Introduction to Machine Learning

CS 307 --- Fall 2022

The Expectation-Maximization Algorithm

Reading:
Section 20.3, R&N
Section 6.6, Mitchell

Semi-Supervised Learning

Problem setting:

We have a small amount of labeled data and a large amount of unlabeled data → Learning solely from the labeled data will unlikely result in an accurate classifier

Goal: use the unlabeled data in the learning process in an attempt to acquire a more accurate classifier

Idea: use EM

treat the labels of unlabeled instances as hidden data

The Polvia/Gromland Example

- Three variables:
  - Attribute 1: Hair = {blond, dark}
  - Attribute 2: Height = {tall, short}
  - Class: Country = {Gromland, Polvia}

Training Data:

(B, T, G)
(D, T, G)
(B, S, P)
(D, S, P)
(D, S, P)
(B, S, G)
(D, S, G)
(D, T, P)
(B, S, ?)
(D, T, ?)
(D, S, ?)

Using EM for Semi-Supervised Learning

Need to

- design a generative model
  - what are the model parameters?
- identify the sufficient statistics
- design the E-step
  - compute the expected values of the sufficient statistics
- design the M-step
  - find parameter values that maximize the expected complete log likelihood

Using Naïve Bayes

Can start with either the E-step or the M-step

Makes more sense to start with the M-step:

initialize the model parameters by training a NB classifier on only the labeled instances

Then iterate

E-step: probabilistically label the unlabeled instances

M-step: retrain the classifier on both labeled & unlabeled data

The Likelihood Function

What is the likelihood function we are trying to maximize?

The log likelihood function is

$$\left.\begin{aligned}\log L(Data \mid \theta) &= \sum_{x_i \in D} \log(P(y_i \mid \theta)P(x_i \mid y_i;\theta)) \\ &\quad + \sum_{x_i \in U} \log \sum_{j=1}^{|C|} P(c_j \mid \theta)P(x_i \mid c_j;\theta)\end{aligned}\right\} \begin{array}{c}\text{Incomplete}\\ \text{Log-Likelihood}\end{array}$$

where D is our labeled data and U is our unlabeled data.

Unfortunately, maximization by partial derivatives with a log of sums is computationally intractable.

The Likelihood Function (Cont')

Nevertheless, if we had access to the complete data, we could rewrite the log likelihood function as

$$\log L(Data \mid \theta) = \sum_{x_i \in Data} \sum_{j=1}^{|C|} z_{ij} \log(P(y_i = c_j \mid \theta) P(x_i \mid y_i = c_j; \theta)) \Bigg\} \text{ Complete Log-Likelihood}$$

where $z_{ij} \in \{0, 1\}$.

Of course, we don't have access to the complete data in practice. As a result, EM computes the expected values of z_{ij} in the E-step, and maximizes the expected complete log likelihood in the M-step.

Can we start at the E-step?

Yes. Randomly assign class labels to the unlabeled instances (so there is no hidden info anymore)

Then iterate:

M-step: retrain a classifier on both labeled & unlabeled data

E-step: probabilistically (re)label the unlabeled instances

Role of Labeled Data

Does that mean we don’t need any labeled data?

What is the implication for learning with generative vs. discriminative models?

Using EM in Practice

EM may not work well in practice …

Potential problems

- get stuck at a local maximum

Solutions

- select a number of different starting points
- search by simulated annealing
- combine EM with active learning

Three-Coin Example

We observe a series of coin tosses generated in the following way:

A person has three coins.

- Coin 0: probability of Head is λ
- Coin 1: probability of Head p
- Coin 2: probability of Head q

Two Estimation Problems

Scenario I: Toss coin 0. If Head – toss coin 1; o/w -- toss coin 2

Observing the sequence HHHHT, THTHT, HHHHT, HHTTH

produced by Coin 0 , Coin1 and Coin2

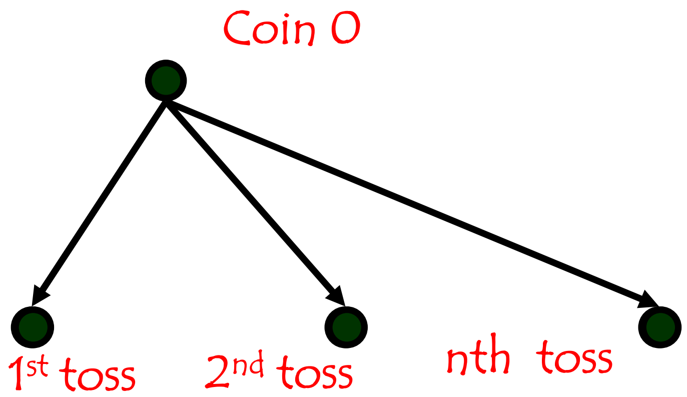
Estimate most likely values for p, q (the probability of H in each coin) and the probability to use each of the coins (λ)

Scenario II: Toss coin 0. If Head – toss coin 1; o/w -- toss coin 2

Observing the sequence HHHT, HTHT, HHHT, HTTH

produced by Coin 1 and/or Coin 2

Estimate most likely values for p, q and λ



Key Intuition

Scenario I: we knew which of the data points (HHHT), (HTHT), (HTTH) came from Coin1 and which from Coin2, so there was no problem.

Scenario II: we don’t know whether a given data point came from Coin 1/2 (hidden data), so we can use EM

1) Start at the M-step: guess the model parameters

2) Iterate

E-step: Compute the likelihood of the data given this model.

M-step: Re-estimate the model parameters to maximize the likelihood of the data.

This process can be iterated and can be shown to converge to a local maximum of the likelihood function

E-Step

We will assume (for a minute) that we know the parameters $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\alpha}$ and use it to estimate which Coin it is

Then, we will use the estimation for the tossed Coin, to estimate the most likely parameters and so on...

What is the probability that the ith data point came from Coin1 ?

$$\mathbf{P}_1^i = \mathbf{P}(\mathbf{Coin1} \mid \mathbf{D}^i) = \frac{\mathbf{P}(\mathbf{D}^i \mid \mathbf{Coin1})\,\mathbf{P}(\mathbf{Coin1})}{\mathbf{P}(\mathbf{D}^i)} =$$

$$= \frac{\tilde{\alpha}\,\tilde{\mathbf{p}}^{\mathbf{h}}(\mathbf{1}-\tilde{\mathbf{p}})^{\mathbf{m-h}}}{\tilde{\alpha}\,\tilde{\mathbf{p}}^{\mathbf{h}}(\mathbf{1}-\tilde{\mathbf{p}})^{\mathbf{m-h}} + (\mathbf{1}-\alpha\tilde{\ })\tilde{\mathbf{q}}^{\mathbf{h}}(\mathbf{1}-\tilde{\mathbf{q}})^{\mathbf{m-h}}}$$

M-Step

At this point we would like to compute the likelihood of the data, and find the parameters that maximize it.

We will maximize the incomplete log likelihood of the data (n data points):

$$\mathbf{LL} = \sum_{i=1}^{n} \mathbf{log\,P(D^i\,|\,p,q,}\alpha)$$

But, one of the variables – the coin's name - is hidden.

Which value do we plug in for it in order to compute the likelihood of the data?

