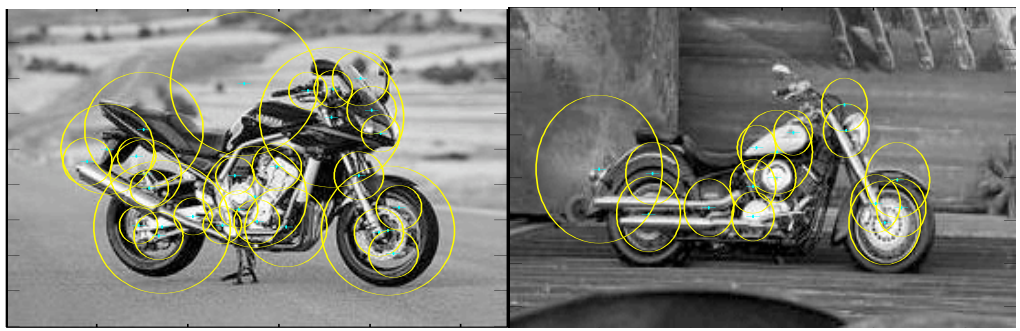
Clustering and Gaussian Mixtures

Oliver Schulte - CMPT 883

Learning with Latent Variables

- In many settings not all variables are observed (labelled) in the training data: $\boldsymbol{x}_i = (\boldsymbol{x}_i, \boldsymbol{h}_i)$
  - e.g. Speech recognition: have speech signals, but not phoneme labels
  - e.g. Object recognition: have object labels (car, bicycle), but not part labels (wheel, door, seat)
  - Unobserved variables are called latent variables



figs from Fergus et al.

Latent Variables and Simplicity

- Latent variables can explain observed correlations with a simple model.
  - Fewer parameters.
  - Common in science: The heart, genes, energy, gravity, ...

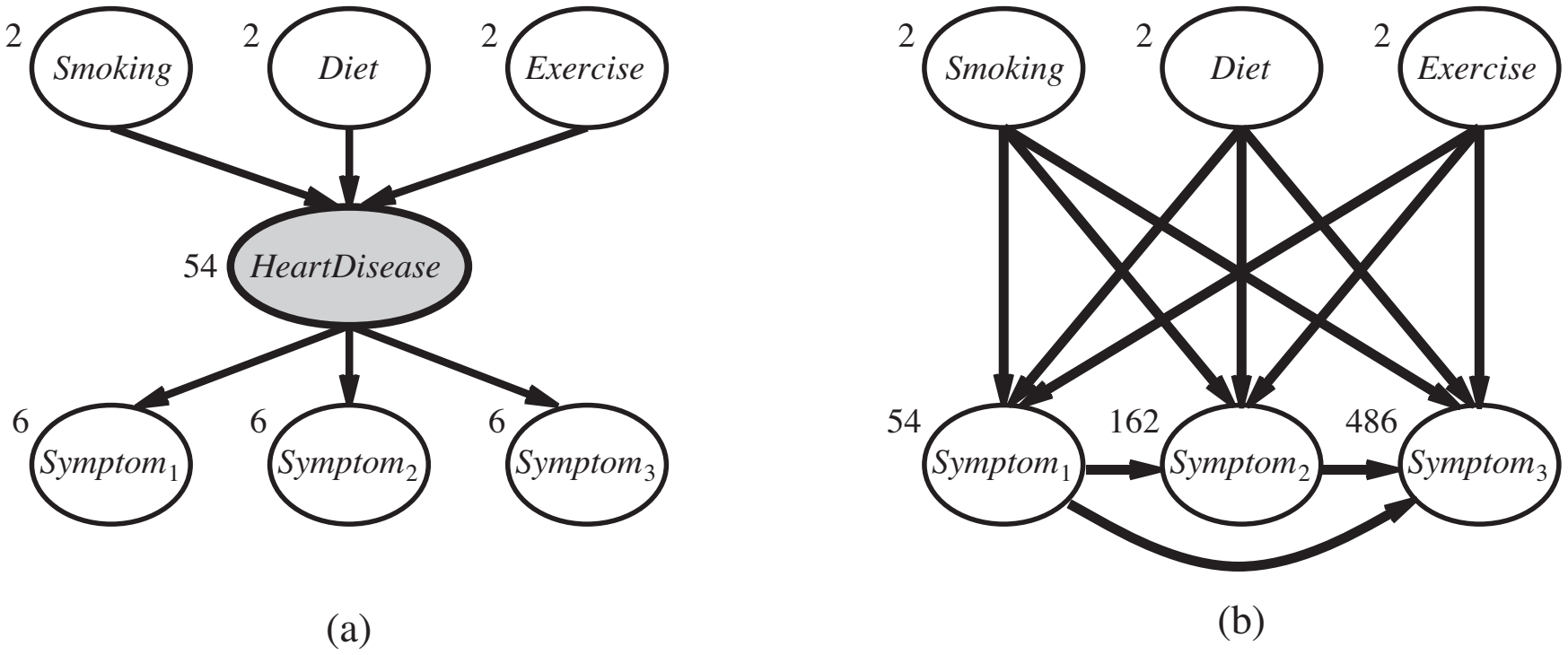


Fig. Russell and Norvig 20.10

Latent Variable Models: Pros

- Statistically powerful, often good predictions. Many applications:
- Learning with **missing data**.
- **Clustering:** "missing" cluster label for data points.
- **Principal Component Analysis:** data points are generated in linear fashion from a small set of unobserved components. (more later)
- **Matrix Factorization, Recommender Systems:**
  - Assign users to unobserved "user types", assign items to unobserved "item types".
  - Use similarity between user type, item type to predict preference of user for item.
  - Winner of $1M Netflix challenge.
- If latent variables have an intuitive interpretation (e.g., "action movies", "factors"), discovers **new features**.

Latent Variable Models: Cons

- From a user's point of view, like a black box if latent variables don't have an intuitive interpretation.
- Statistically, hard to guarantee convergence to a correct model with more data (the identifiability problem).
- Harder computationally, usually no closed form for maximum likelihood estimates.

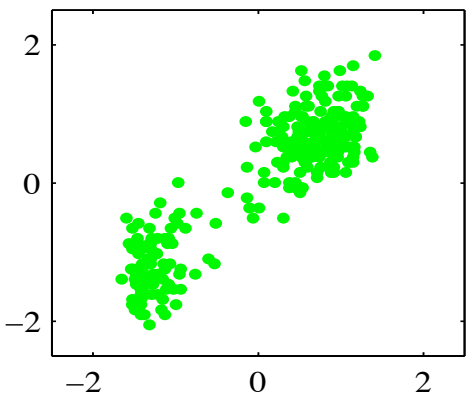
Latent Learning Methods

- The Expectation-Maximization algorithm provides a *general-purpose* local search algorithm for learning parameter values in probabilistic models with latent variables.
- Deep Learning investigates methods for finding new latent variables.

Outline

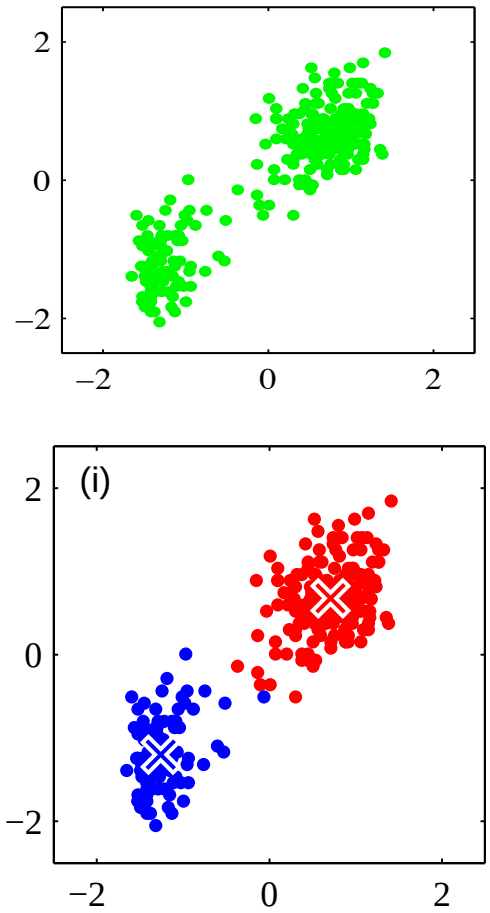
Outline

Clustering



- Given a dataset $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, each $\boldsymbol{x}_i \in \mathbb{R}^D$, partition the dataset into K clusters
- Intuitively, a cluster is a group of points, which are close together and far from others

Distortion Measure



- Formally, introduce prototypes (or cluster centers) $\boldsymbol{\mu}_k \in \mathbb{R}^D$
- Use binary r_{nk}, 1 if point n is in cluster k, 0 otherwise (1-of-K coding scheme again)
- Find $\{\boldsymbol{\mu}_k\}$, $\{r_{nk}\}$ to minimize distortion measure:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

Minimizing Distortion Measure

- Minimizing J directly is hard

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- However, two things are easy
  - If we know μ_k, minimizing J wrt r_{nk}
  - If we know r_{nk}, minimizing J wrt μ_k
- This suggests an iterative procedure
  - Start with initial guess for μ_k
  - Iteration of two steps:
    - Minimize J wrt r_{nk}
    - Minimize J wrt μ_k
  - Rinse and repeat until convergence

Minimizing Distortion Measure

- Minimizing J directly is hard

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- However, two things are easy
  - If we know $\boldsymbol{\mu}_k$, minimizing J wrt r_{nk}
  - If we know r_{nk}, minimizing J wrt $\boldsymbol{\mu}_k$
- This suggests an iterative procedure
  - Start with initial guess for $\boldsymbol{\mu}_k$
  - Iteration of two steps:
    - Minimize J wrt r_{nk}
    - Minimize J wrt $\boldsymbol{\mu}_k$
  - Rinse and repeat until convergence

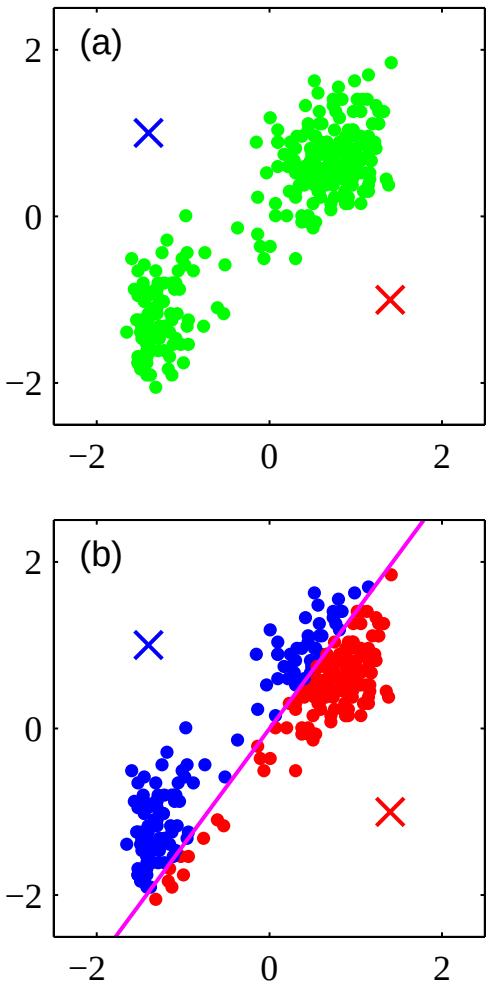
Minimizing Distortion Measure

- Minimizing J directly is hard

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- However, two things are easy
  - If we know $\boldsymbol{\mu}_k$, minimizing J wrt r_{nk}
  - If we know r_{nk}, minimizing J wrt $\boldsymbol{\mu}_k$
- This suggests an iterative procedure
  - Start with initial guess for $\boldsymbol{\mu}_k$
  - Iteration of two steps:
    - Minimize J wrt r_{nk}
    - Minimize J wrt $\boldsymbol{\mu}_k$
  - Rinse and repeat until convergence

