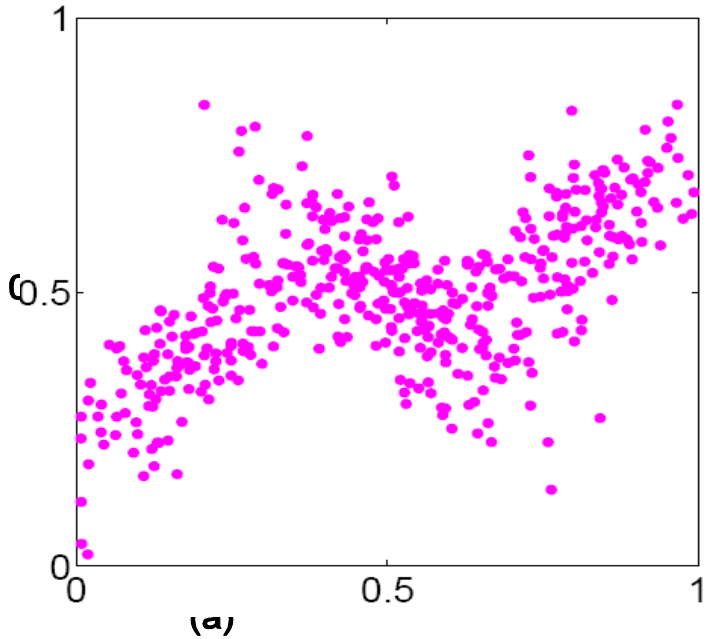


Gaussian Mixture

- To generate a data point:
  - first pick one of the components with probability π_k
  - then draw a sample x_i from that component distribution
- Each data point is generated by one of K components, a latent variable is associated with each x_i

$$z_i = (z_{i1}, \ldots, z_{iK})$$

$$\sum_{k=1}^{K} z_{ik} = 1 \text{ and } p(z_{ik} = 1) = \pi_k$$



Gaussian Mixture

- Loss function: the negative log likelihood of the data.
  - Equivalently, maximize the log likelihood.

$$\ln p(x|\pi, \mu, \Sigma) = \sum_{i=1}^{n} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k, \Sigma_k)\}$$

- Without knowing values of latent variables, we have to maximize the incomplete log likelihood.
  - Sum over components appears inside the logarithm, no closed-form solution.

Fitting the Gaussian Mixture

- Given the complete data set $(x, z) = (x_i, z_i)_{i=1,\ldots,n}$
  - Maximize the complete log likelihood.

$$\ln p(x, z|\pi, \mu, \Sigma) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \{\ln \pi_k + \ln \mathcal{N}(x_i|\mu_k, \Sigma_k)\}$$

- Need a procedure that would let us optimize the incomplete log likelihood by working with the (easier) complete log likelihood instead.

Expectation-Maximization (EM) Algorithm

- **E-step:** for given parameter values we can compute the expected values of the latent variables (responsibilities of data points)

Bayes rule

$$
\begin{aligned}
r_{ik} \equiv E(z_{ik}) &= p(z_{ik}=1|x_i,\pi,\mu,\Sigma) \\
&= \frac{p(z_{ik}=1)p(x_i|z_{ik}=1,\pi,\mu,\Sigma)}{\sum_{k=1}^{K} p(z_{ik}=1)p(x_i|z_{ik}=1,\pi,\mu,\Sigma)} \\
&= \frac{\pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_i|\mu_k,\Sigma_k)}
\end{aligned}
$$

- Note that $r_{ik} \in [0,1]$ instead of $\{0,1\}$ but we still have

$$\sum_{k=1}^{K} r_{ik} = 1 \text{ for all } i$$

The EM Algorithm

- **M-step:** maximize the expected complete log likelihood

$$E[\ln p(x, z|\pi, \mu, \Sigma)] = \sum_{i=1}^{n} \sum_{k=1}^{K} r_{ik} \{\ln \pi_k + \ln \mathcal{N}(x_i|\mu_k, \Sigma_k)\}$$

  - Parameter update:

$$\pi_k = \frac{\sum_i r_{ik}}{n} \qquad \mu_k = \frac{\sum_i r_{ik} x_i}{\sum_i r_{ik}}$$

$$\Sigma_k = \frac{\sum_i r_{ik}(x_i - \mu_k)(x_i - \mu_k)^T}{\sum_i r_{ik}}$$

The EM Algorithm

- Iterate E-step and M-step until the log likelihood of data does not increase any more.
  - Converge to local optima.
  - Need to restart algorithm with different initial guess of parameters (as in K-means).

- Relation to K-means
  - Consider GMM $\Sigma_k = \delta^2 I$ with common covariance.
  - As $\delta^2 \to 0, r_{ik} \to 0 \text{ or } 1$, two methods coincide.