We think of the likelihood $\log(D^i|p,q,\alpha)$ as a random variable that depends on the value of the coin in the i^{th} toss. Therefore, instead of maximizing the incomplete LL we will maximize the expectation of this random variable (over the coin's name) --- the expected complete log likelihood

Maximizing the Expected Complete LL

We maximize the expectation of this random variable (over the coin name).

$$\mathbf{E[LL]} = \mathbf{E}[\sum_{i=1}^{n} \log \mathbf{P}(\mathbf{D}^{\mathbf{i}} \mid \mathbf{p},\mathbf{q},\alpha)] = \sum_{i=1}^{n} \mathbf{E}[\log \mathbf{P}(\mathbf{D}^{\mathbf{i}} \mid \mathbf{p},\mathbf{q},\alpha)] =$$

$$= \sum_{1}^{n} \mathbf{P}_{\mathbf{1}}^{\mathbf{i}} \log \mathbf{P}(\mathbf{1},\mathbf{D}^{\mathbf{i}} \mid \mathbf{p},\mathbf{q},\alpha) + (\mathbf{1} - \mathbf{P}_{\mathbf{1}}^{\mathbf{i}}) \log \mathbf{P}(\mathbf{0},\mathbf{D}^{\mathbf{i}} \mid \mathbf{p},\mathbf{q},\alpha)$$

Maximizing the Expected Complete LL

Explicitly, we get:

$$E(\sum_i \log P(D^i \mid \tilde{p},\tilde{q},\tilde{\alpha}) =$$

$$= \sum_i P_1^i \log P(1,D^i \mid \tilde{p},\tilde{q},\tilde{\alpha}) + (1-P_1^i)\log P(0,D^i \mid \tilde{p},\tilde{q},\tilde{\alpha}) =$$

$$= \sum_i P_1^i \log(\tilde{\alpha}\,\tilde{p}^{h_i}(1-\tilde{p})^{m-h_i}) + (1-P_1^i)\log((1-\tilde{\alpha})\,\tilde{q}^{h_i}(1-\tilde{q})^{m-h_i}) =$$

$$= \sum_i P_1^i(\log\tilde{\alpha} + h_i \log\tilde{p} + (m-h_i)\log(1-\tilde{p})) +$$

$$(1-P_1^i)(\log(1-\tilde{\alpha}) + h_i \log\tilde{q} + (m-h_i)\log(1-\tilde{q}))$$

M-Step: Re-Estimating Model Parameters

Finally, to find the most likely parameters, we maximize the derivatives with respect to $\tilde{p}, \tilde{q}, \tilde{\alpha}$:

$$\frac{dE}{d\tilde{\alpha}} = \sum_{i=1}^{n} \frac{P_1^i}{\tilde{\alpha}} - \frac{1 - P_1^i}{1 - \tilde{\alpha}} = 0 \qquad \Rightarrow \qquad \tilde{\alpha} = \frac{\sum P_1^i}{n}$$

$$\frac{dE}{d\tilde{p}} = \sum_{i=1}^{n} P_1^i \left(\frac{h_i}{\tilde{p}} - \frac{m - h_i}{1 - \tilde{p}}\right) = 0 \qquad \Rightarrow \qquad \tilde{p} = \frac{\sum P_1^i \frac{h_i}{m}}{\sum P_1^i}$$

$$\frac{dE}{d\tilde{q}} = \sum_{i=1}^{n} (1 - P_1^i)\left(\frac{h_i}{\tilde{p}} - \frac{m - h_i}{1 - \tilde{p}}\right) = 0 \qquad \Rightarrow \qquad \tilde{q} = \frac{\sum (1 - P_1^i) \frac{h_i}{m}}{\sum (1 - P_1^i)}$$

EM as a Soft/Probabilistic Clustering Algorithm

K-means is a **special case** of EM

Goal of K-means: represent a data set in terms of K clusters each of which is summarized by a prototype μ_k

Initialize prototypes, then iterate between two phases:

E step: assign each data point to nearest prototype

M step: update prototypes to be the cluster means

EM as a Soft/Probabilistic Clustering Algorithm

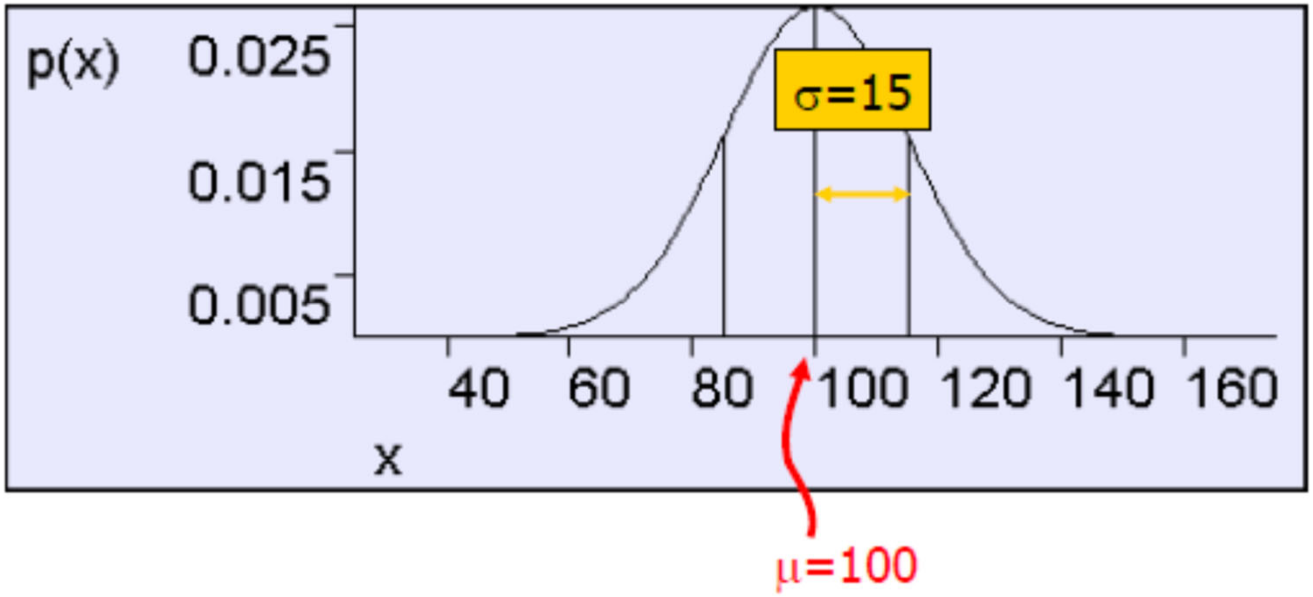
Represent the probability distribution of the data as a **mixture model**

- captures uncertainty in cluster assignments
- gives model for data distribution

Consider **mixtures** of Gaussians

The Gaussian Distribution

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



$$E[X] = \mu$$

$$\mathrm{Var}[X] = \sigma^2$$

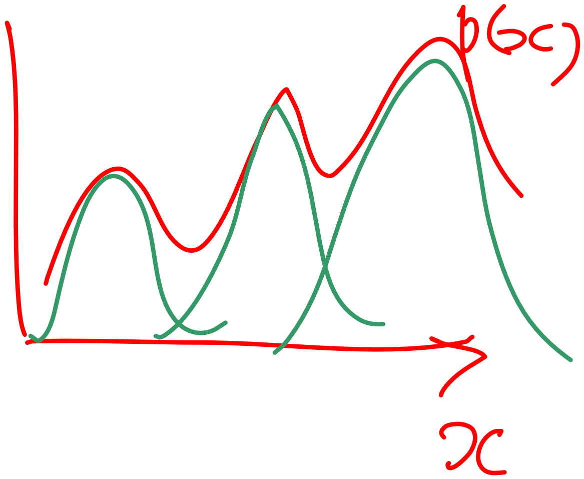
Shorthand: We say X ~ N(μ,σ^2) to mean "X is distributed as a Gaussian with parameters μ and σ^2".

