Clustering & Mixture Models

CS 534: Machine Learning

Slides adapted from David Sontag, Luke Zettlemoyer, Carlos Guestrin, Andrew Moore, Dan Klein, Ryan Tibshirani, Trevor Hastie, Rob Tibshirani, Nicholas Ruozzi, and Vibhav Gogate

Unsupervised Learning: Motivation

- What if we don't have a response variable?
  - Cases where it is easier to obtain unlabeled data than labeled data
- What if we have high-dimensional data?
- Is there an informative way to visualize this data?
- Can we discover subgroups amongst these variables?

Clustering: Overview

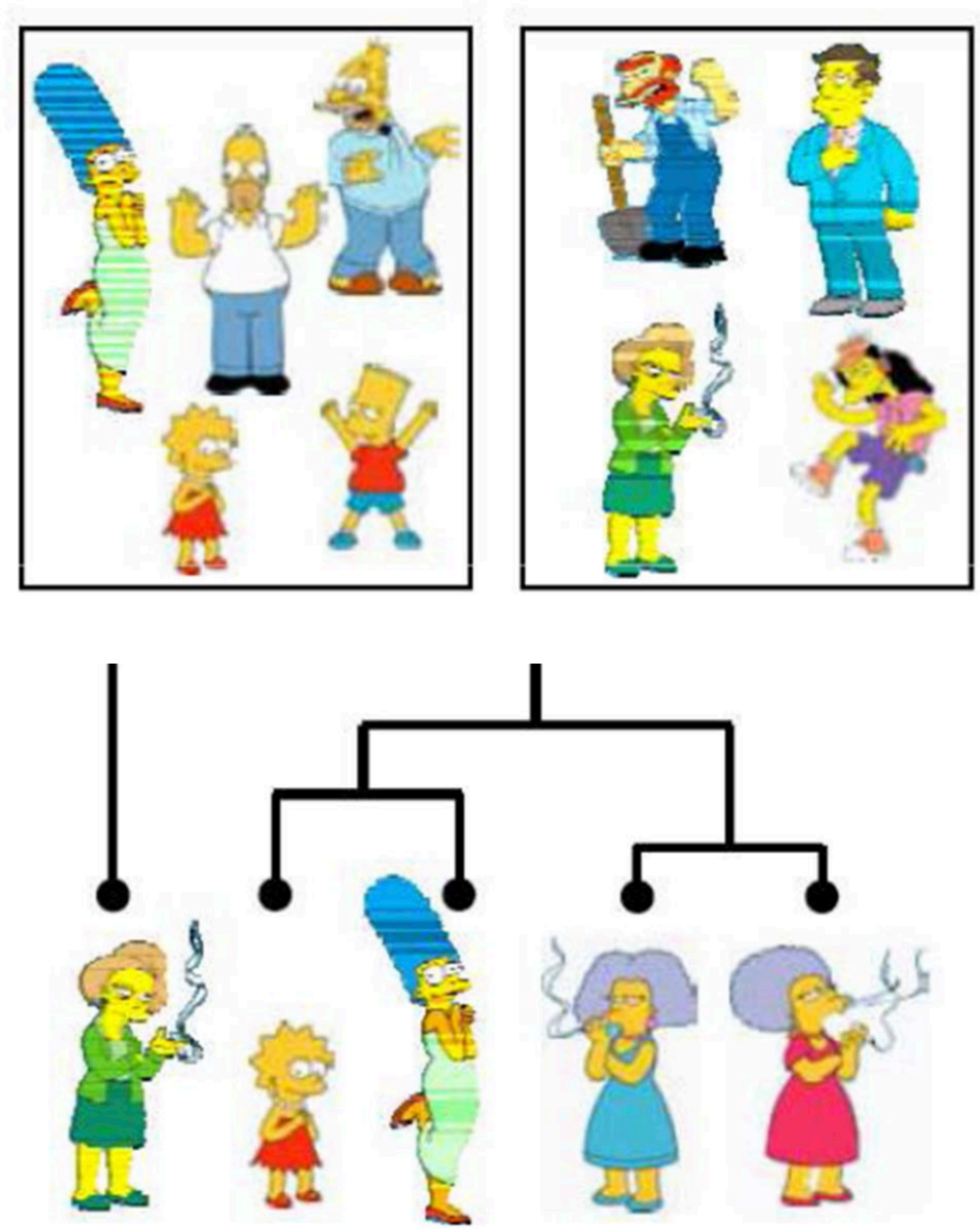
- Divide data into groups (clusters) — points in any one group are more ‘similar’ to each other than points outside the group
- Why?
  - Summarize: Reduced representation of the full set
  - Discovery: Looking for new insights into the structure the data

Dimensionality Reduction vs Clustering

- Dimensionality reduction (e.g., PCA) looks for a low-dimensional representation of the observations
- Clustering looks for homogenous subgroups amongst observations