EM Algorithm Summary

Given the data set: $x_1, x_2, \ldots, x_n$, and number of cluster K

Initialize π_k $\mu_k \Sigma_k$ randomly ($k = 1, 2, \ldots K$)

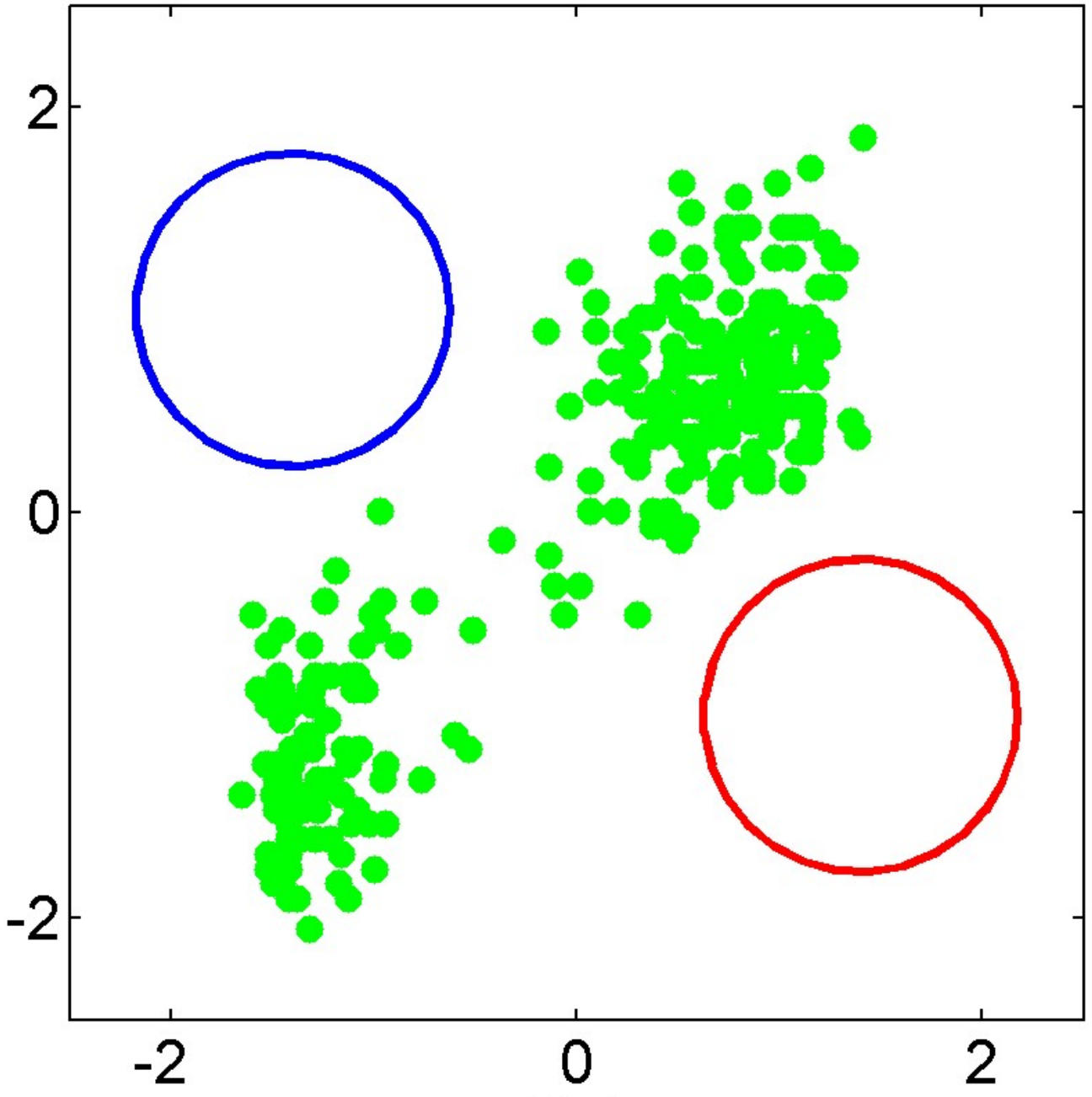
Loop (until convergence)