Determining Membership Variables



- Step 1 in an iteration of K-means is to minimize distortion measure J wrt cluster membership variables r_{nk}

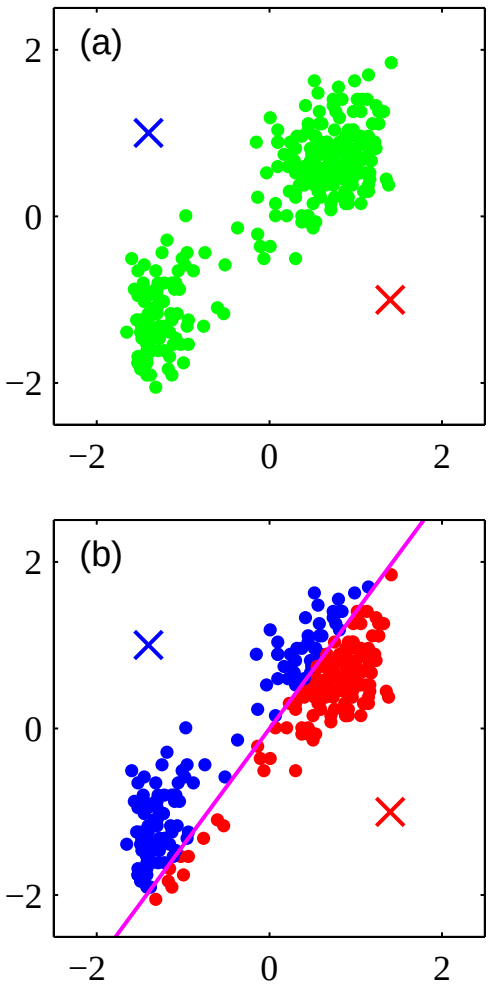
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Terms for different data points x_n are independent, for each data point set r_{nk} to minimize:

$$\sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 \text{ How?}$$

- Simply set $r_{nk} = 1$ for the cluster center μ_k with smallest distance

Determining Membership Variables



- Step 1 in an iteration of K-means is to minimize distortion measure J wrt cluster membership variables r_{nk}

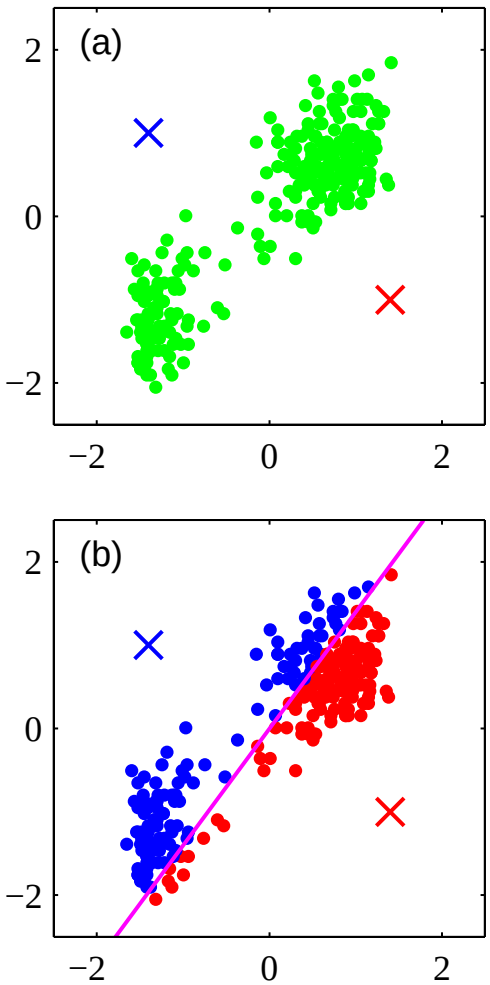
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Terms for different data points $\boldsymbol{x}_n$ are independent, for each data point set r_{nk} to minimize:

$$\sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 \text{ How?}$$

- Simply set $r_{nk} = 1$ for the cluster center μ_k with smallest distance

Determining Membership Variables



- Step 1 in an iteration of K-means is to minimize distortion measure J wrt cluster membership variables r_{nk}

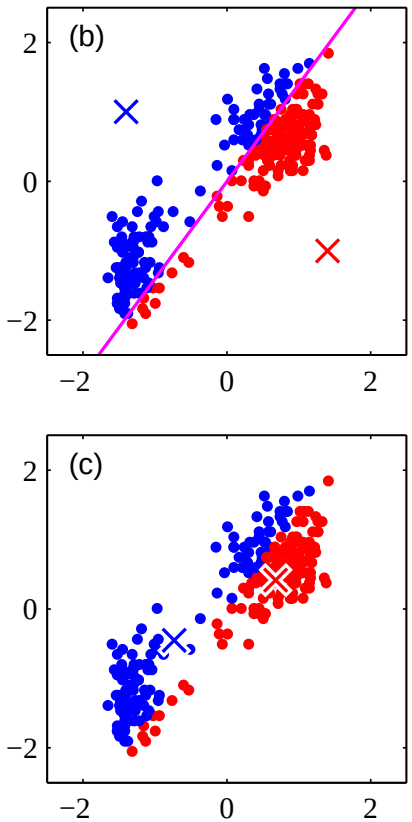
$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$

- Terms for different data points $\boldsymbol{x}_n$ are independent, for each data point set r_{nk} to minimize:

$$\sum_{k=1}^{K} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2 \text{ How?}$$

- Simply set $r_{nk} = 1$ for the cluster center $\boldsymbol{\mu}_k$ with smallest distance

Determining Cluster Centers



- Step 2: fix r_{nk}, minimize J wrt the cluster centers $\boldsymbol{\mu}_k$

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$ switch order of sums

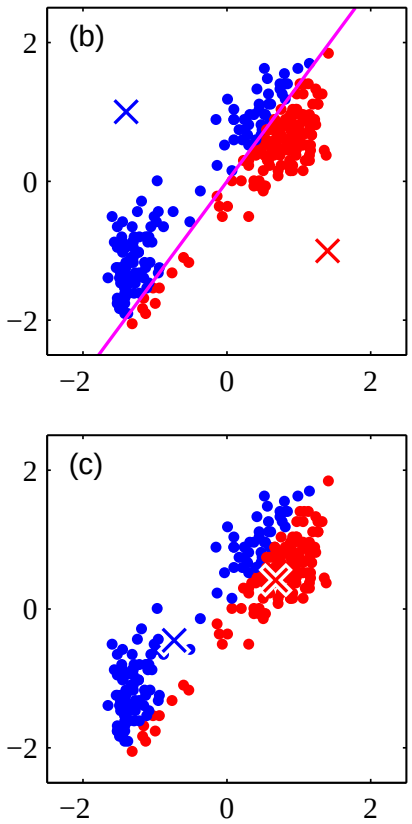
- Exercise: minimize wrt each $\boldsymbol{\mu}_k$ separately.
- Take derivative, set to zero:

$$2 \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\Leftrightarrow \boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \boldsymbol{x}_n}{\sum_n r_{nk}}$$

i.e. mean of datapoints $\boldsymbol{x}_n$ assigned to cluster k

Determining Cluster Centers



- Step 2: fix r_{nk}, minimize J wrt the cluster centers $\boldsymbol{\mu}_k$

$$J = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} ||\boldsymbol{x}_n - \boldsymbol{\mu}_k||^2$$ switch order of sums

- Exercise: minimize wrt each $\boldsymbol{\mu}_k$ separately.
- Take derivative, set to zero:

$$2 \sum_{n=1}^{N} r_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k) = 0$$

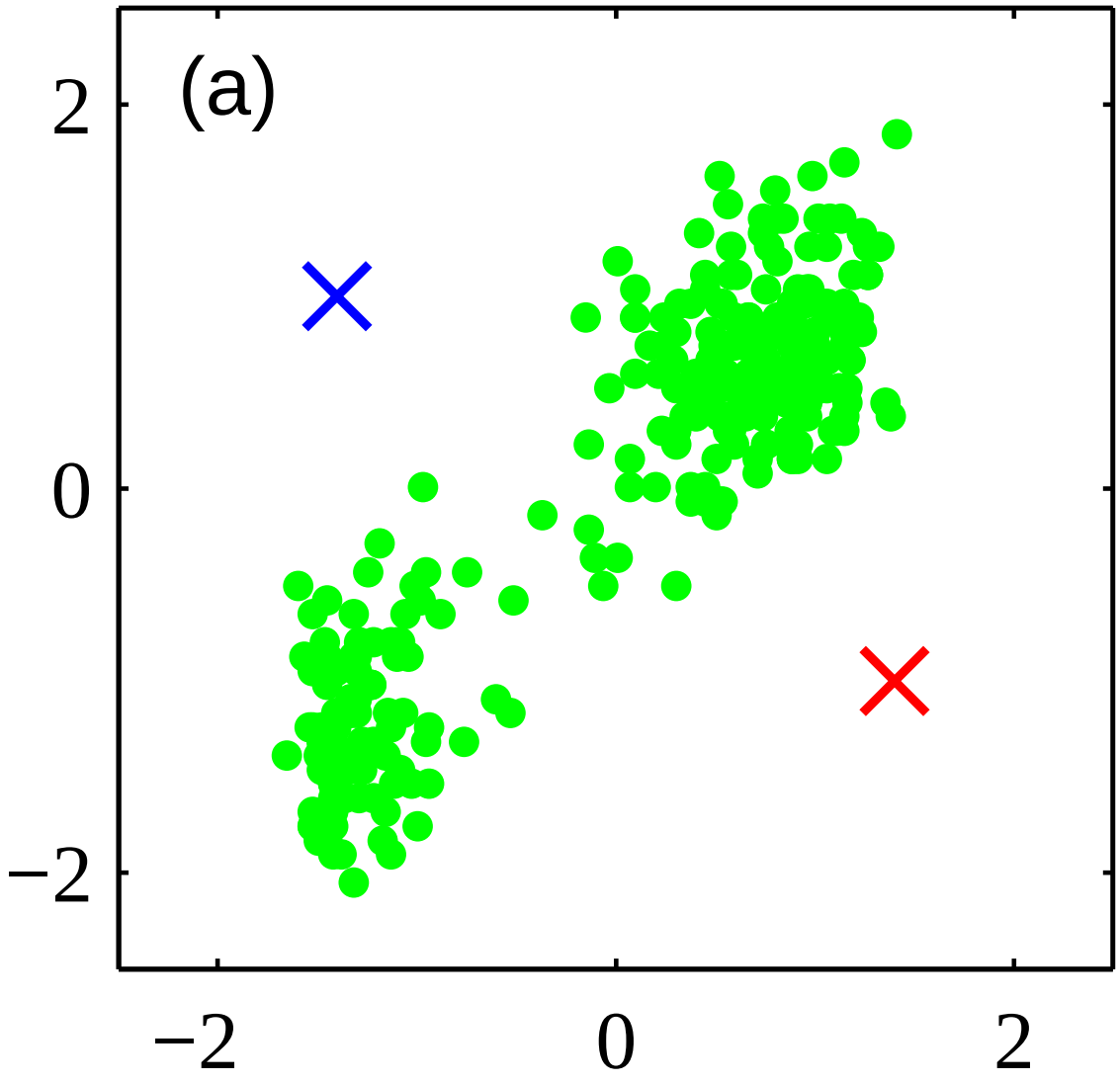
$$\Leftrightarrow \boldsymbol{\mu}_k = \frac{\sum_n r_{nk} \boldsymbol{x}_n}{\sum_n r_{nk}}$$