In the above figure, X ~ N(100,15^2)

Gaussian Mixtures

Linear super-position of Gaussians

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



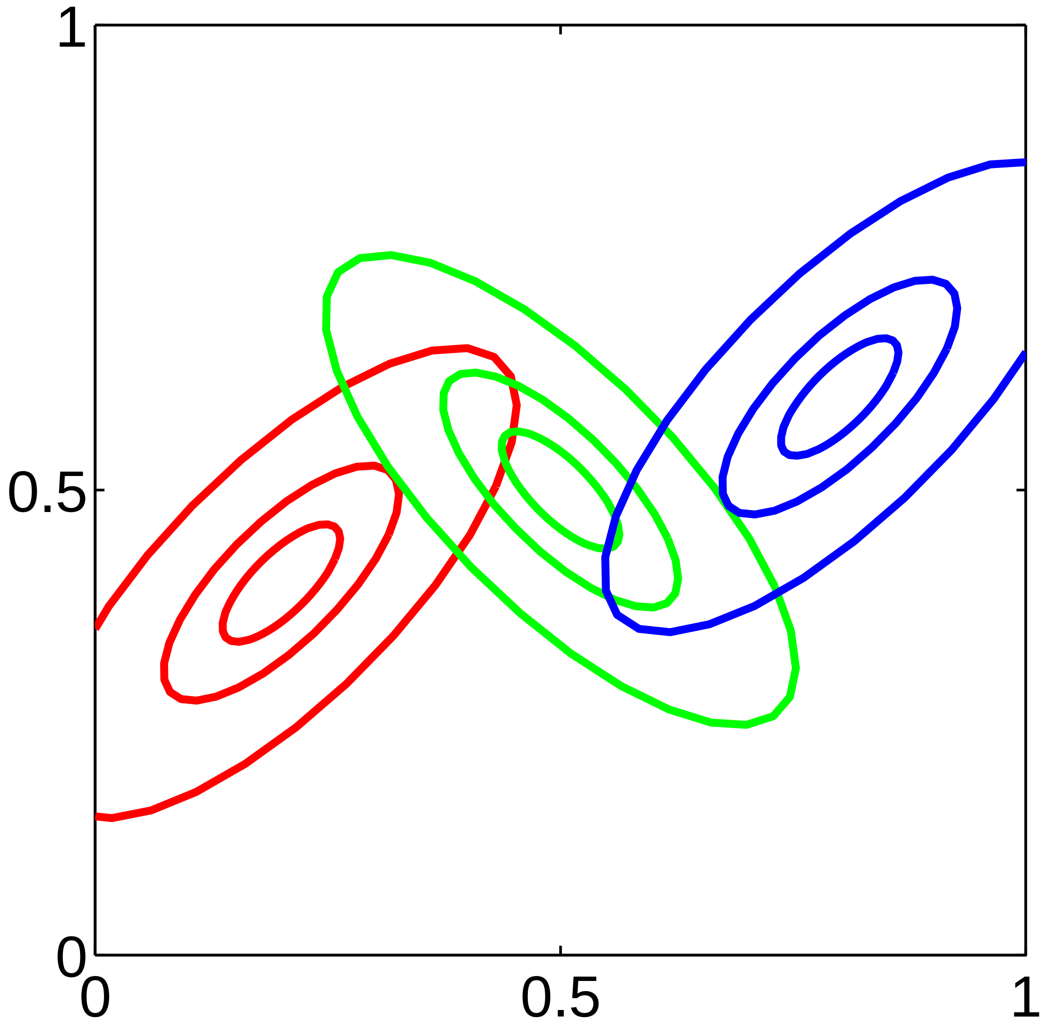
Normalization and positivity require

$$\sum_{k=1}^{K} \pi_k = 1 \qquad 0 \leqslant \pi_k \leqslant 1$$

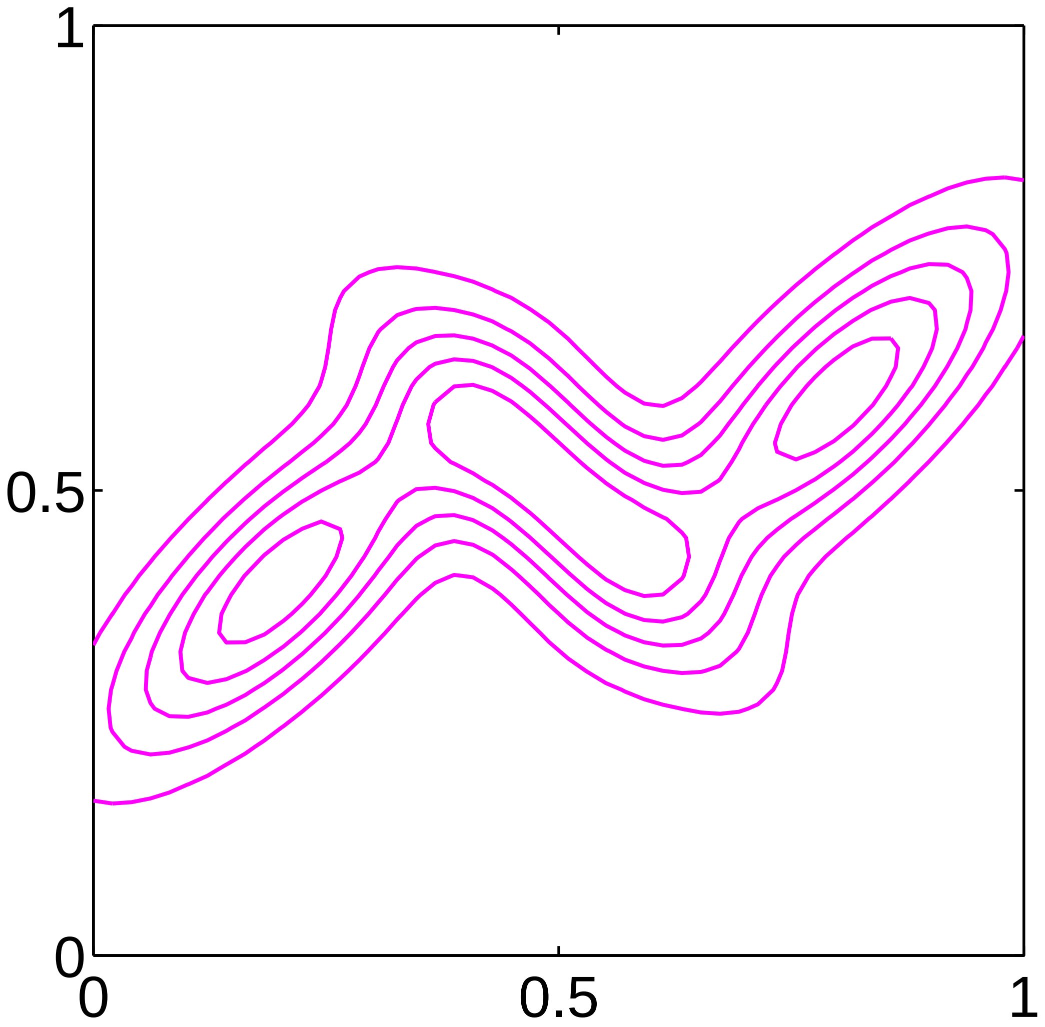
Can interpret the mixing coefficients as prior probabilities