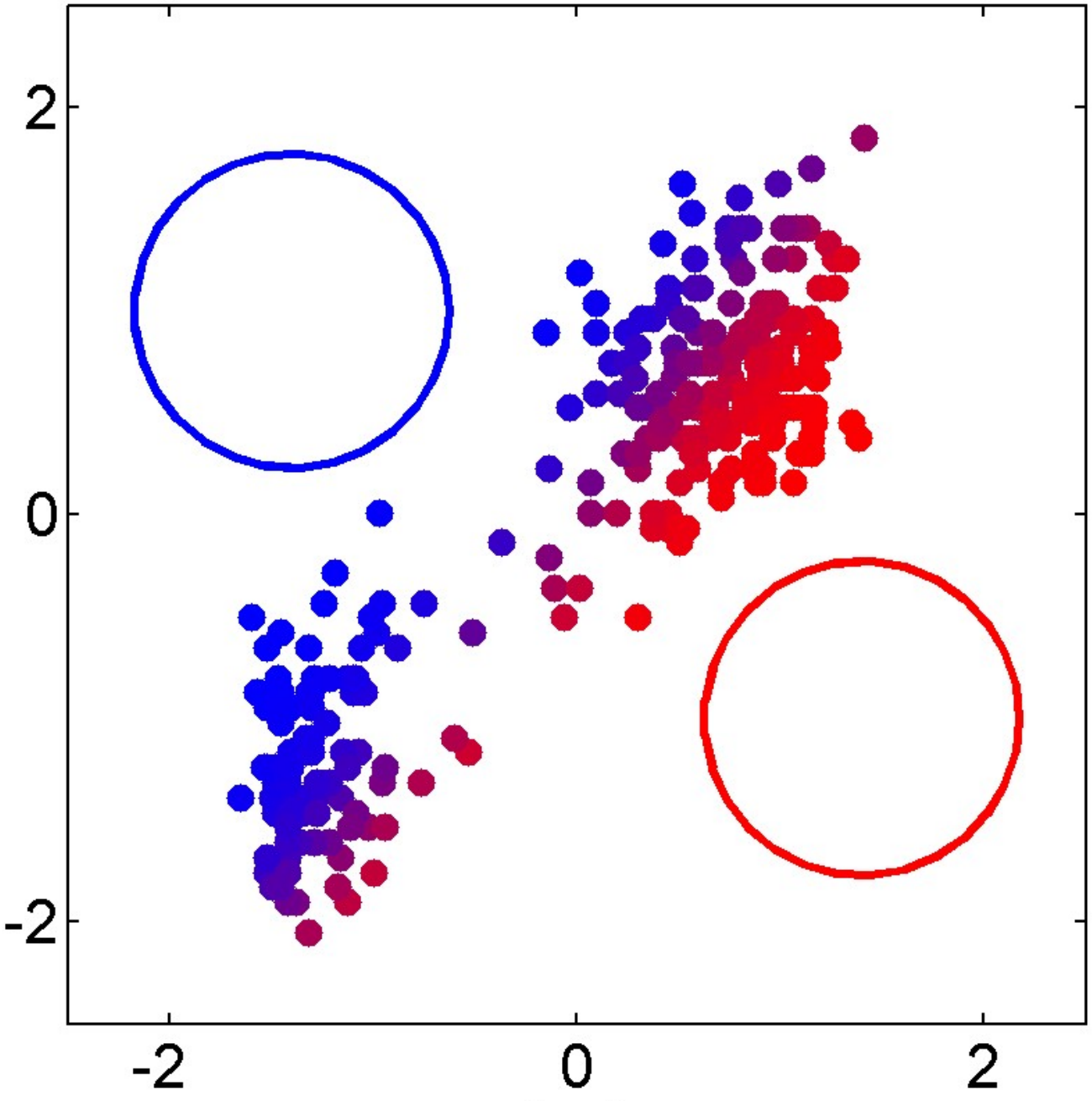
E-step: computer Expectation (r_{ik})