i.e. mean of datapoints $\boldsymbol{x}_n$ assigned to cluster k

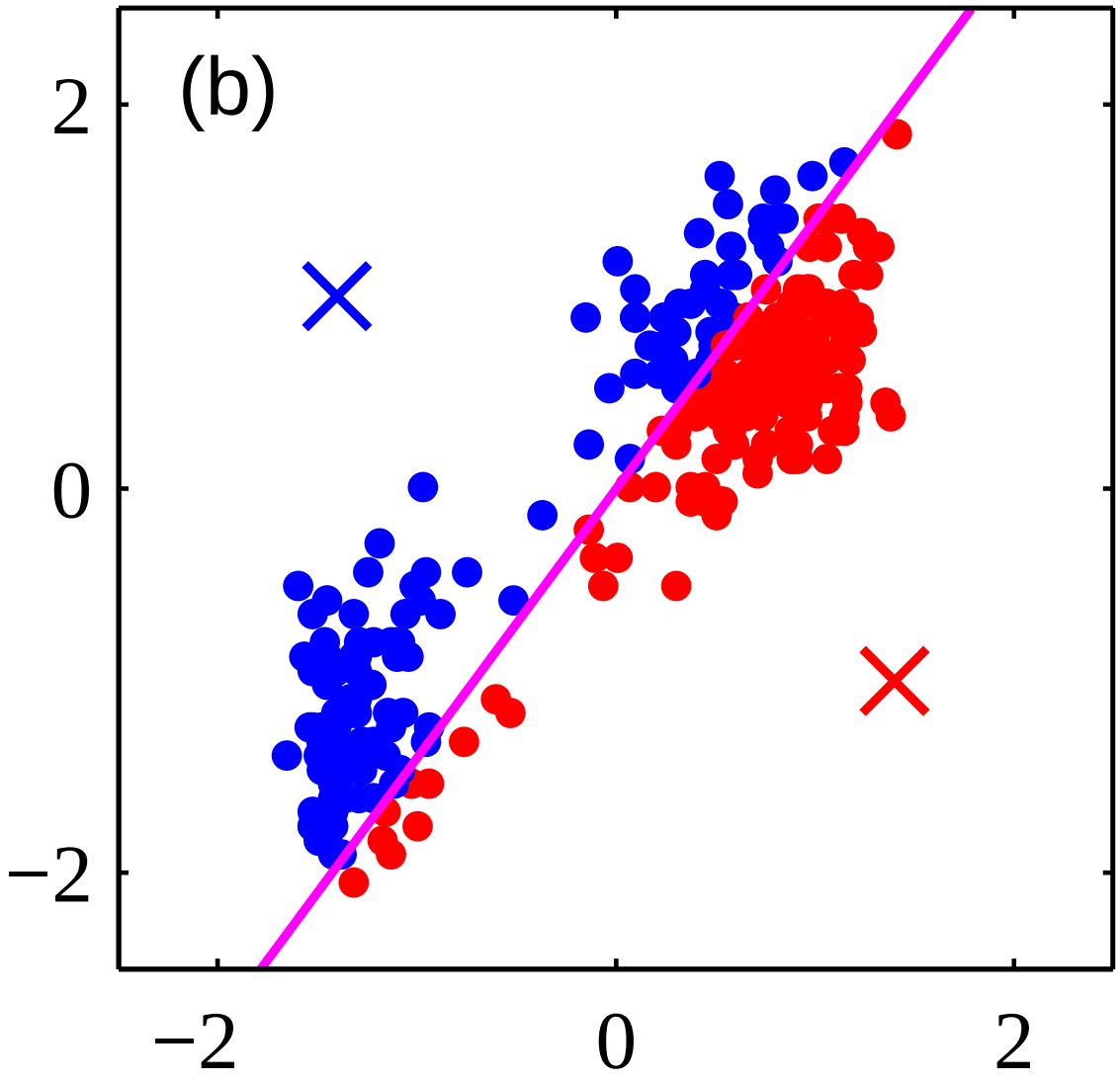
K-means Algorithm

- Start with initial guess for $\boldsymbol{\mu}_k$
- Iteration of two steps:
  - Minimize J wrt r_{nk}
    - Assign points to nearest cluster center
  - Minimize J wrt $\boldsymbol{\mu}_k$
    - Set cluster center as average of points in cluster
- Rinse and repeat until convergence