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

Example: Mixture of 3 Gaussians



Contours of Probability Distribution



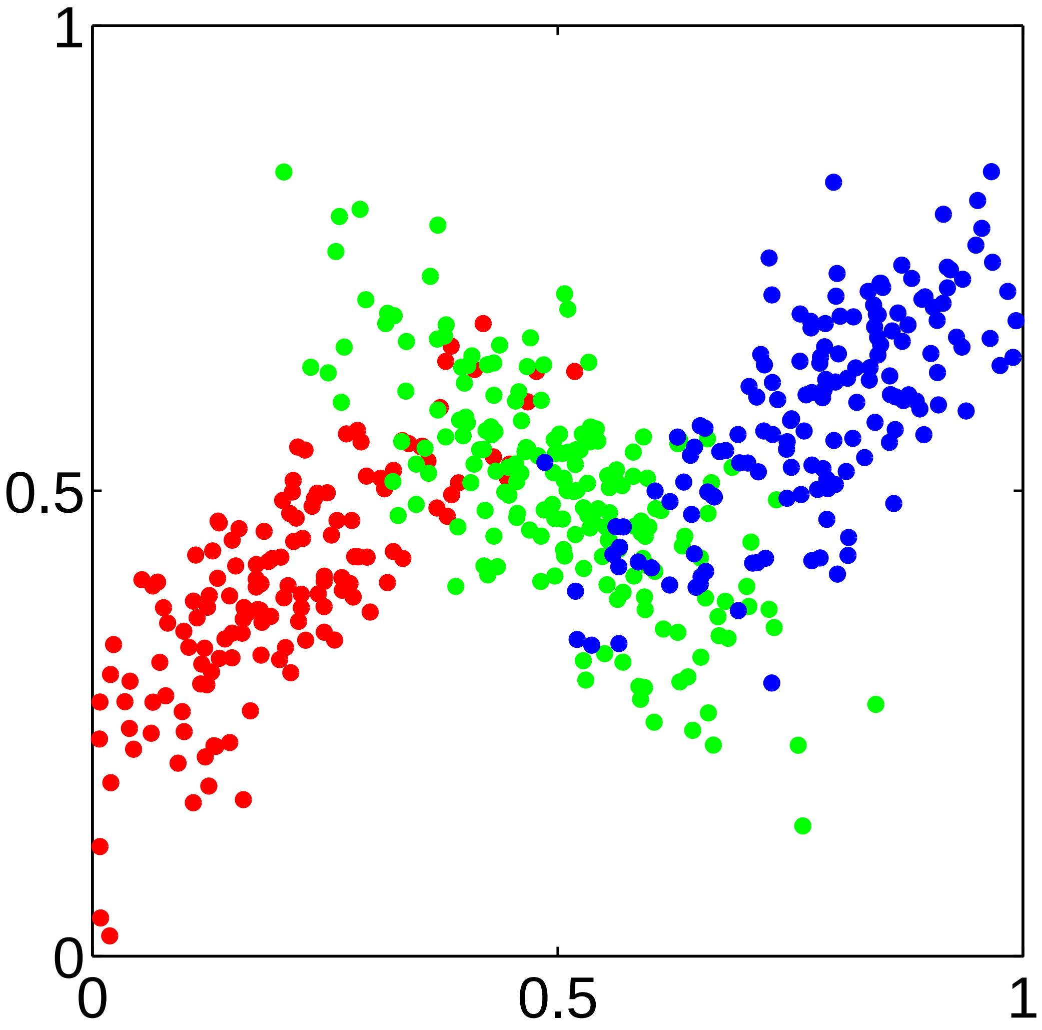
Sampling from the Gaussian

To generate a data point:

- first pick one of the components with probability π_k
- then draw a sample $\mathbf{x}_n$ from that component

Repeat these two steps for each new data point

Synthetic Data Set



Fitting the Gaussian Mixture

We wish to invert this process — given the data set, find the corresponding parameters:

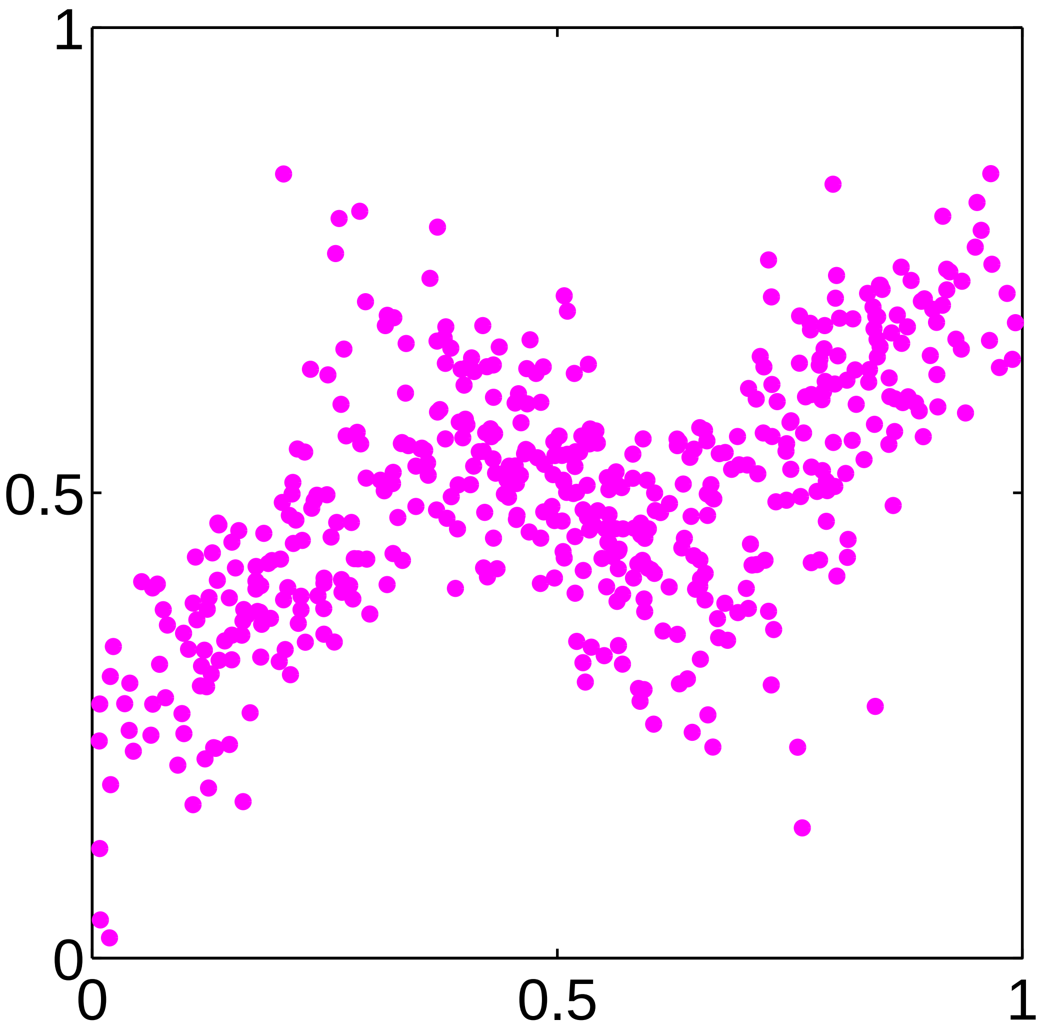
- mixing coefficients
- means
- covariances

If we knew which component generated each data point, the maximum likelihood solution would involve fitting each component to the corresponding cluster