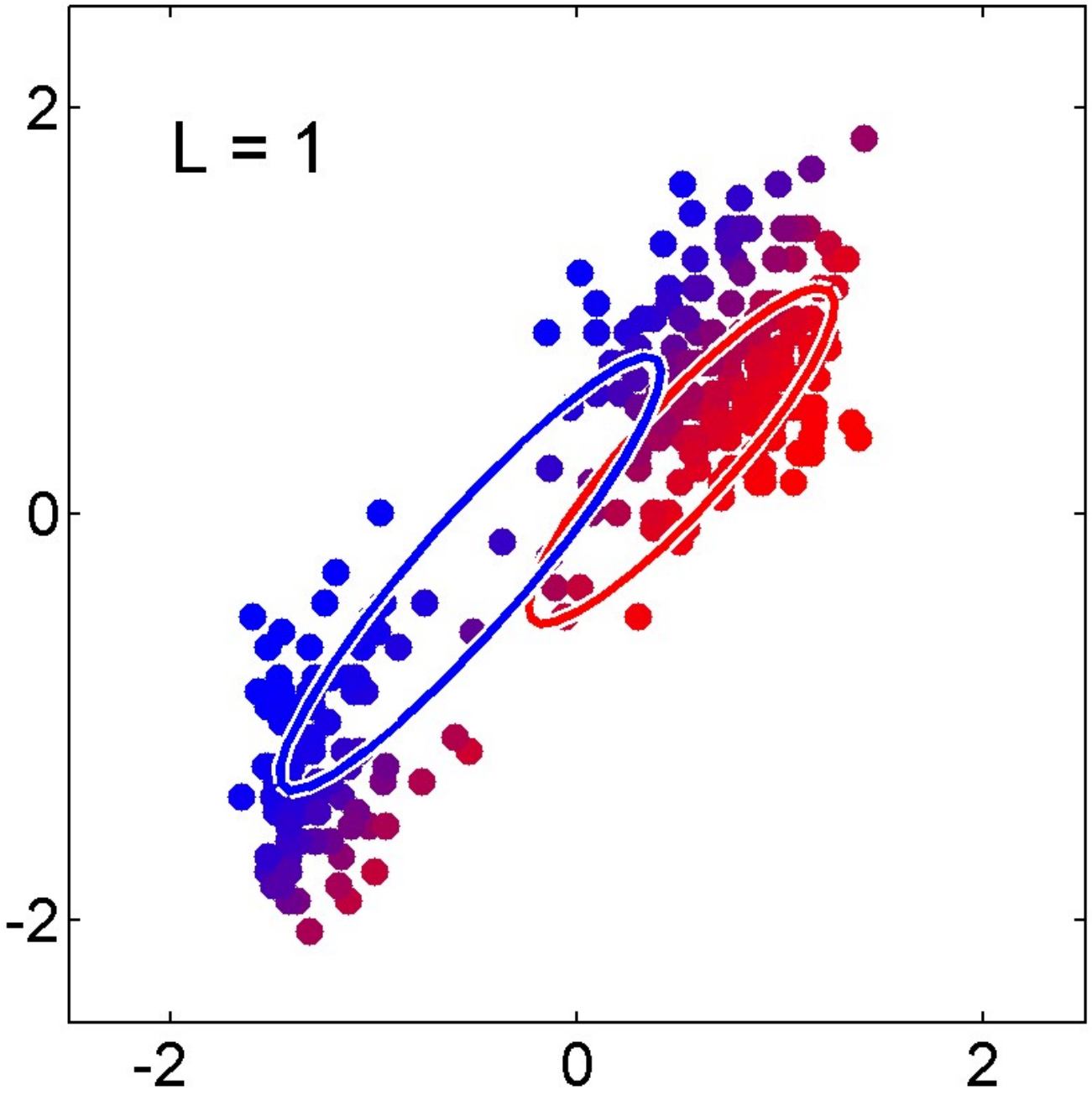
M-step: petameter update (π_k $\mu_k \Sigma_k$)