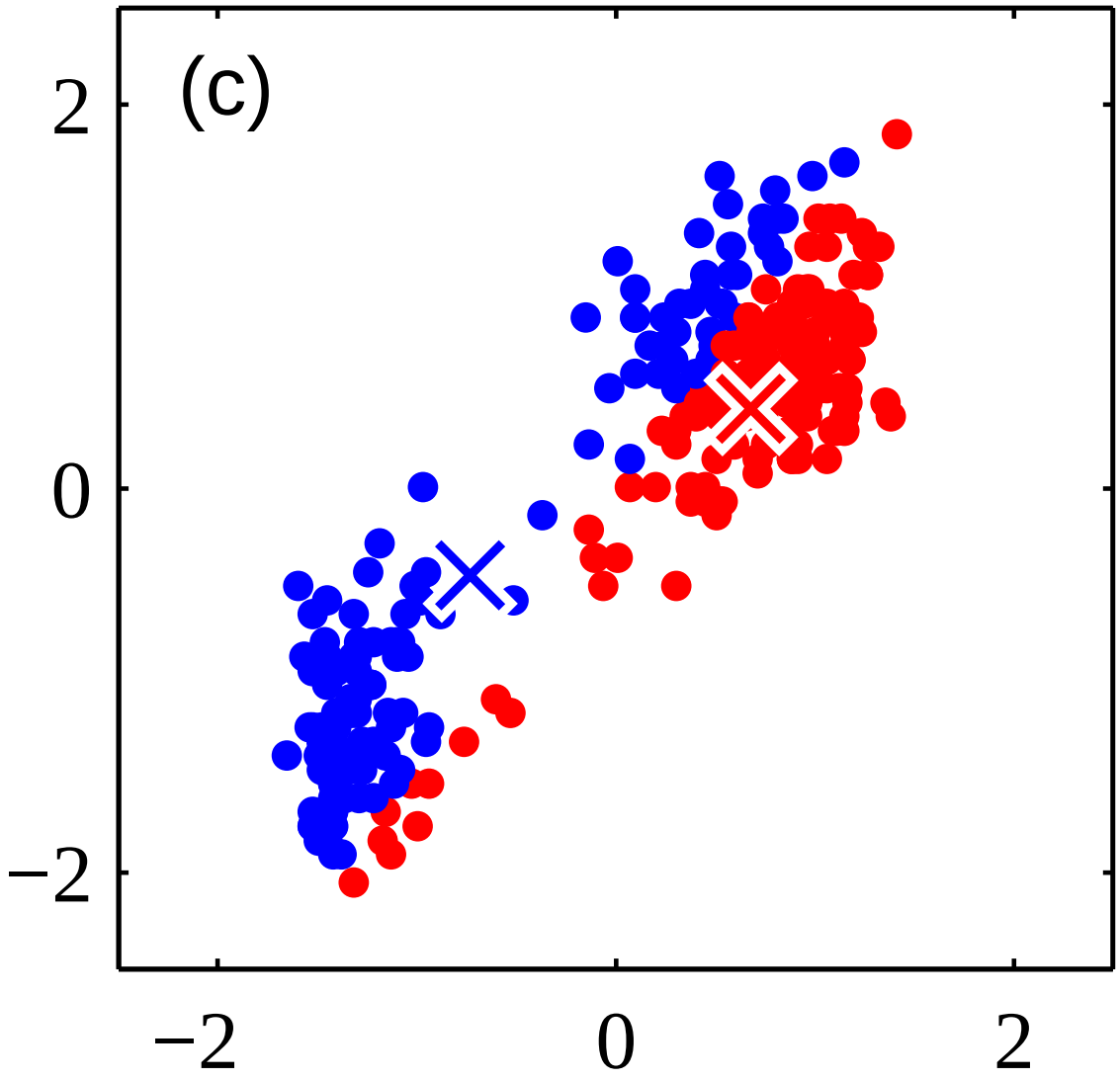
K-means example



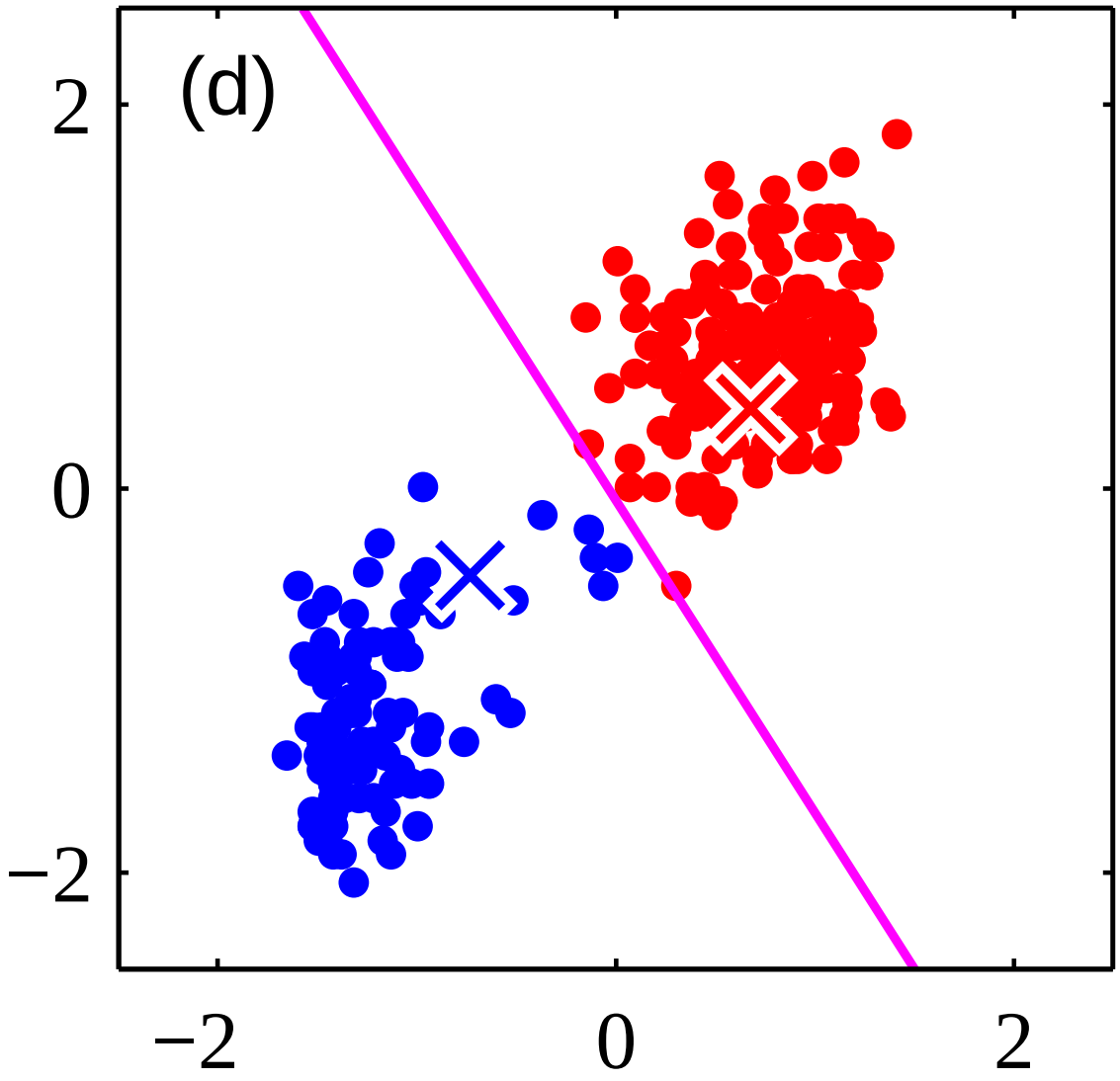
K-means example



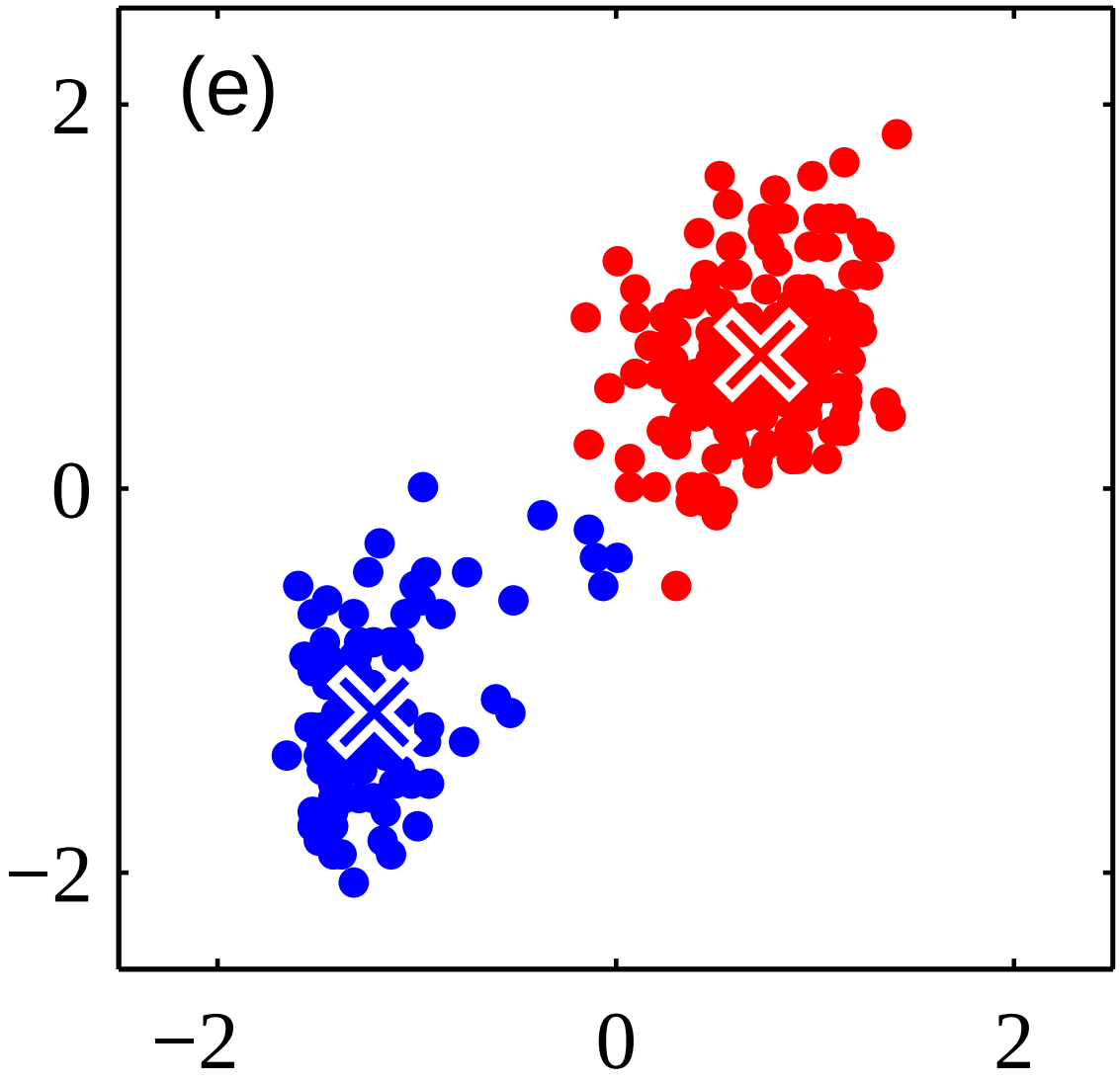
K-means example



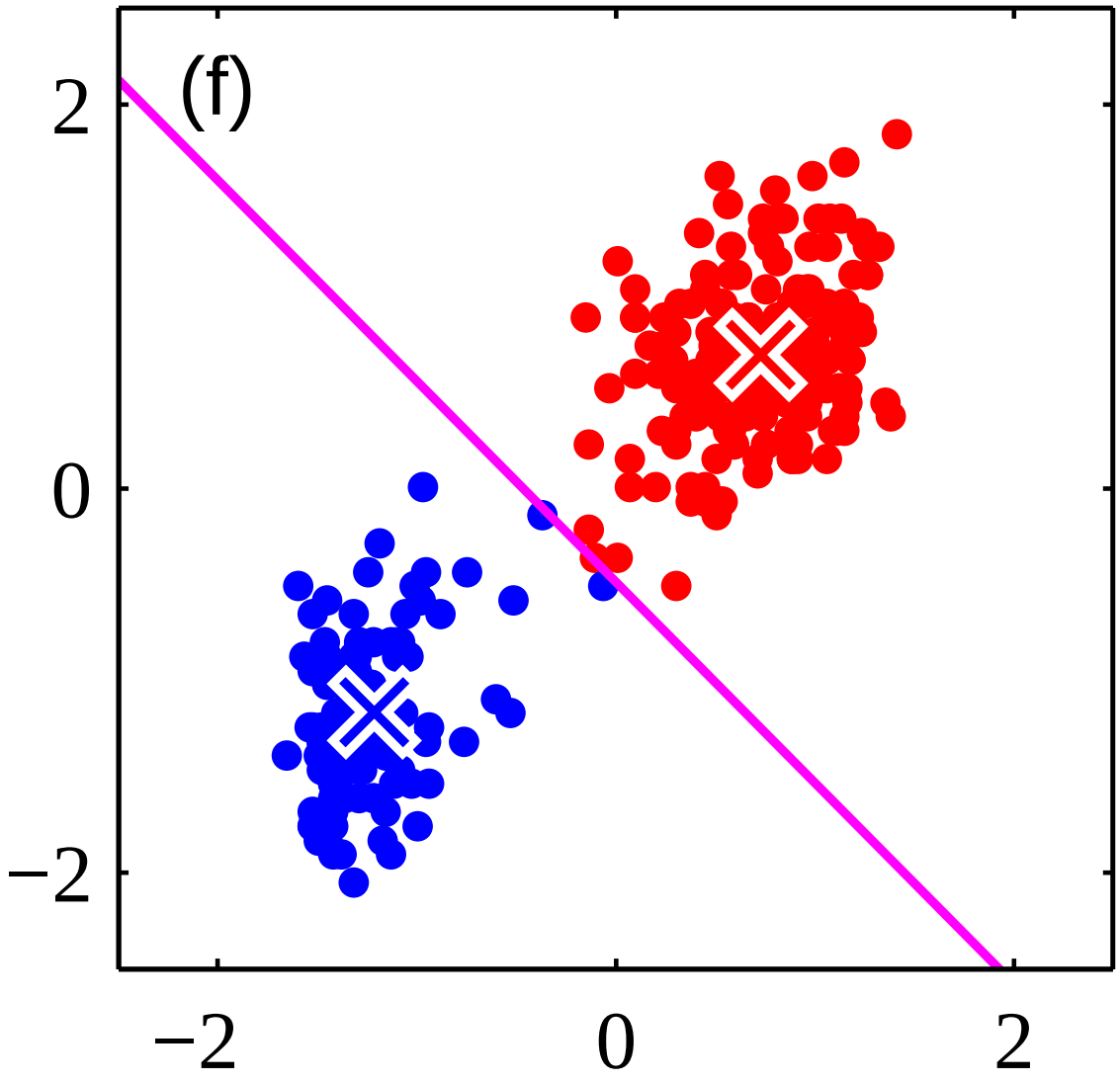
K-means example



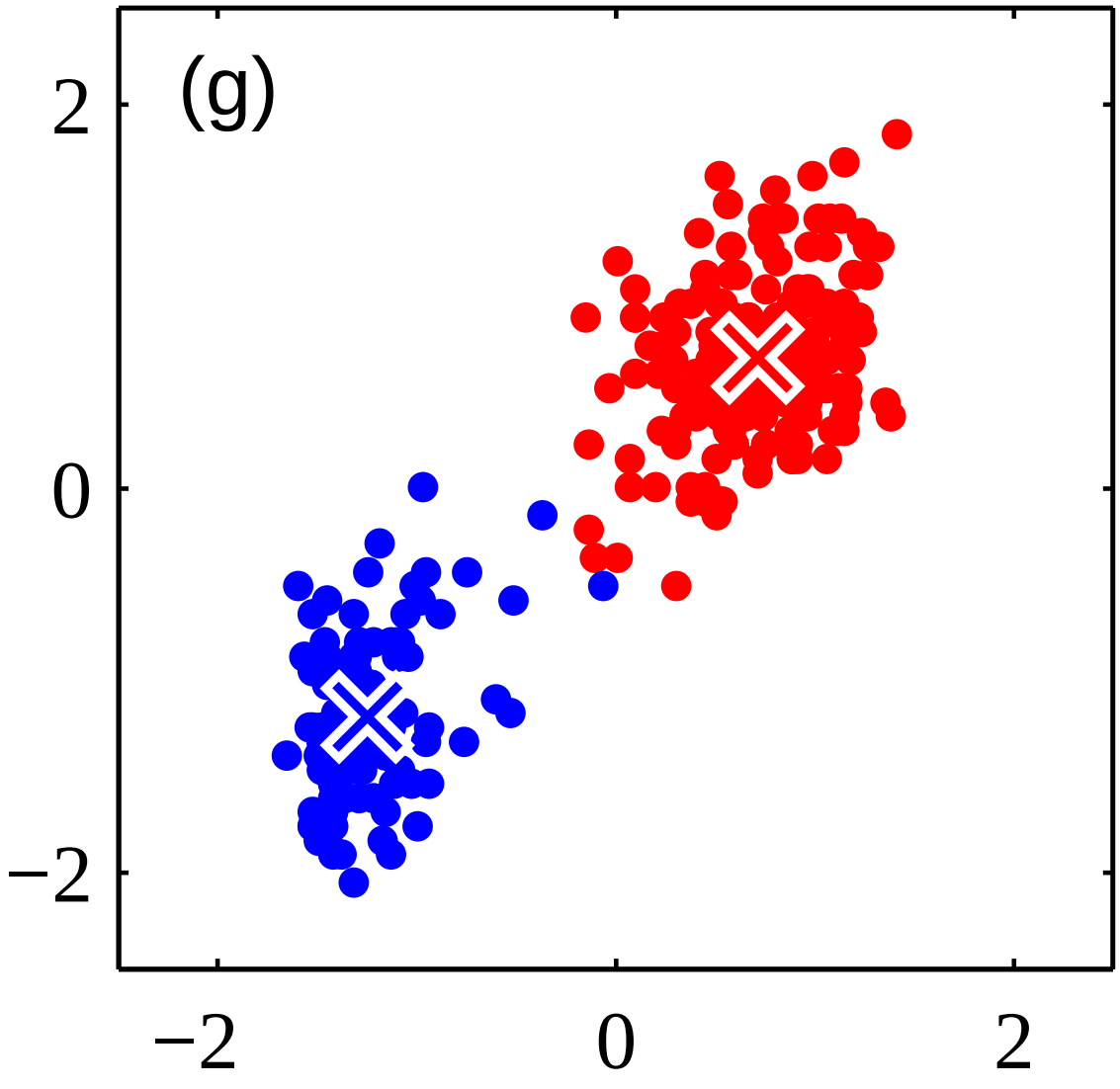
K-means example



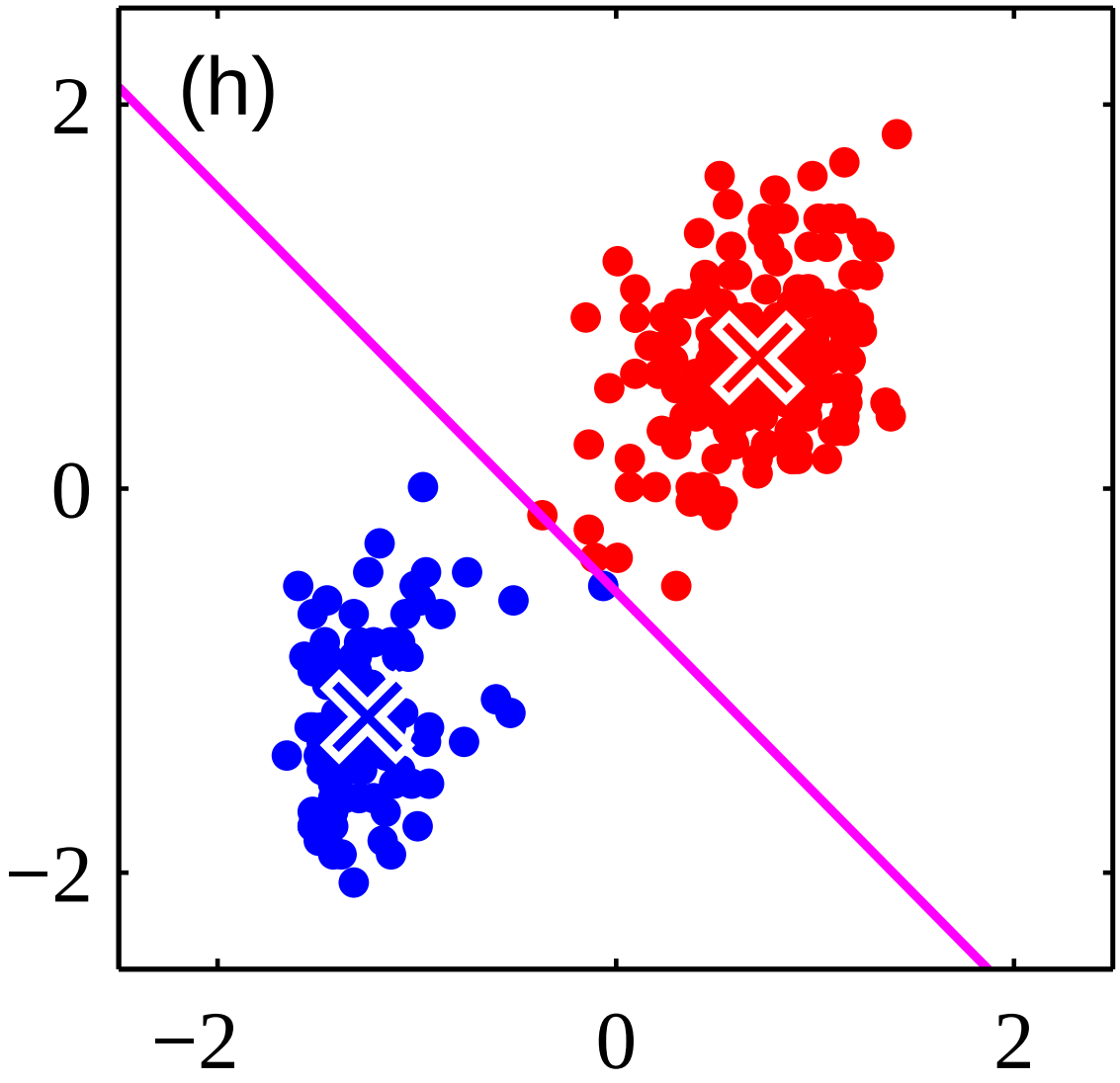
K-means example



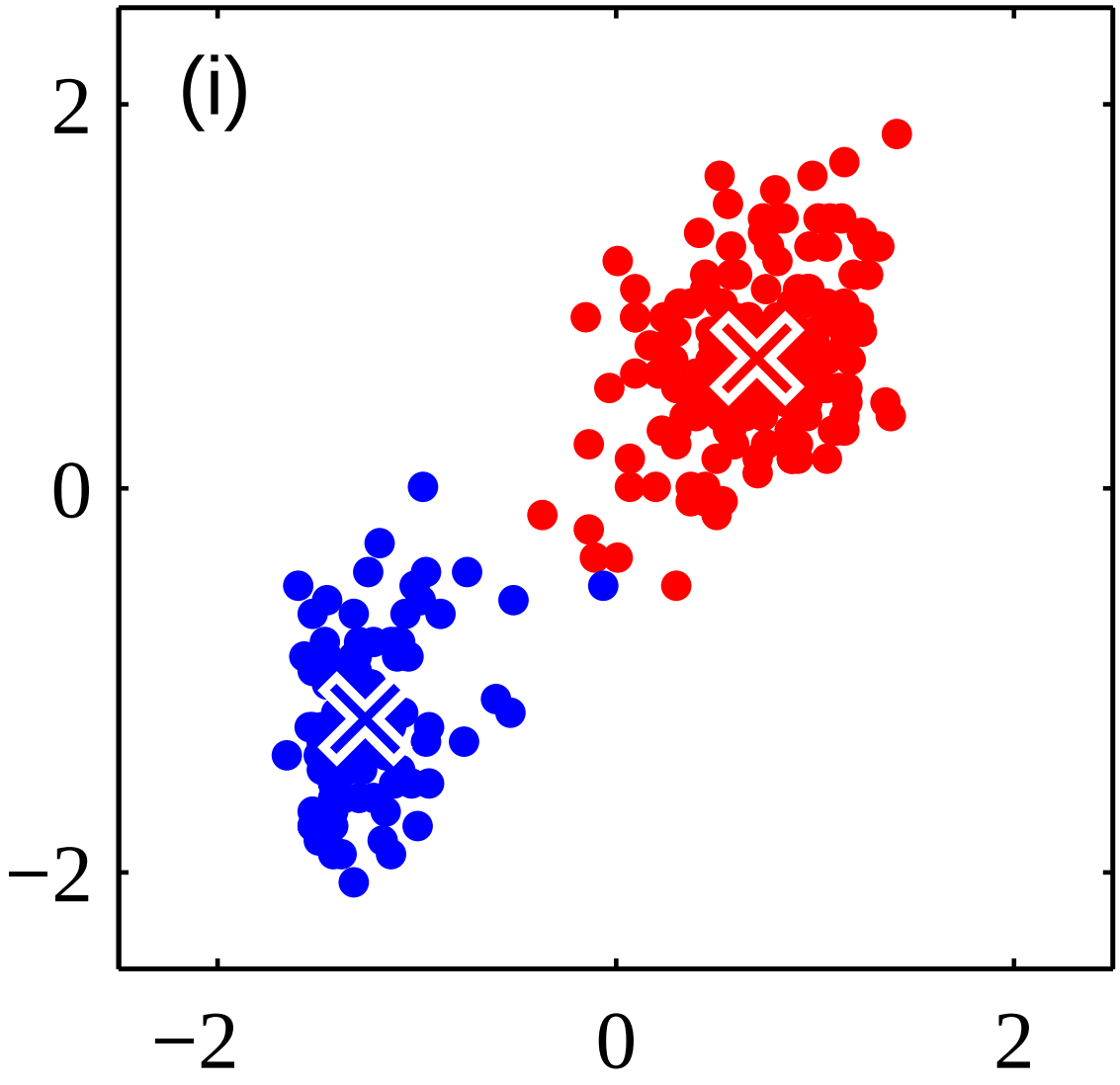
K-means example



K-means example



K-means example



Next step doesn't change membership – stop

K-means Convergence

- Repeat steps until no change in cluster assignments
- For each step, value of J either goes down, or we stop
- Finite number of possible assignments of data points to clusters, so we are guaranteed to converge eventually
- Note it may be a local maximum rather than a global maximum to which we converge