Problem: the data set is unlabeled

We shall refer to the labels as *latent* (= hidden) variables

Synthetic Data Set Without Labels

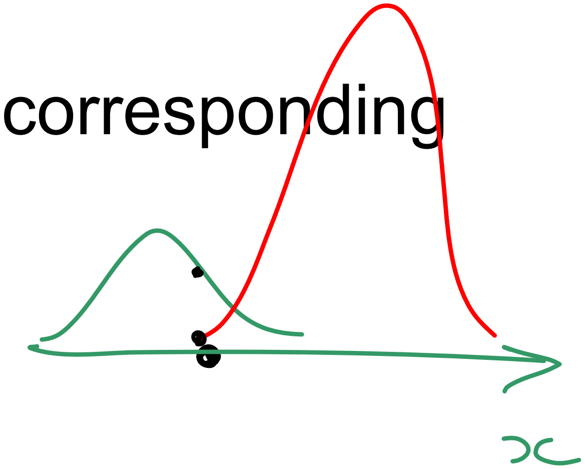


E-Step: Computing the Posterior Probabilities

We can think of the mixing coefficients as prior probabilities for the components

For a given value of $\mathbf{x}$ we can evaluate the corresponding posterior probabilities.

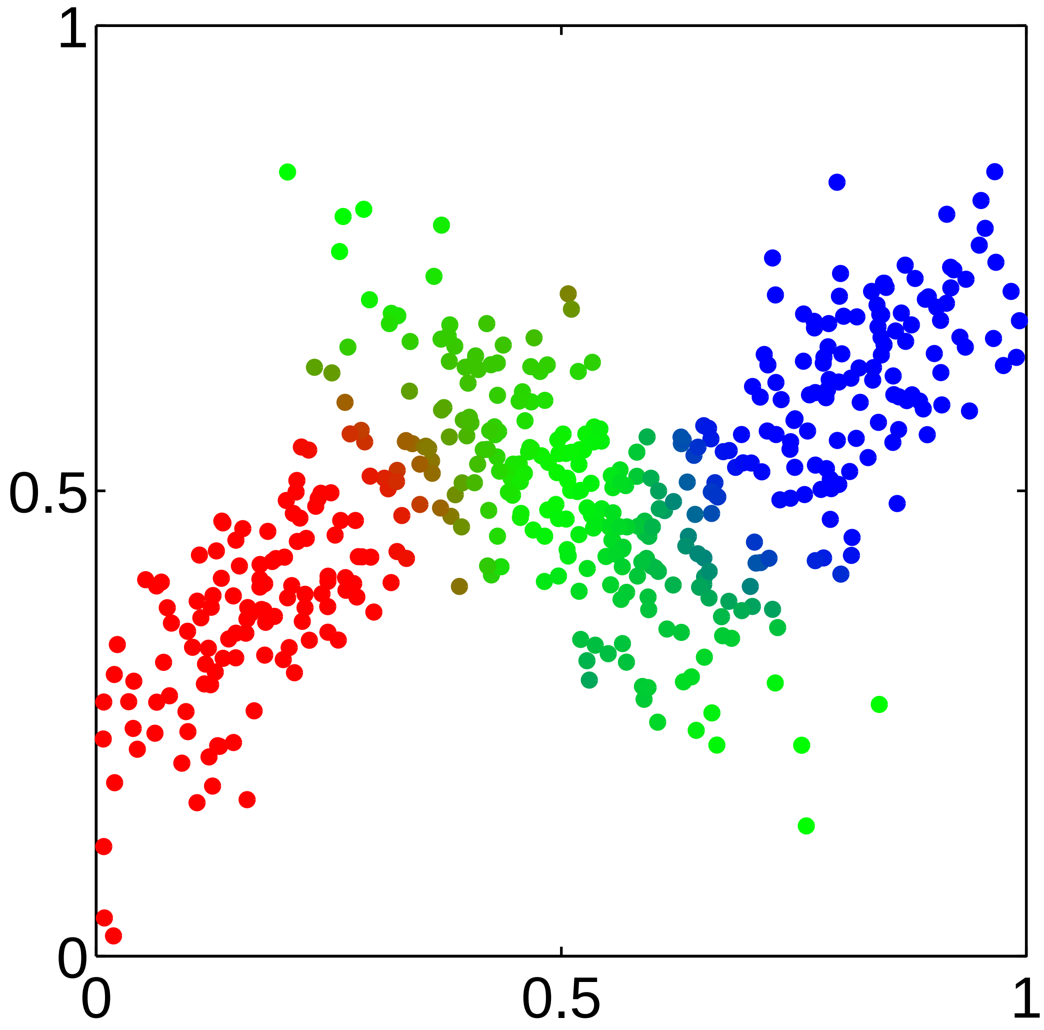
These follow from Bayes rule:



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

$$= \frac{\pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

Posterior Probabilities (color coded)



Maximum Likelihood for the GMM

The log likelihood function takes the form

$$\log p(D|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{i=1}^{N} \log \left\{ \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Note: sum over components appears *inside* the log

There is no closed form solution for maximum likelihood

- Solved by EM
- Fix K

M-Step: Maximizing the Log Likelihood

The incomplete log likelihood:

$$\log P(D \mid \mu, \pi, \Sigma) = \sum_{i=1}^{N} \log\{\sum_{k=1}^{K} \pi_k N_{ik}\} \quad N_{ik} = N_i(\mu_k, \Sigma_k)$$

Let us proceed by differentiating the expected complete log likelihood

M-Step: Maximizing the Log Likelihood

For μ_j:

$$\sum_{i=1}^{N} \underbrace{\frac{\pi_j N_{ij}}{\sum_k \pi_k N_{ik}}}_{\gamma_j(x_i)} \Sigma_j^{-1}(x_i - u_j) = 0$$