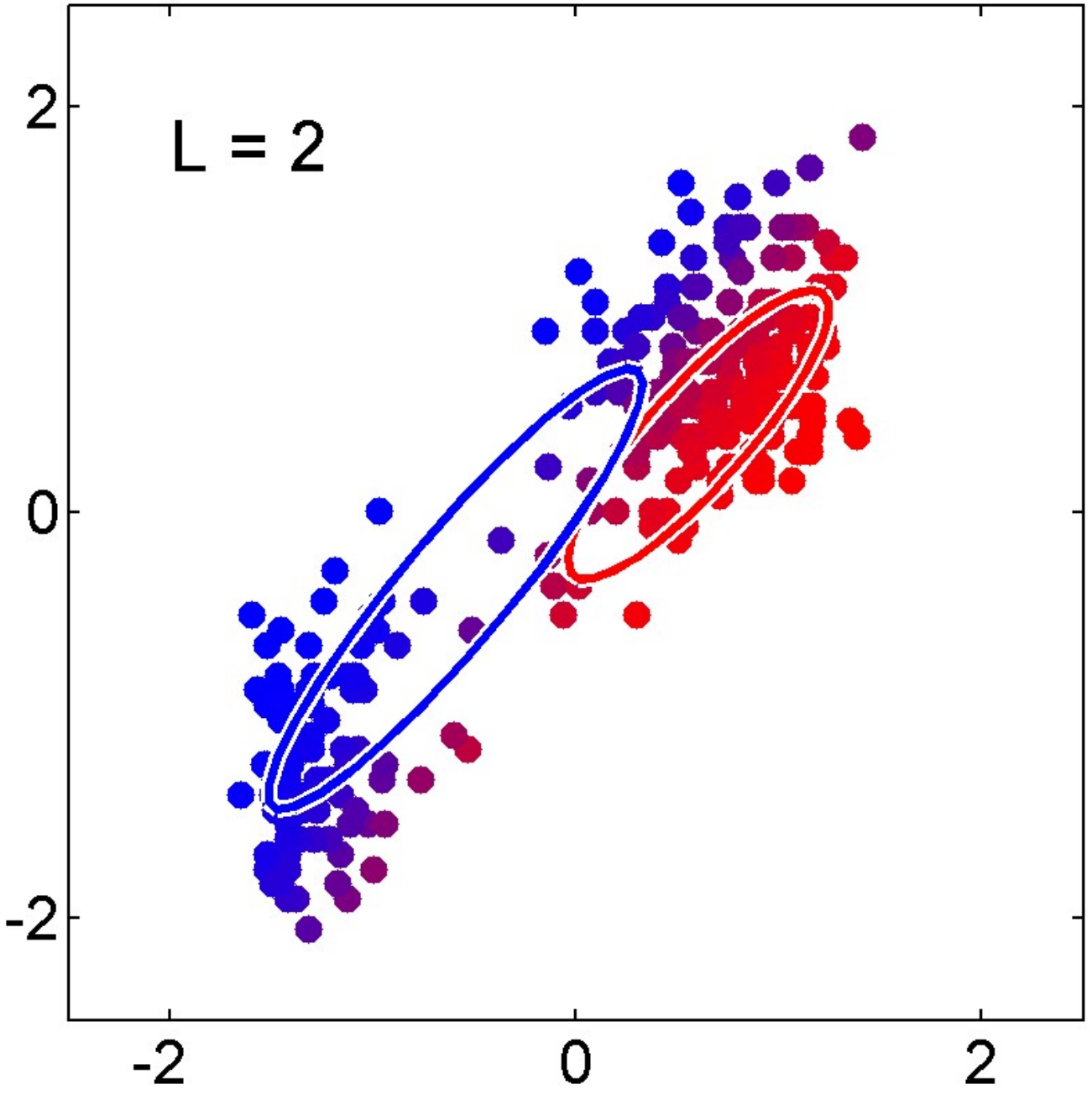
End

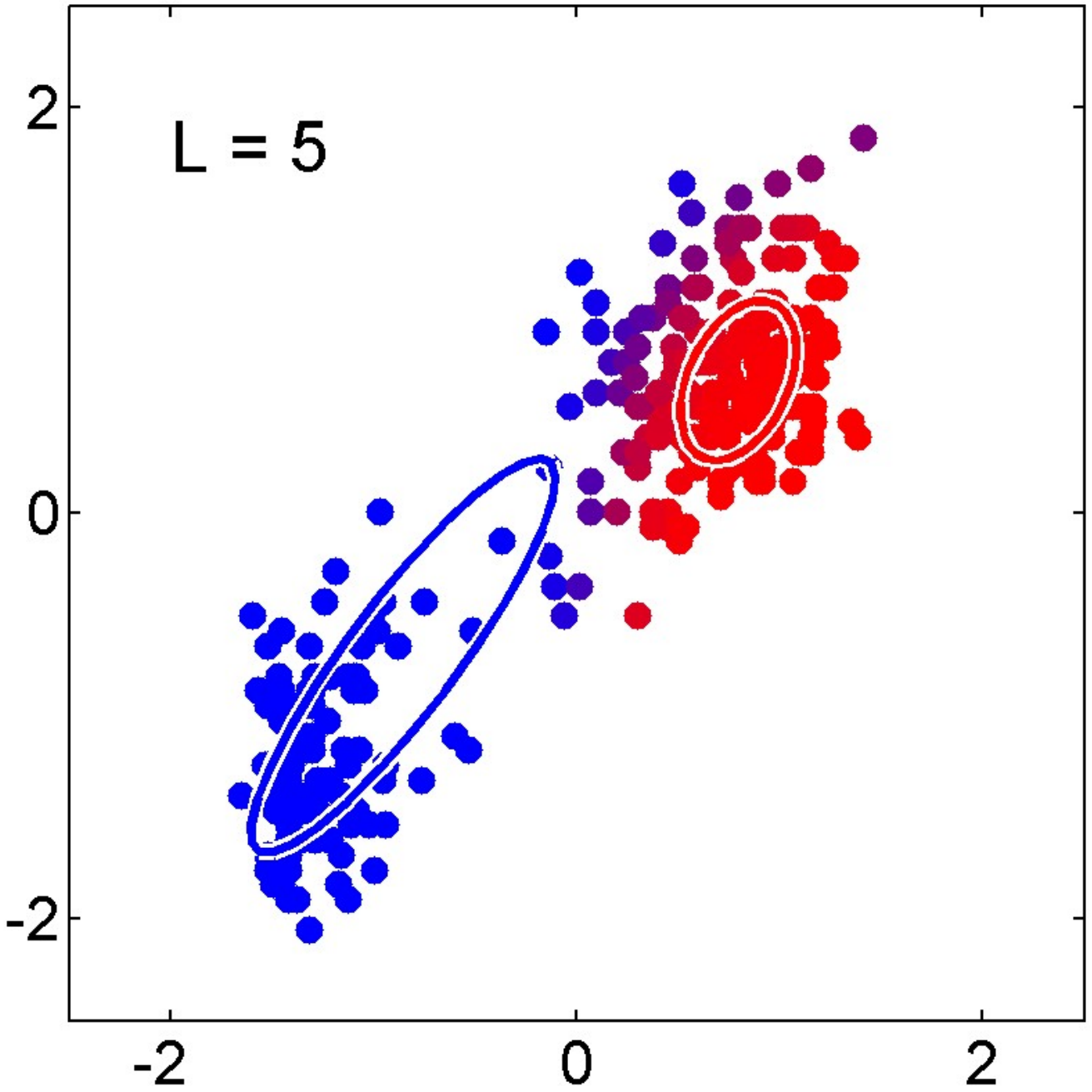
https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