K-means Example - Image Segmentation

$K = 2$

$K = 3$

$K = 10$

Original image

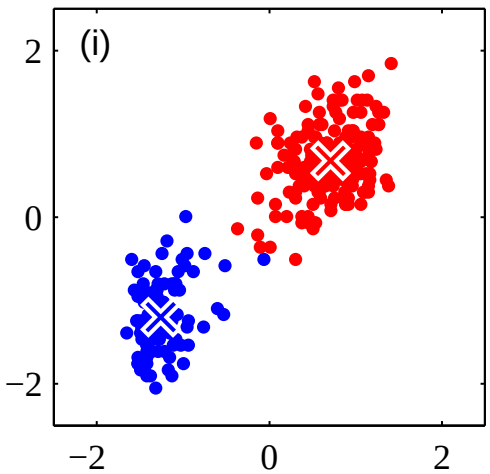
- K-means clustering on pixel colour values
- Pixels in a cluster are coloured by cluster mean
- Represent each pixel (e.g. 24-bit colour value) by a cluster number (e.g. 4 bits for $K = 10$), compressed version

Hierarchical Clustering

See Powerpoint slides.

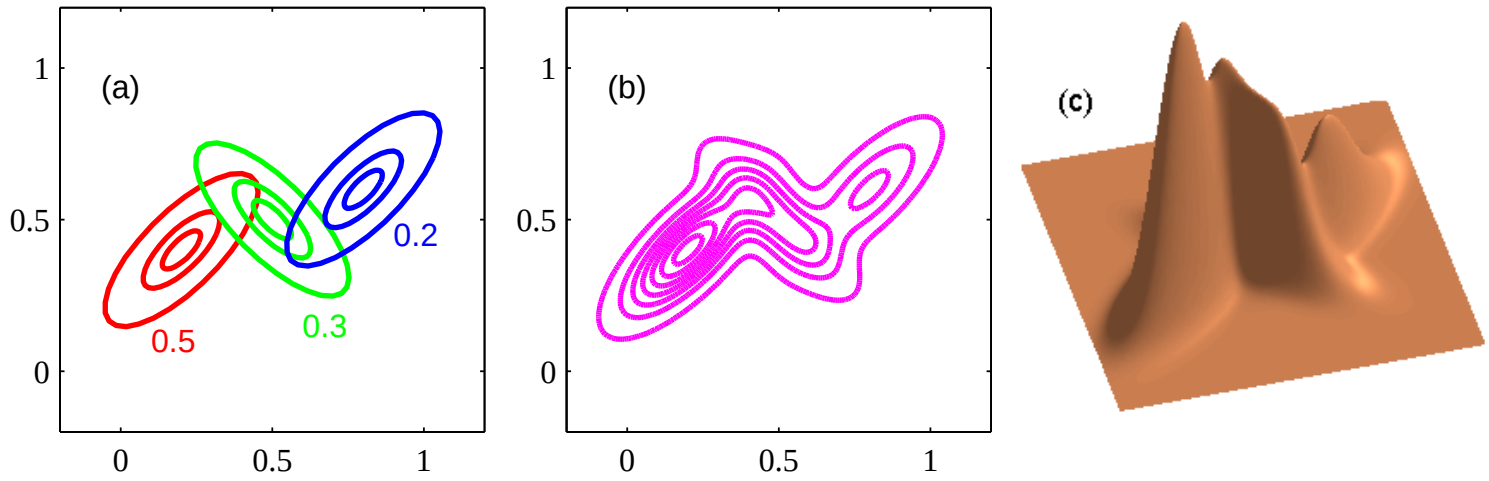
Outline

Hard Assignment vs. Soft Assignment



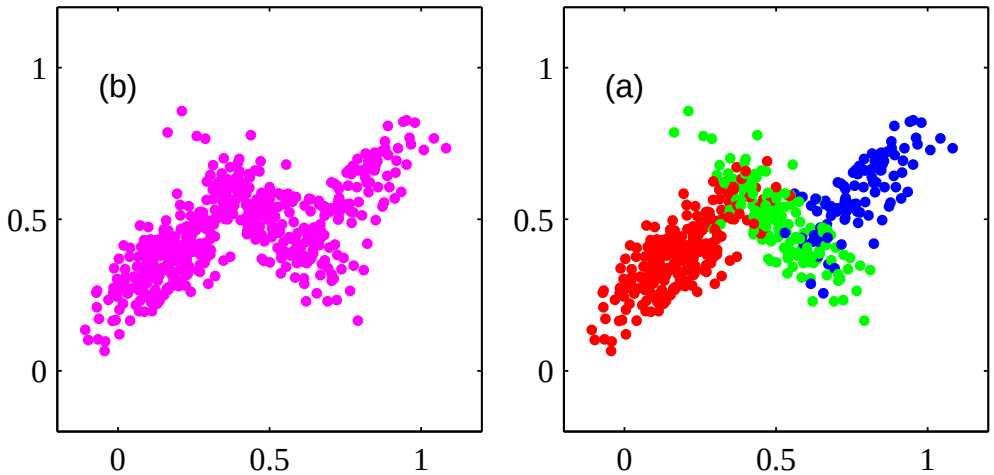
- In the K-means algorithm, a hard assignment of points to clusters is made
- However, for points near the decision boundary, this may not be such a good idea
- Instead, we could think about making a soft assignment of points to clusters