$$\underline{\mu} = \frac{\sum_i \gamma_{ji} x_i}{\sum_i \gamma_{ji}}$$

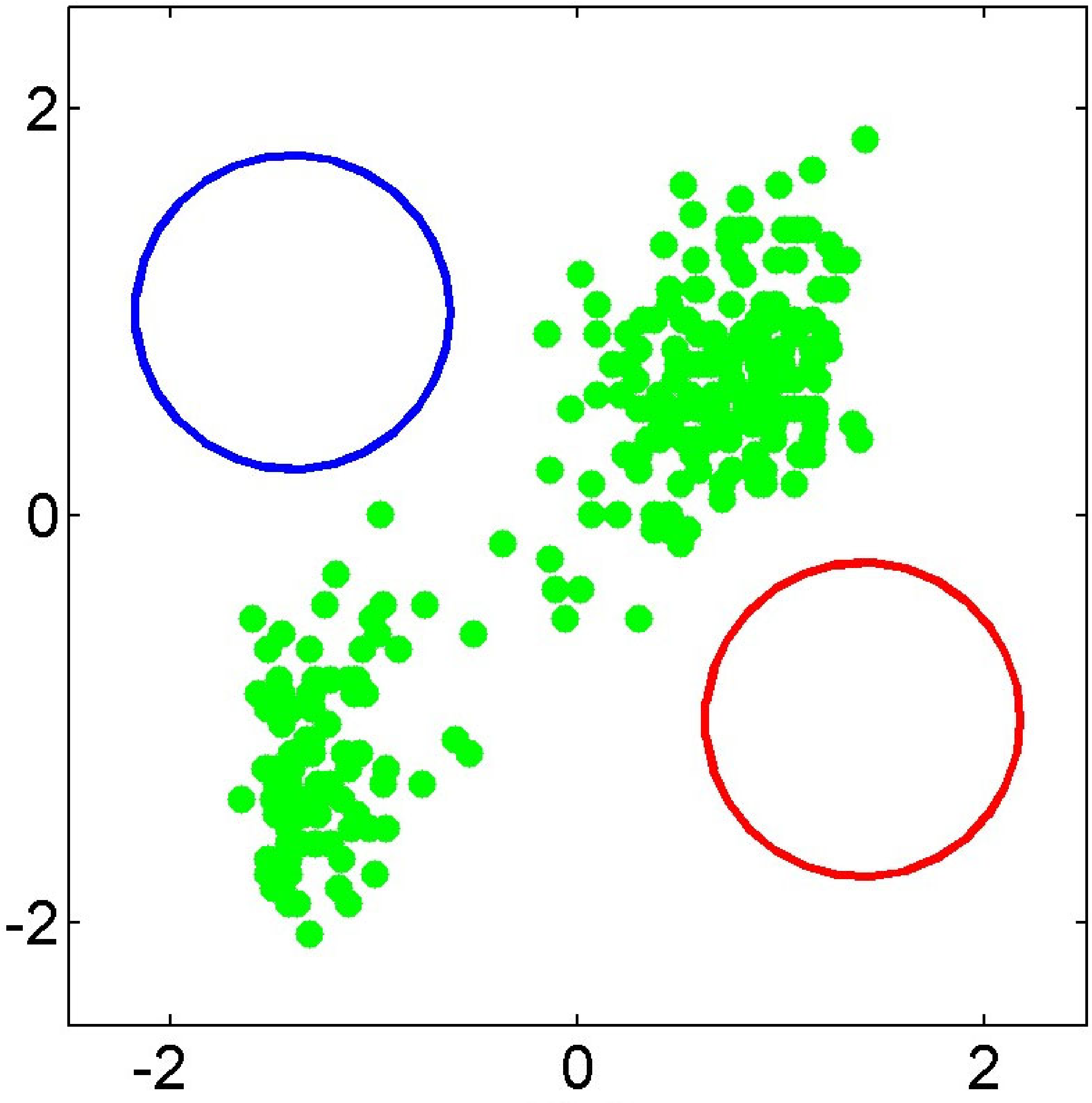
EM Algorithm – Informal Derivation

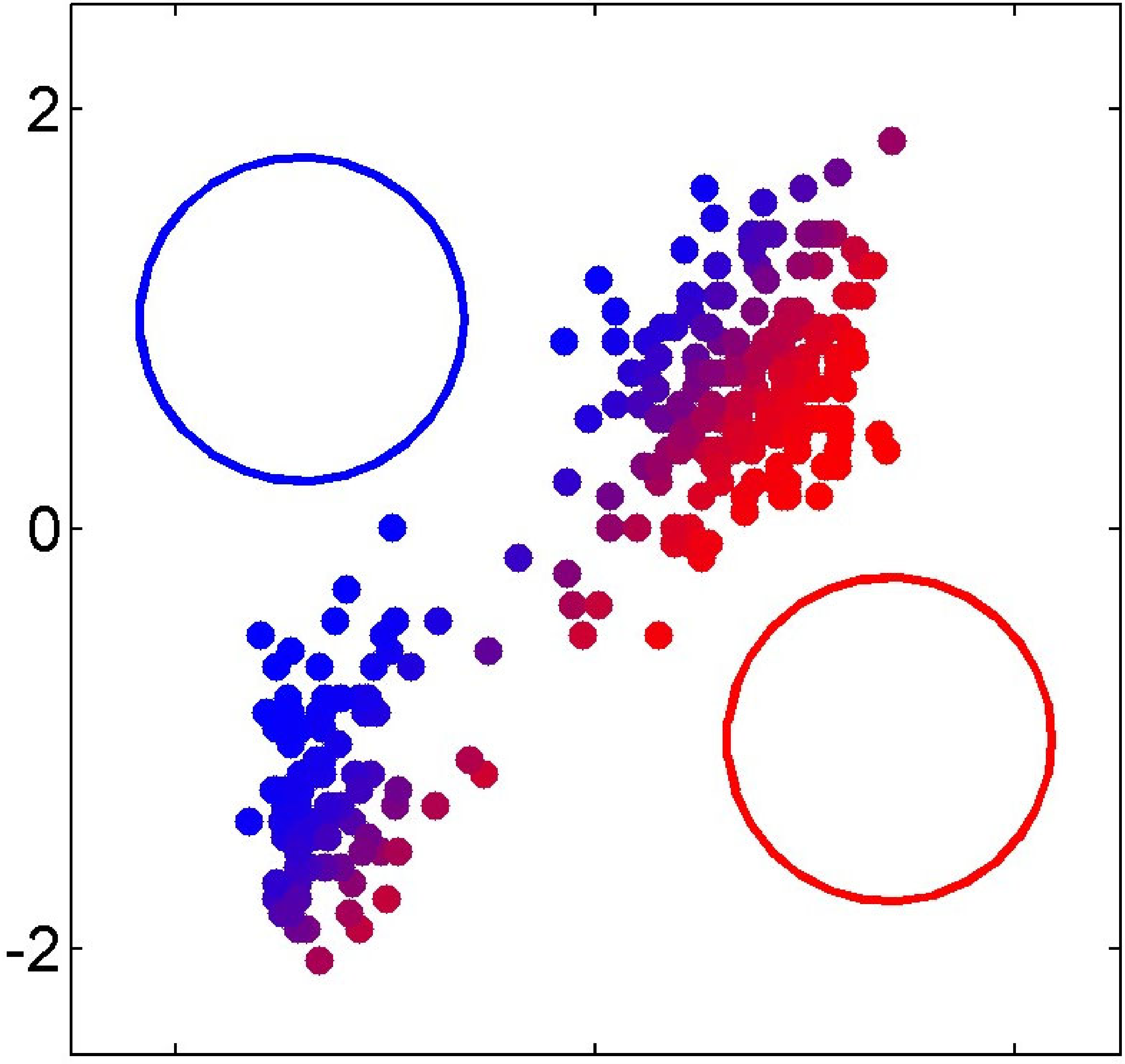
Similarly for the covariances

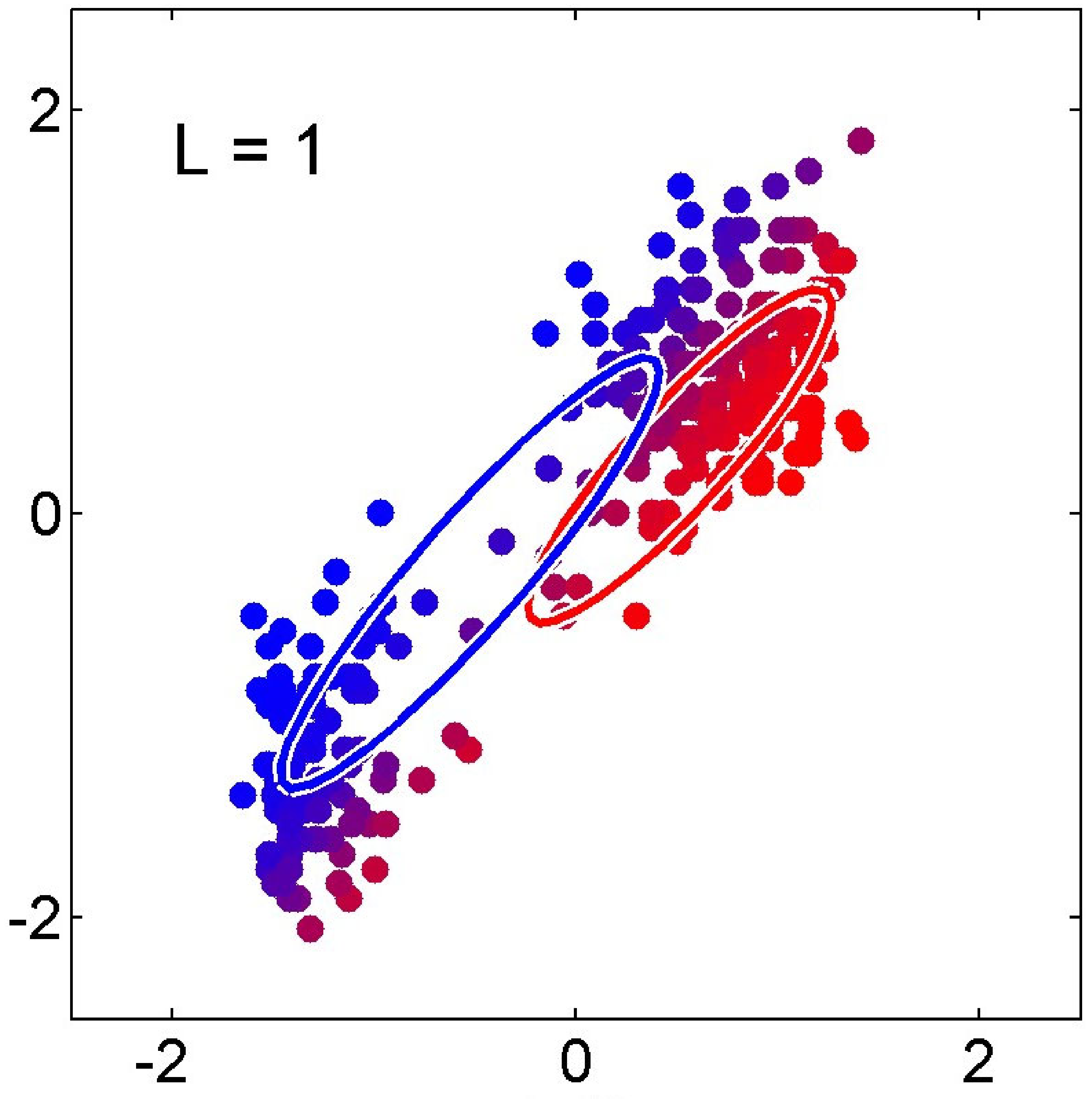