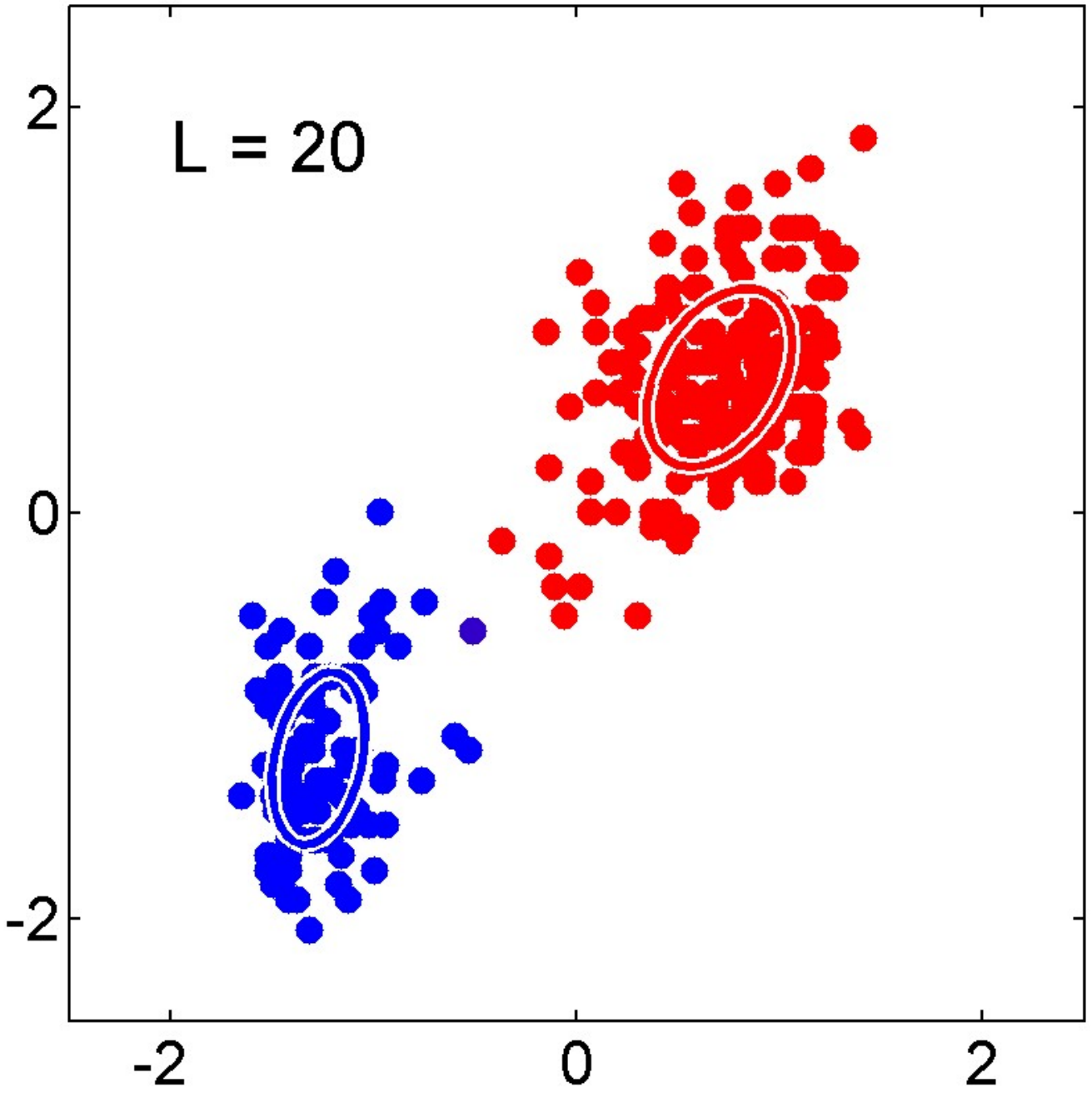

2
0
-2
-2
0
2


2
0
-2
-2
0
2


L = 1
2
0
-2
-2
0
2

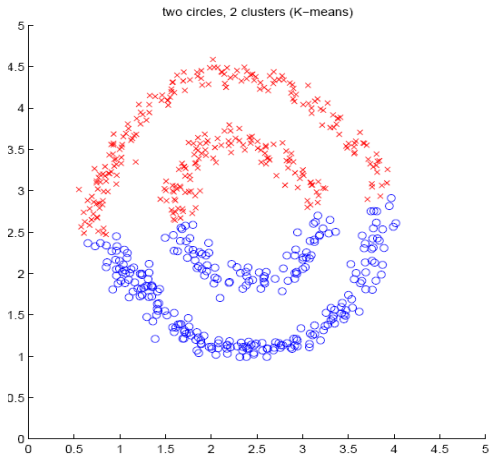

L = 2
2
0
-2
-2
0
2

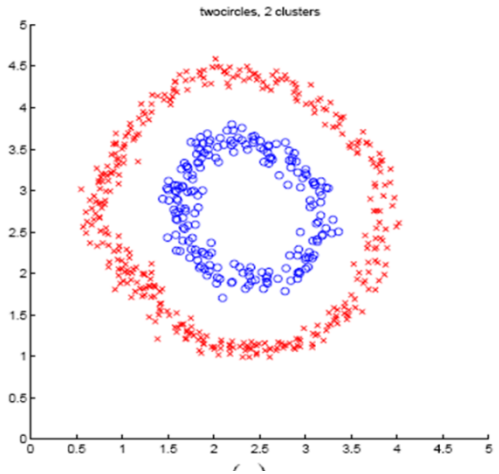

L = 5
2
0
-2
-2
0
2


L = 20
2
0
-2
-2
0
2

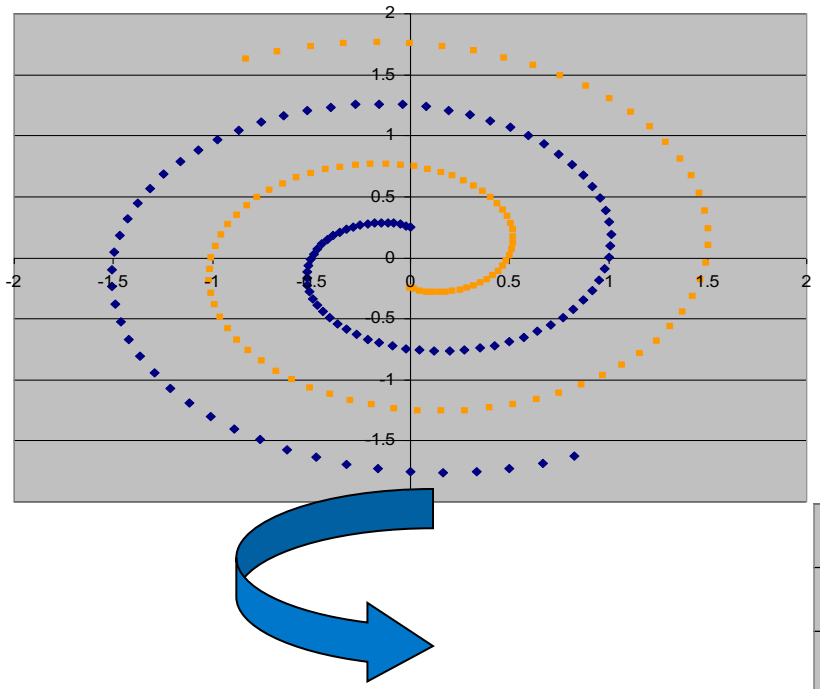
Spectral Clustering

- Another well-known graph based clustering
- Algorithms that cluster points using eigenvectors of **similarity matrices** derived from the data
- Obtain data representation embedded in **the low-dimensional** space that can be easily clustered
- Can handle non-convex clusters





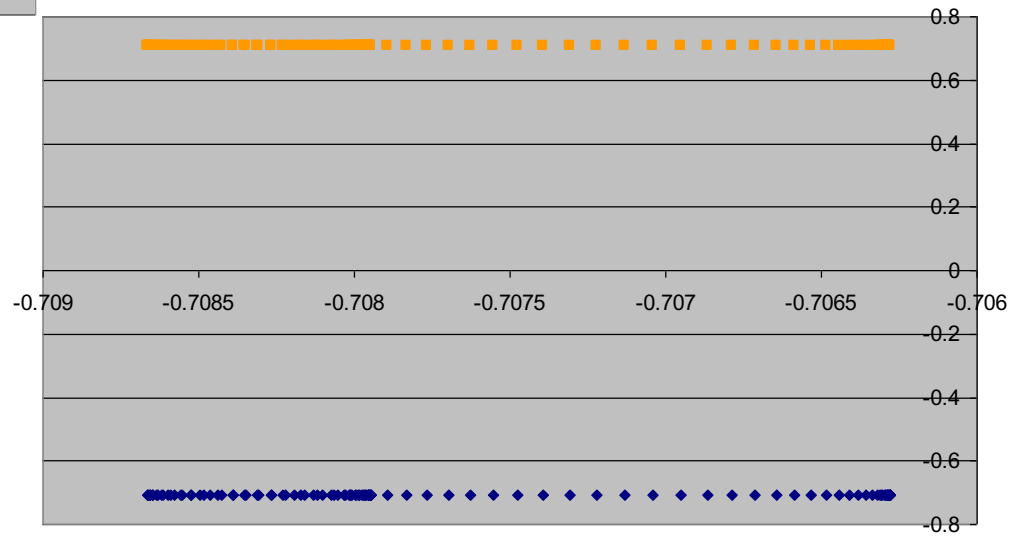
Spectral Clustering Example: 2-Spirals



Dataset exhibits complex cluster shapes

⇒ K-means performs very poorly in this space due bias toward dense spherical clusters.

In the embedded space given by two leading eigenvectors, clusters are trivial to separate.



Spectral Clustering Algorithm (NIPS'02 Ng et al.)

- Motivation
  - Given a set of n points:

$$S = \{s_1, \ldots, s_n\} \in R^l$$

  - We would like to cluster them into k subsets

Algorithm

- Form the affinity matrix $A \in R^{nxn}$
- Define $A_{ij} = e^{-\|s_i - s_j\|^2 / 2\sigma^2}$ if $i \neq j$

  $A_{ii} = 0$

  - Scaling parameter σ^2 chosen by user

Algorithm

- Form the normalized matrix

$$L = D^{-1/2} A D^{-1/2}$$

- Define D a diagonal matrix whose (i,i) element is the sum of A's row i

- Find $x_1, x_2, \ldots, x_k$, the k largest eigenvectors of L

- These form the columns of the new matrix X
  - Note: have reduced dimension from n$\times$n to n $\times$ k