Gaussian Mixture Model



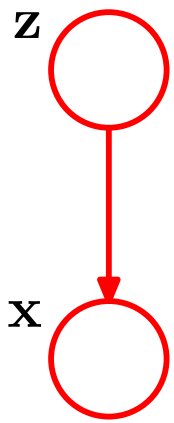
- The Gaussian mixture model (or mixture of Gaussians MoG) models the data as a combination of Gaussians.
- a: constant density contours. b: marginal probability $p(\boldsymbol{x})$. c: surface plot.
- Widely used general approximation for multi-modal distributions. (Dog Show)

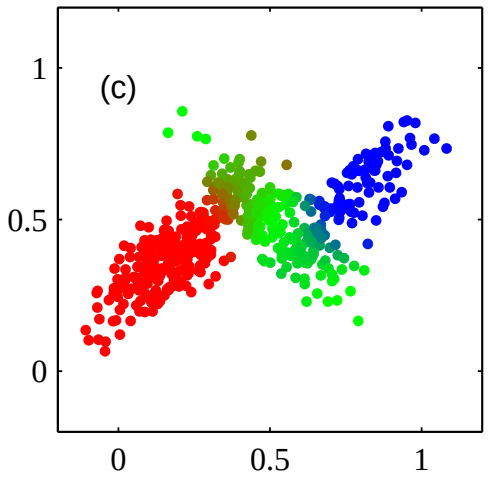
Gaussian Mixture Model



- Above shows a dataset generated by drawing samples from three different Gaussians

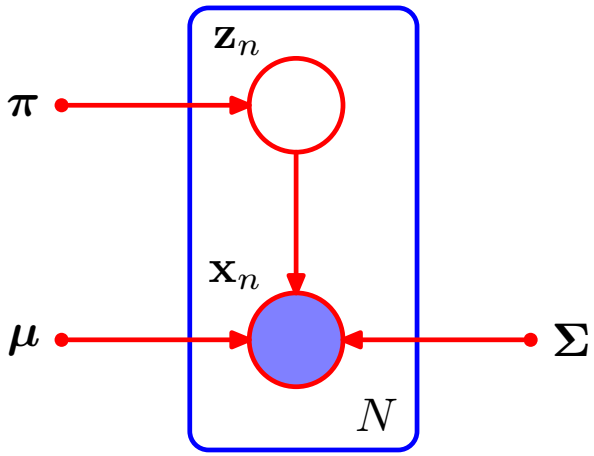
Generative Model

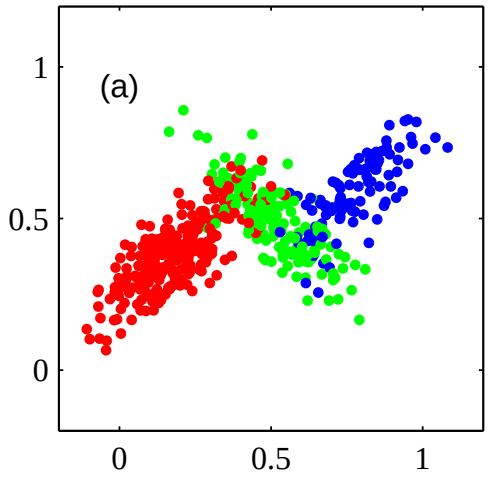




- The mixture of Gaussians is a generative model
- To generate a datapoint $\boldsymbol{x}_n$, we first generate a value for a discrete variable $z_n \in \{1, \ldots, K\}$
- We then generate a value $\boldsymbol{x}_n \sim \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ for the corresponding Gaussian.

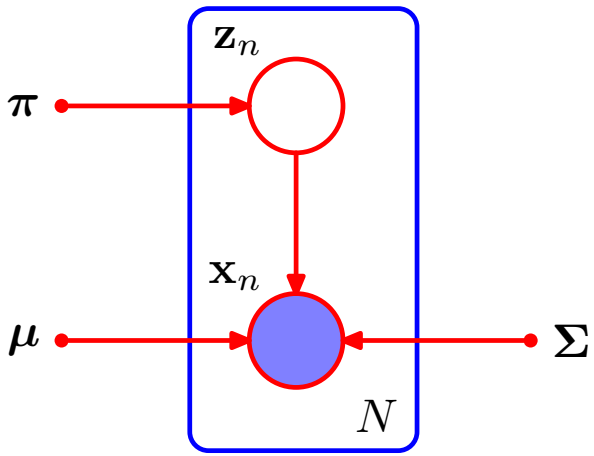
Graphical Model

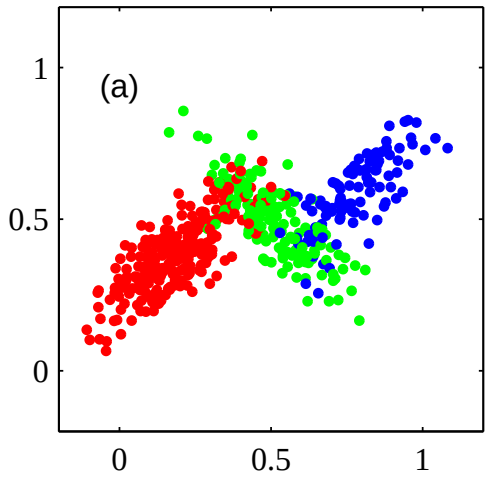




- Full graphical model using plate notation
  - Note z_n is a latent variable, unobserved
- Bayes net needs distributions $p(z_n)$ and $p(\boldsymbol{x}_n|z_n)$
- The one-of-K representation is helpful here: $z_{nk} \in \{0, 1\}$, $\boldsymbol{z}_n = (z_{n1}, \ldots, z_{nK})$

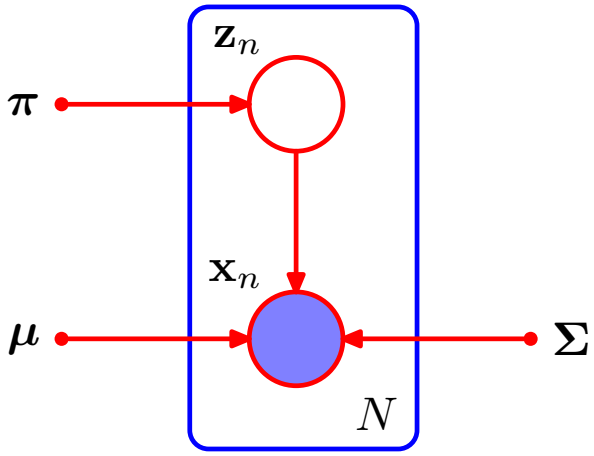
Graphical Model - Latent Component Variable

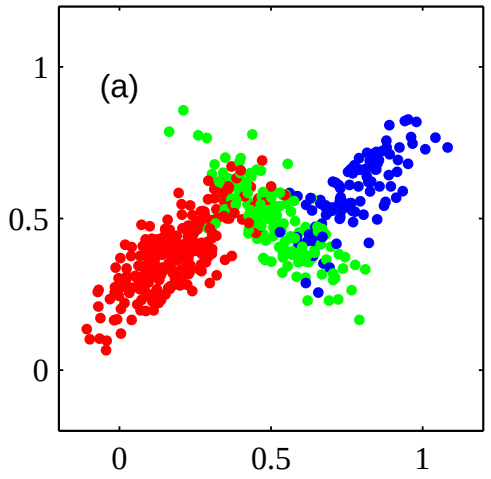




- Use a Bernoulli distribution for $p(\mathbf{z}_n)$
  - i.e. $p(z_{nk} = 1) = \pi_k$
  - Parameters to this distribution $\{\pi_k\}$
  - Must have $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$
- $p(\mathbf{z}_n) = \prod_{k=1}^{K} \pi_k^{z_{nk}}$

Graphical Model - Observed Variable



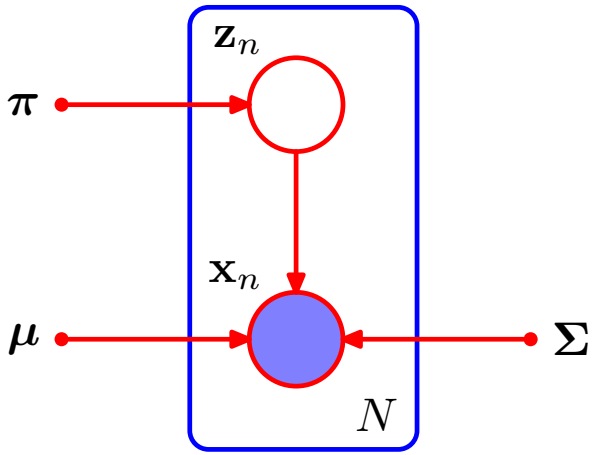


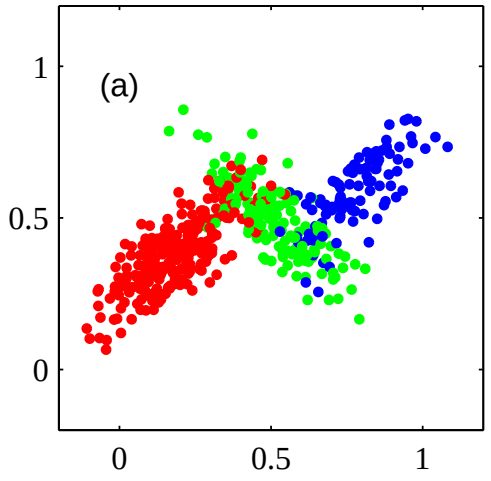
- Use a Gaussian distribution for $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$
  - Parameters to this distribution $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$

$$p(\boldsymbol{x}_n|z_{nk}=1) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$p(\boldsymbol{x}_n|\boldsymbol{z}_n) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}$$

Graphical Model - Joint distribution





- The full joint distribution is given by:

$$
\begin{aligned}
p(\boldsymbol{x}, \boldsymbol{z}) &= \prod_{n=1}^{N} p(\boldsymbol{z}_n) p(\boldsymbol{x}_n | \boldsymbol{z}_n) \\
&= \prod_{n=1}^{N} \prod_{k=1}^{K} \pi_k^{z_{nk}} \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_{nk}}
\end{aligned}
$$