$$\boldsymbol{\Sigma}_j = \frac{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)(\mathbf{x}_n - \boldsymbol{\mu}_j)(\mathbf{x}_n - \boldsymbol{\mu}_j)^{\top}}{\sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)}$$

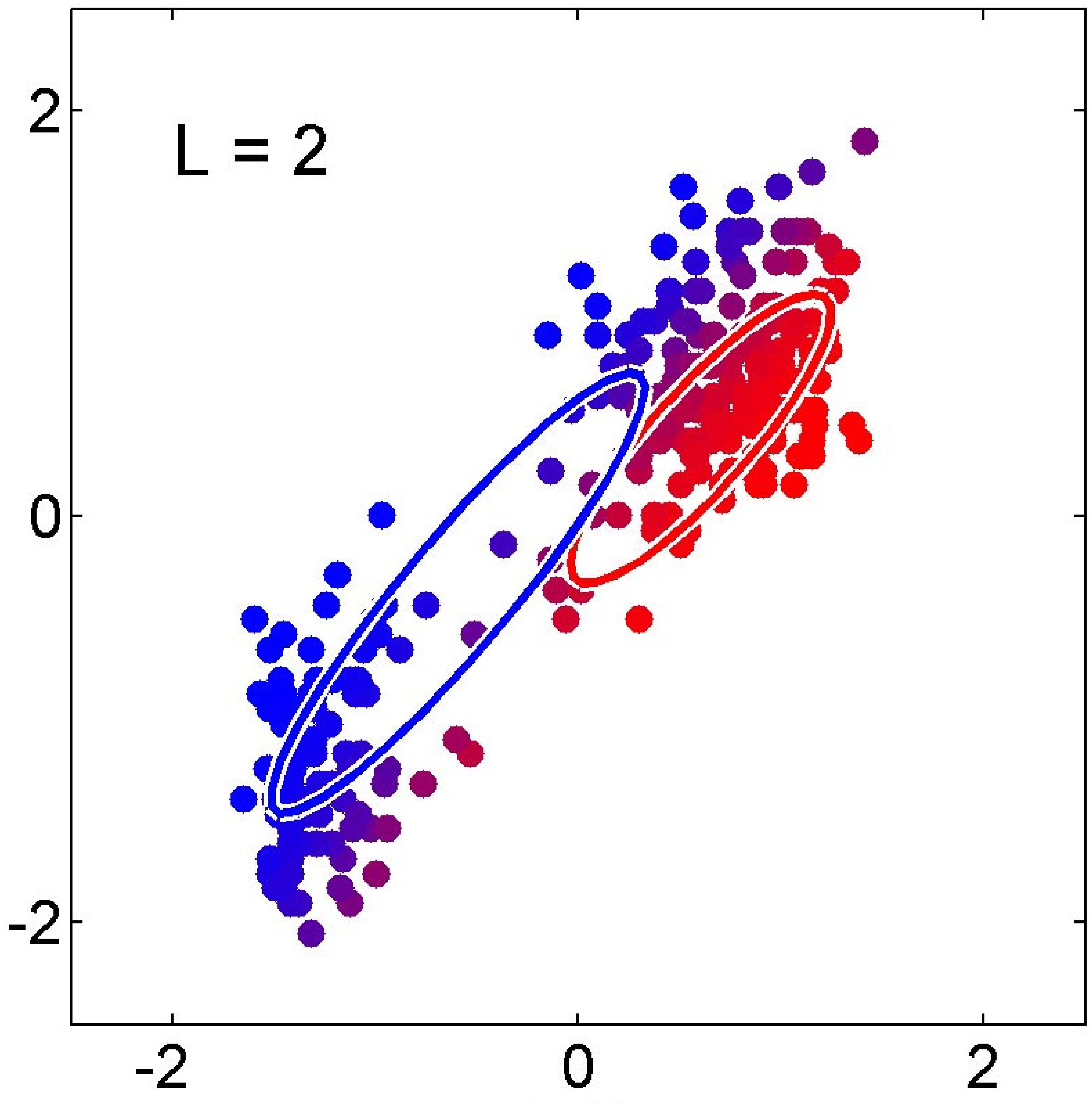
For mixing coefficients use a Lagrange multiplier (constrain: sum up to 1)