Algorithm

- Form the matrix $Y \in R^{nxk}$
  - Normalize each of X's rows to have unit length

$$Y_{ij} = X_{ij} / (\sum_{j} X_{ij}^{2})^{0.5}$$

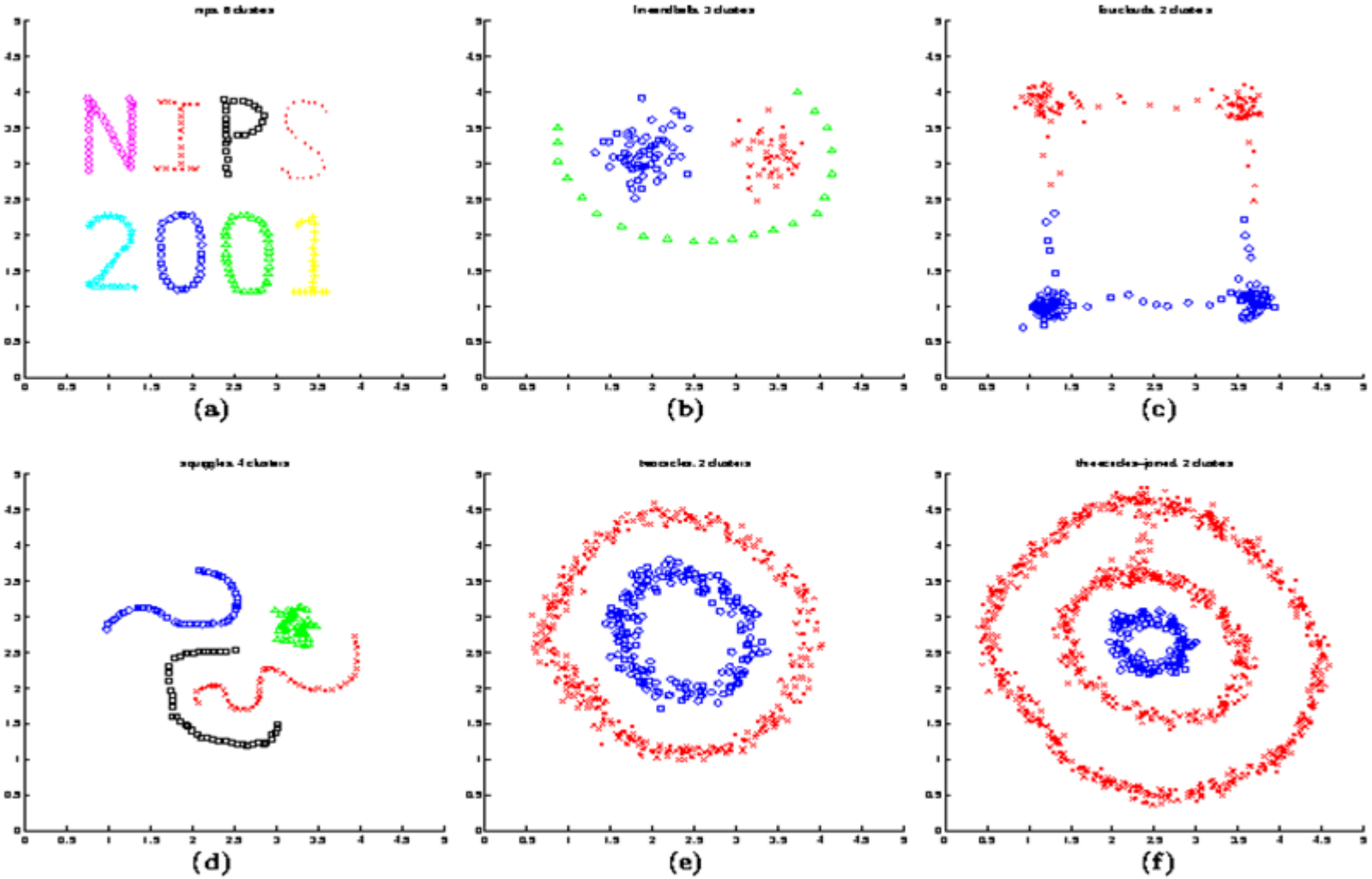
- Treat each row of Y as a point in R^{k}
- Cluster Y into k clusters via K-means

- Final Cluster Assignment
  - Assign point $\boldsymbol{S}_i$ to cluster j if row i of Y was assigned to cluster j

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

Why?

- If we eventually use K-means, why not just apply K-means to the original data?
- It makes use of the spectrum of the similarity matrix of the data to perform dimensionality reduction for clustering in the fewer/lower dimensional space.
- This method allows us to cluster non-convex regions


(a)
(b)
(c)
(d)
(e)
(f)

Pros and Cons of Spectral Clustering

- Pros
  - Simple and empirically proven successful method for clustering
  - Can effectively discover non-convex patterns
  - Enjoy solid theoretical foundation

- Cons
  - Choosing a similarity matrix/graph can be non-trivial and may require extensive preprocessing.
  - Require to select the scaling factor carefully
  - Finding automatically the number of groups