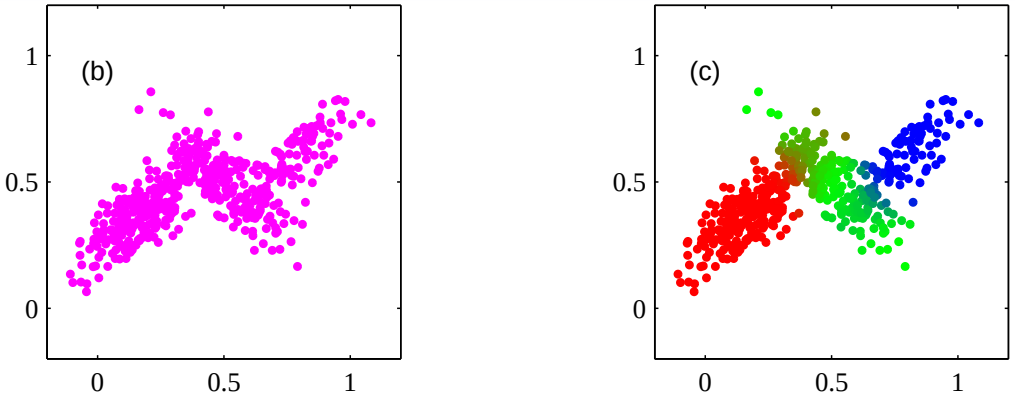
MoG Marginal over Observed Variables

- The marginal distribution $p(\boldsymbol{x}_n)$ for this model is:

$$\begin{aligned} p(\boldsymbol{x}_n) &= \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n) = \sum_{z_n} p(\boldsymbol{z}_n) p(\boldsymbol{x}_n | \boldsymbol{z}_n) \\ &= \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \end{aligned}$$

- A mixture of Gaussians

MoG Conditional over Latent Variable



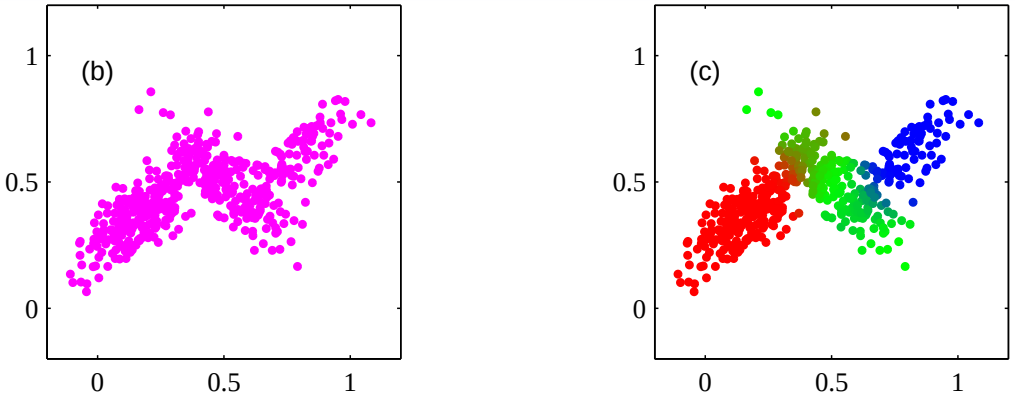
- To apply EM, need the conditional distribution $p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are the model parameters.
- It is denoted by $\gamma(z_{nk})$ and can be computed as: Exercise—how?

$$\gamma(z_{nk}) \equiv p(z_{nk} = 1|\boldsymbol{x}_n) = \frac{p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\boldsymbol{x}_n|z_{nj} = 1)}$$

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

- $\gamma(z_{nk})$ is the responsibility of component k for datapoint n

MoG Conditional over Latent Variable



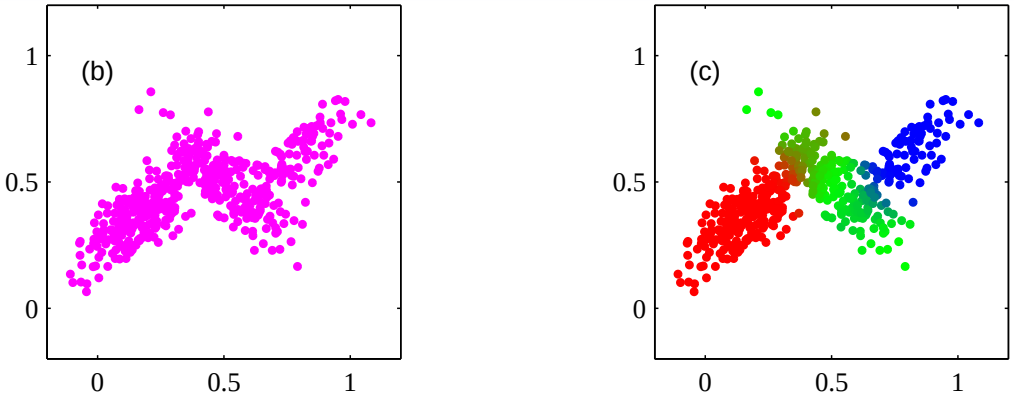
- To apply EM, need the conditional distribution $p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are the model parameters.
- It is denoted by $\gamma(z_{nk})$ and can be computed as: Exercise—how?

$$\gamma(z_{nk}) \equiv p(z_{nk} = 1|\boldsymbol{x}_n) = \frac{p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\boldsymbol{x}_n|z_{nj} = 1)}$$

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

- $\gamma(z_{nk})$ is the responsibility of component k for datapoint n

MoG Conditional over Latent Variable



- To apply EM, need the conditional distribution $p(z_{nk} = 1|\boldsymbol{x}_n, \boldsymbol{\theta})$ where $\boldsymbol{\theta}$ are the model parameters.
- It is denoted by $\gamma(z_{nk})$ and can be computed as: Exercise—how?

$$\begin{aligned}\gamma(z_{nk}) \equiv p(z_{nk} = 1|\boldsymbol{x}_n) &= \frac{p(z_{nk} = 1)p(\boldsymbol{x}_n|z_{nk} = 1)}{\sum_{j=1}^{K} p(z_{nj} = 1)p(\boldsymbol{x}_n|z_{nj} = 1)} \\ &= \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}\end{aligned}$$

- $\gamma(z_{nk})$ is the responsibility of component k for datapoint n

Expectation-Maximization for Gaussian Mixtures

- Initialize parameters, then iterate:
  - **E step**: Calculate responsibilities using current parameters

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

  - **M step**: Re-estimate parameters using these $\gamma(z_{nk})$

$$\begin{aligned} N_k &= \sum_{n=1}^{N} \gamma(z_{nk}) \\ \boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \\ \boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \\ \pi_k &= \frac{N_k}{N} \end{aligned}$$

- Think of N_k as effective number of points in component k.

Expectation-Maximization for Gaussian Mixtures

- Initialize parameters, then iterate:
  - **E step**: Calculate responsibilities using current parameters

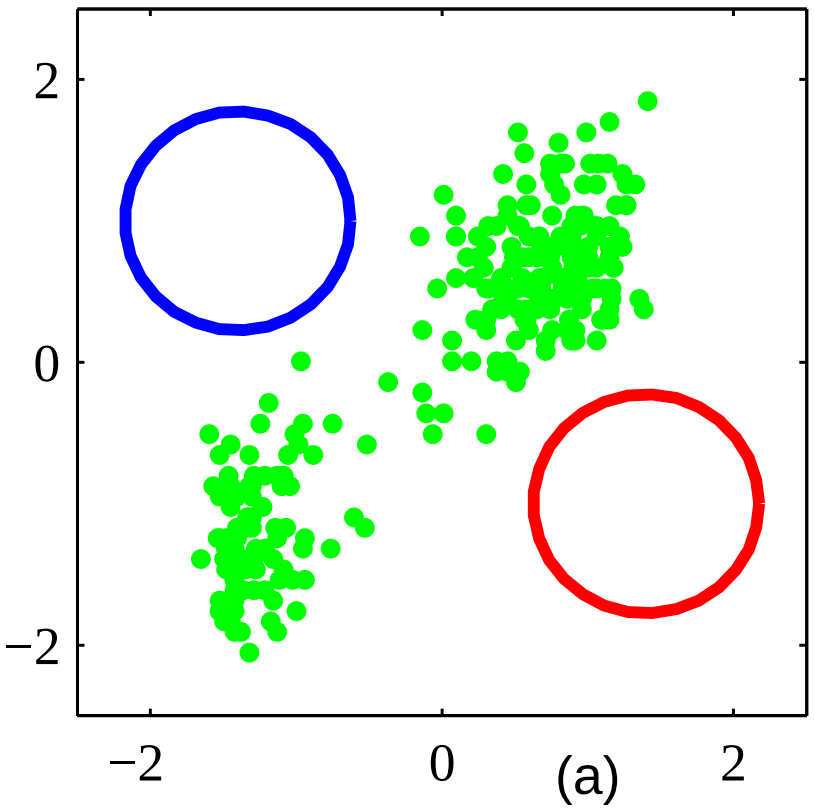
$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

  - **M step**: Re-estimate parameters using these $\gamma(z_{nk})$

$$\begin{aligned} N_k &\equiv \sum_{n=1}^{N} \gamma(z_{nk}) \\ \boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \boldsymbol{x}_n \\ \boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T \\ \pi_k &= \frac{N_k}{N} \end{aligned}$$

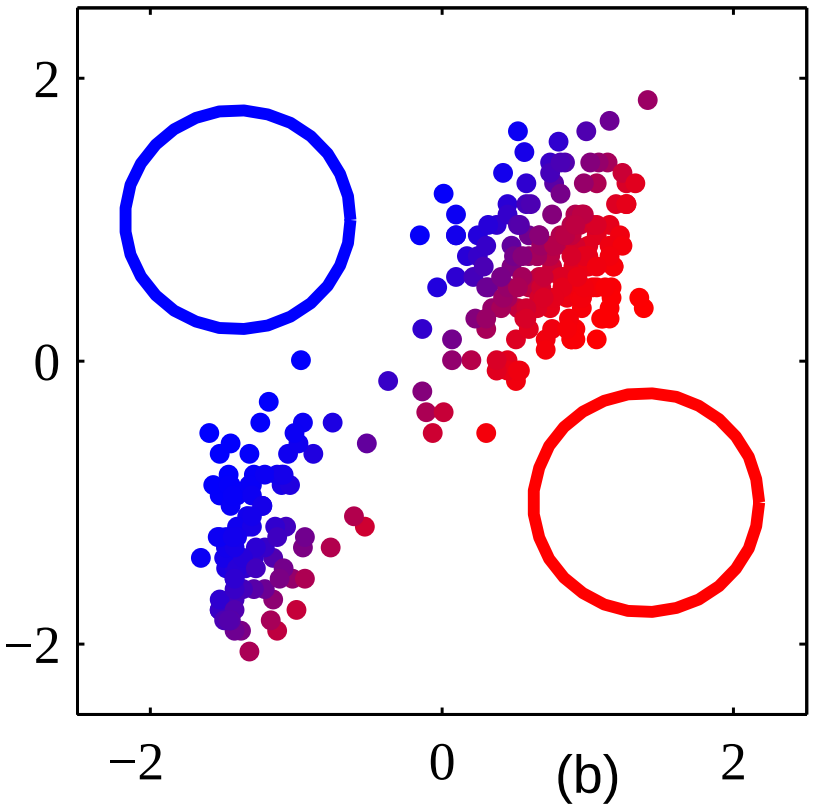
- Think of N_k as effective number of points in component k.