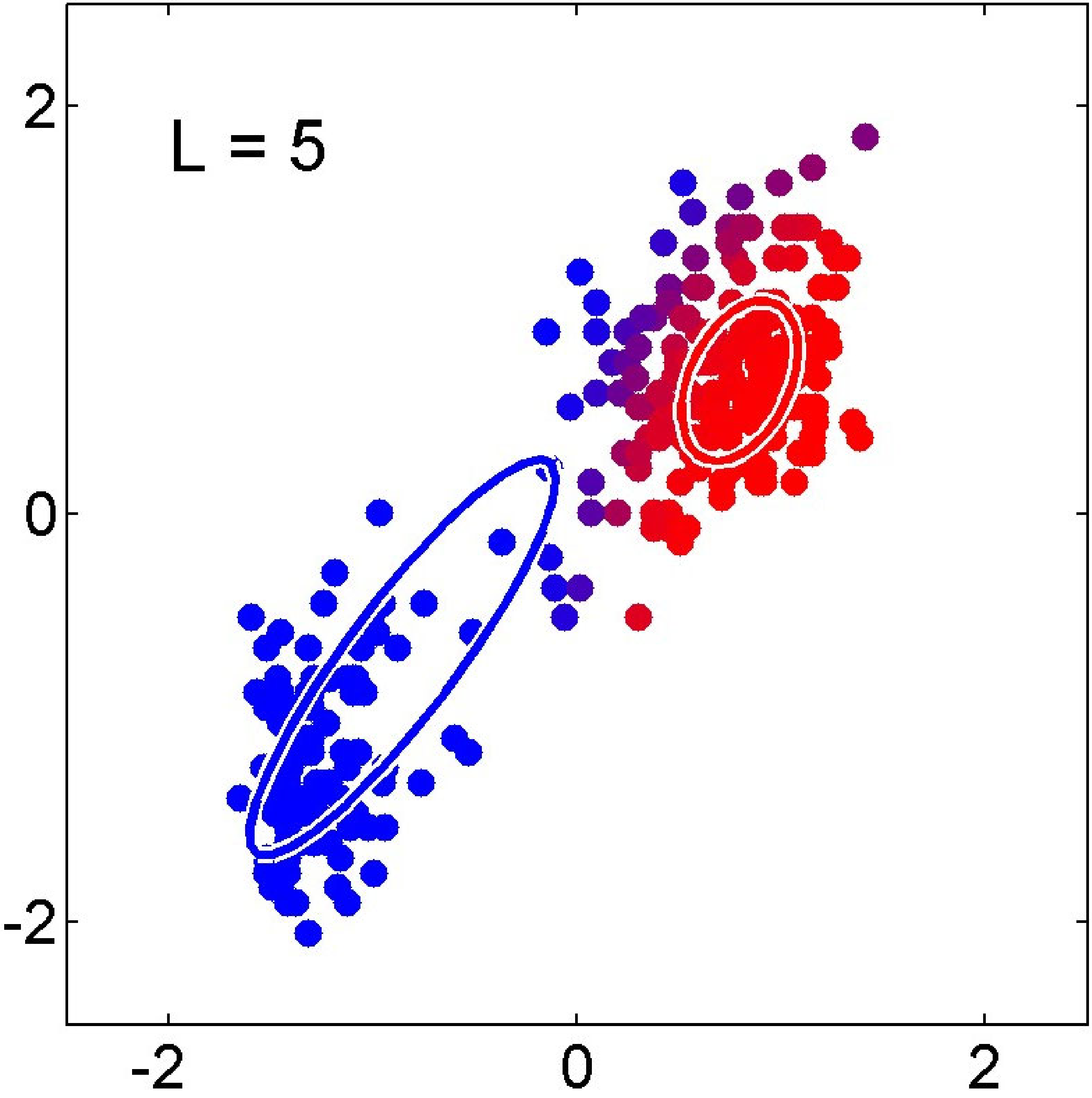
$$\pi_j = \frac{1}{N} \sum_{n=1}^{N} \gamma_j(\mathbf{x}_n)$$

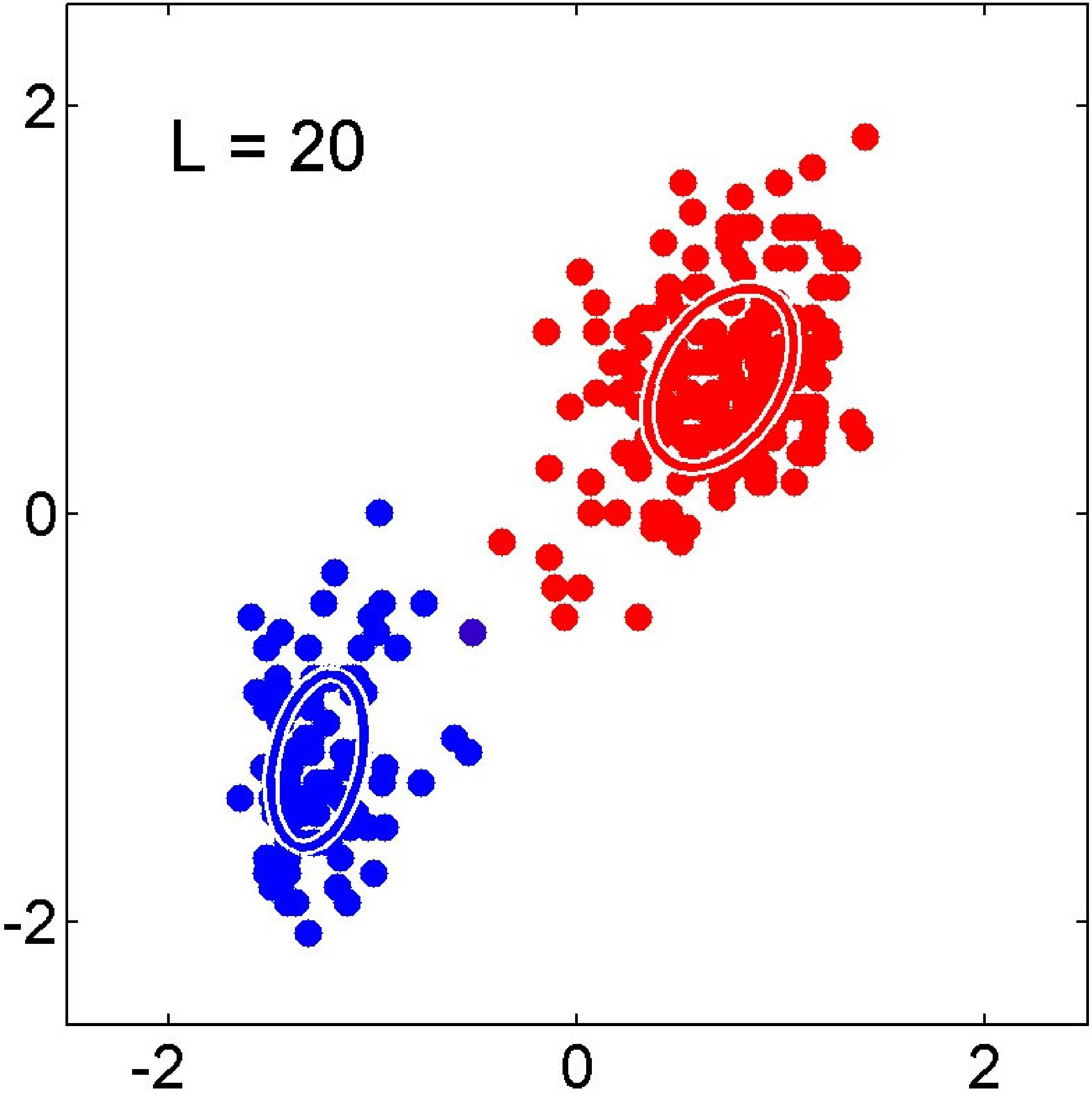

2
0
-2
-2
0
2


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

The EM Algorithm in general

Given observed variable X, unobserved Z

E-step: probabilistically fill in the missing data

M-step: maximize Q to find the new θ’.

$$Q(\theta'|\theta) = E_{Z|X,\theta}[\log P(X, Z|\theta')]$$