MoG EM - Example



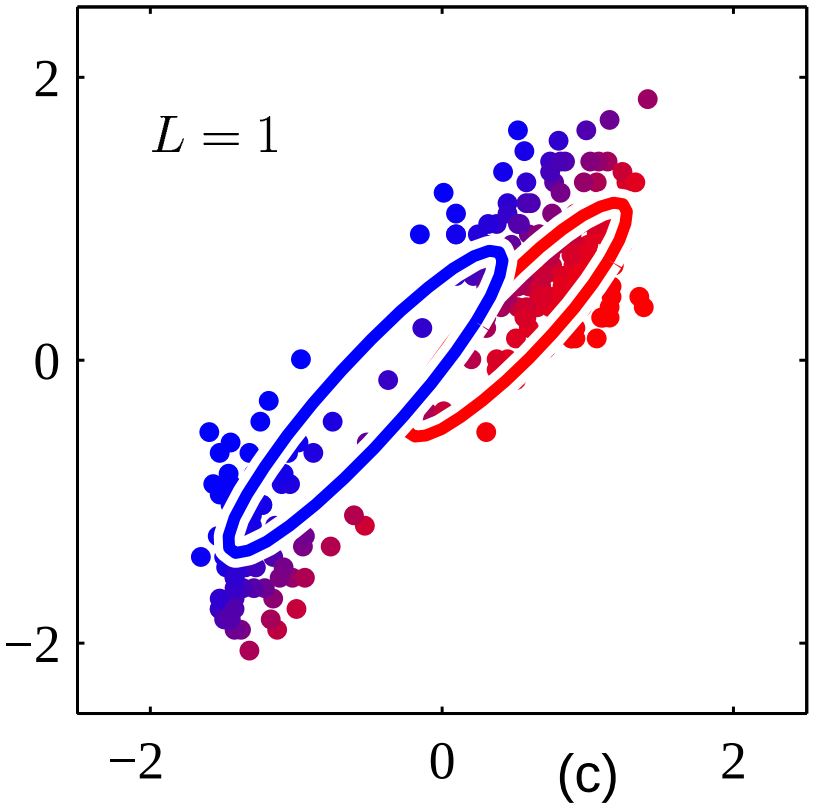
- Same initialization as with K-means before
  - Often, K-means is actually used to initialize EM

MoG EM - Example



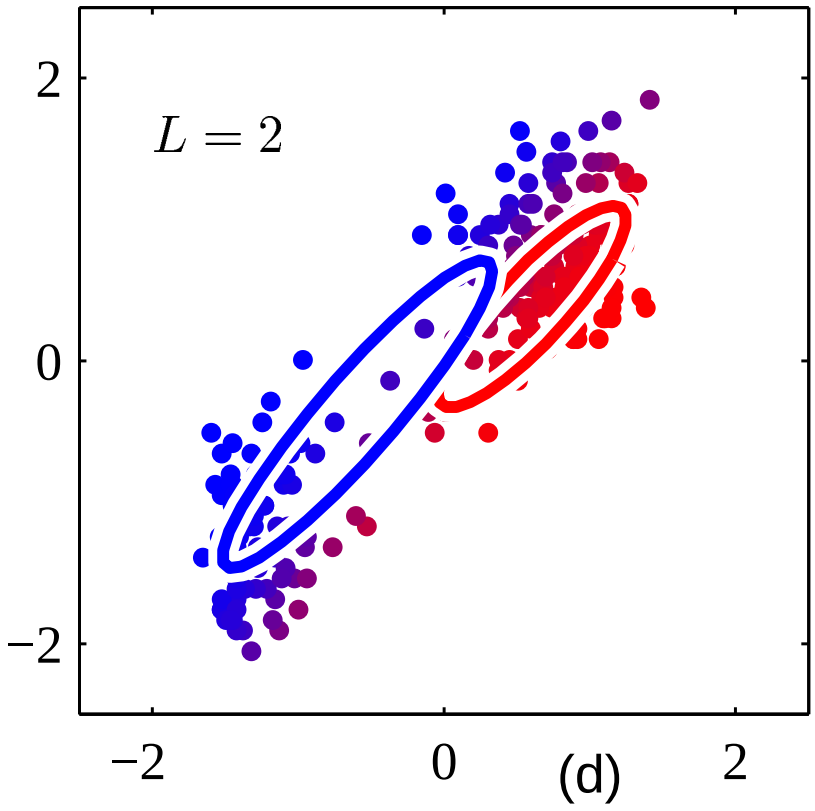
- Calculate responsibilities $\gamma(z_{nk})$

MoG EM - Example



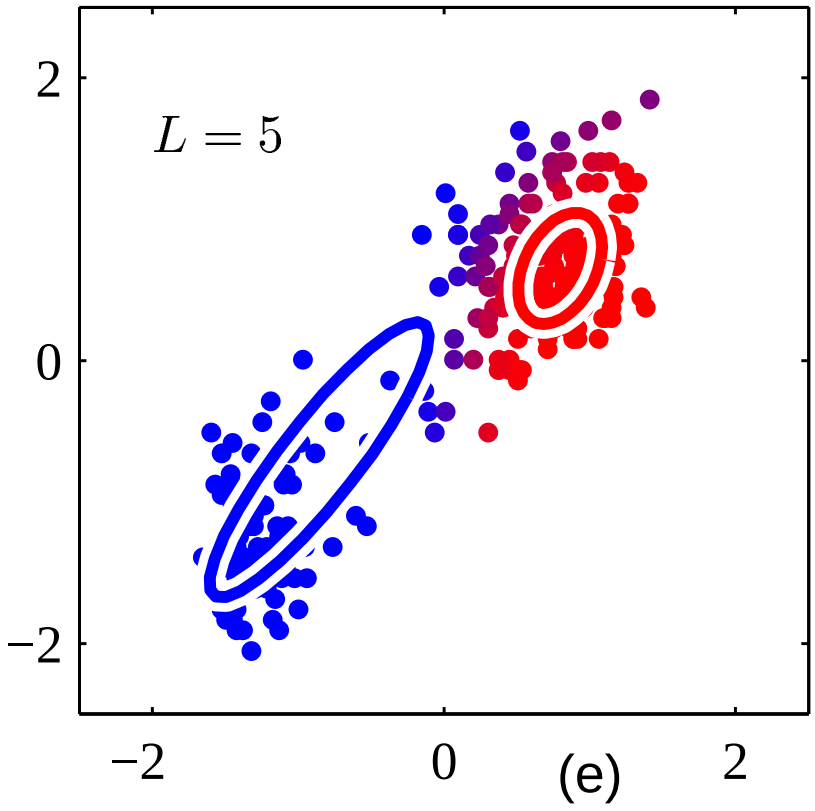
- Calculate model parameters $\{\pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$ using these responsibilities

MoG EM - Example



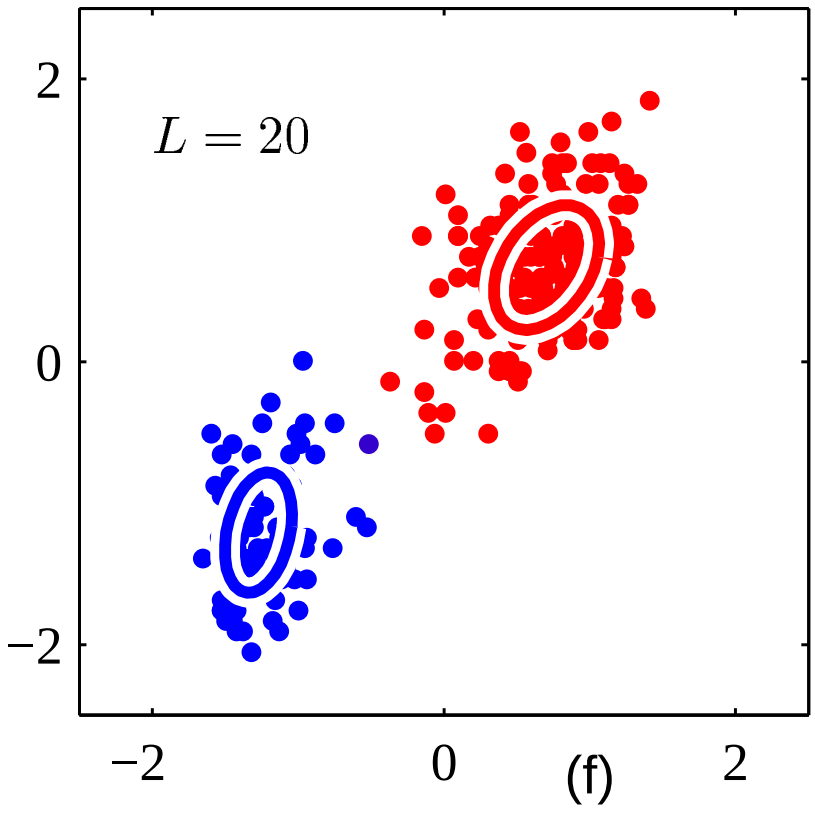
- Iteration 2

MoG EM - Example



- Iteration 5

MoG EM - Example



- Iteration 20 - converged

EM - Summary

- EM finds local maximum to likelihood

$$p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$$

- Iterates two steps:
  - **E step** calculates the distribution of the missing variables $\boldsymbol{Z}$
  - (Hard EM "fills in" the variables).
  - **M step** maximizes expected complete log likelihood (expectation wrt **E step** distribution)

EM and Maximum Likelihood

1. The EM procedure is guaranteed to increase at each step, the data log-likelihood $\ln p(\boldsymbol{X}|\boldsymbol{\theta}) = \sum_{\boldsymbol{Z}} \ln p(\boldsymbol{X}, \boldsymbol{Z}|\boldsymbol{\theta})$.
2. Therefore converges to *local log-likelihood maximum.* More theoretical analysis in Bishop.

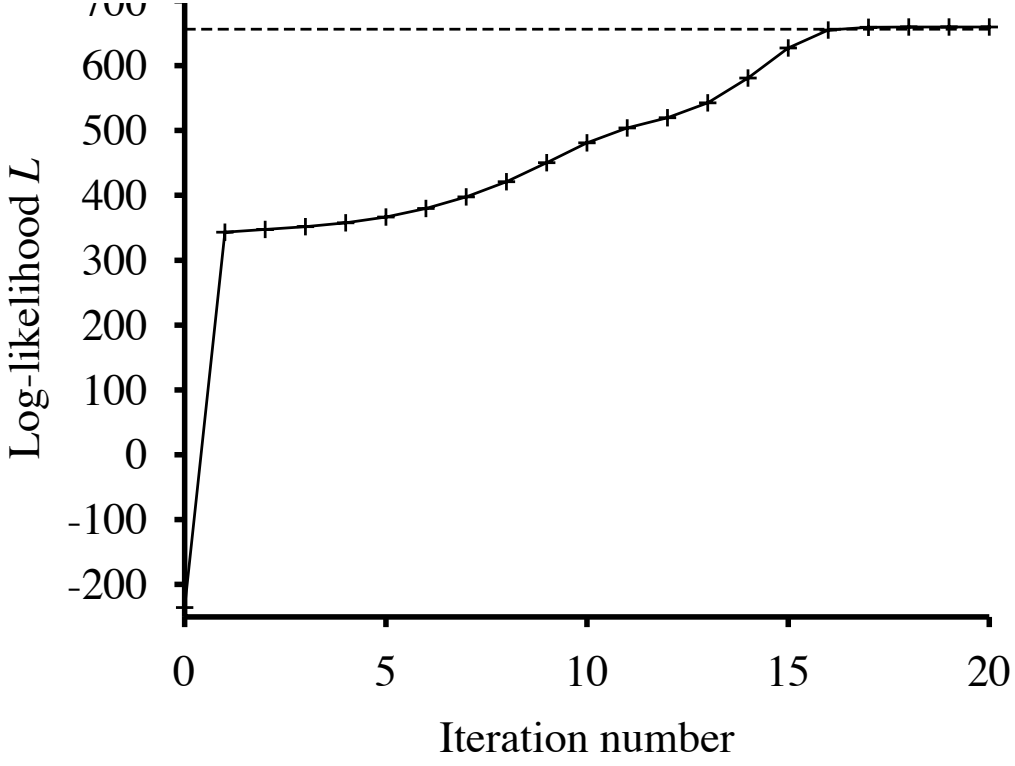


Fig. Russell and Norvig 20.12

Conclusion

- Readings: Bishop Ch. 9.1, 9.2, 9.4
- K-means clustering
- Gaussian mixture model: latent variables as part of a model for how data are generated.
- Expectation-maximization, a general method for learning parameters of models when not all variables are observed