Clustering Algorithms

- Partition algorithms
  - K-means
  - Gaussian mixture models
- Hierarchical algorithms
  - Agglomerative
  - Divisive



Dissimilarity & Within-Cluster Scatter

- Dissimilarity can be thought of as the distance between two points
  - Example: Euclidean distance

$$d(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j||_2^2$$

- Within-cluster scatter: How far away points are assigned to the same cluster

$$W = \frac{1}{2} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i,j \in S_k} d(\mathbf{x}_i, \mathbf{x}_j)$$

K-means Clustering

K-means

- Pick an initial set of k means (usually at random)
- Repeat until no points' assignment changes
  - Partition data points, assigning each data point to the closest cluster mean
  - Update the k cluster means so that the i^{th} mean is the average of all data points assigned to cluster i

Example: K-means

- Pick K random points as cluster centers
- This example uses K = 2

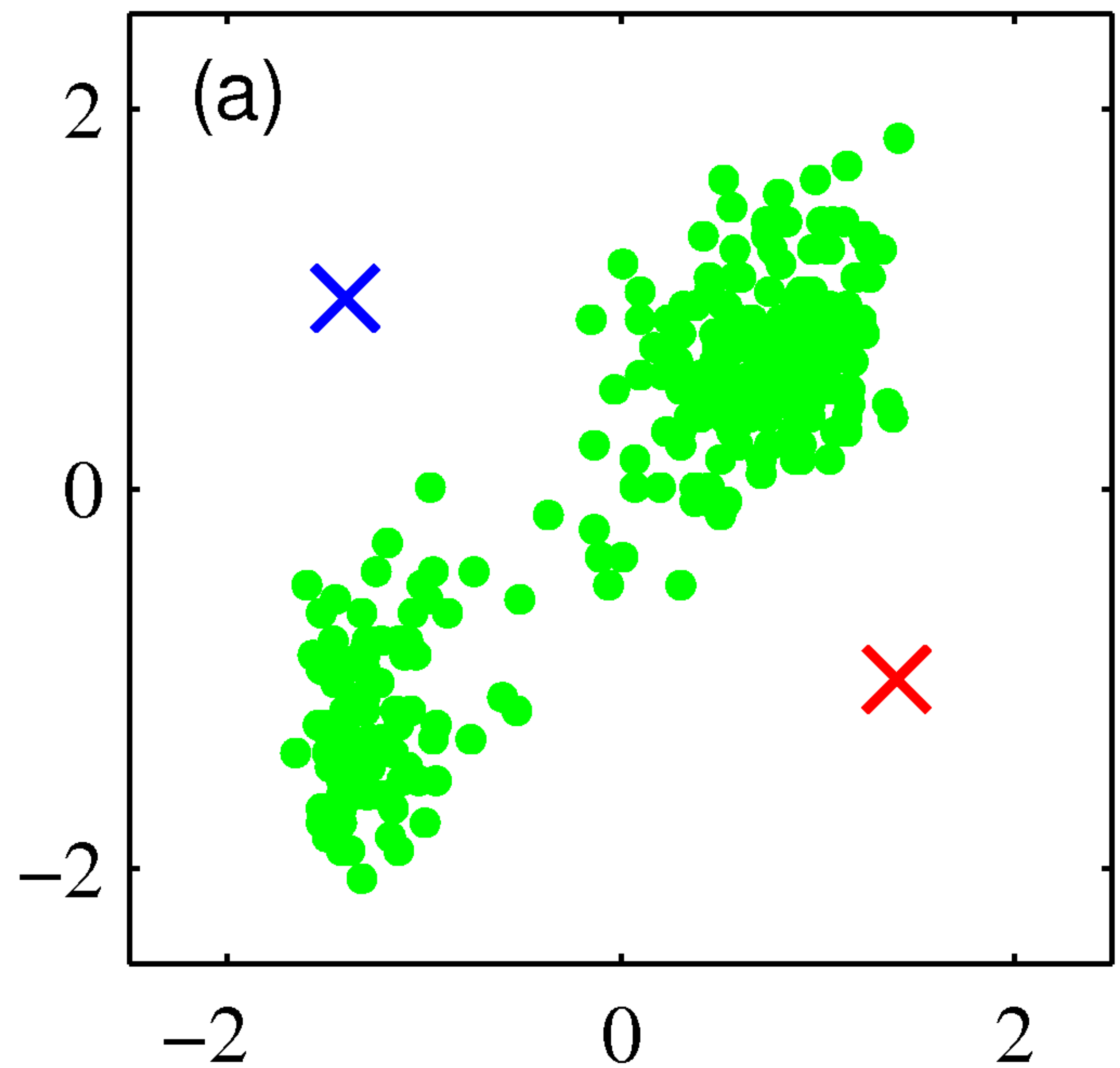
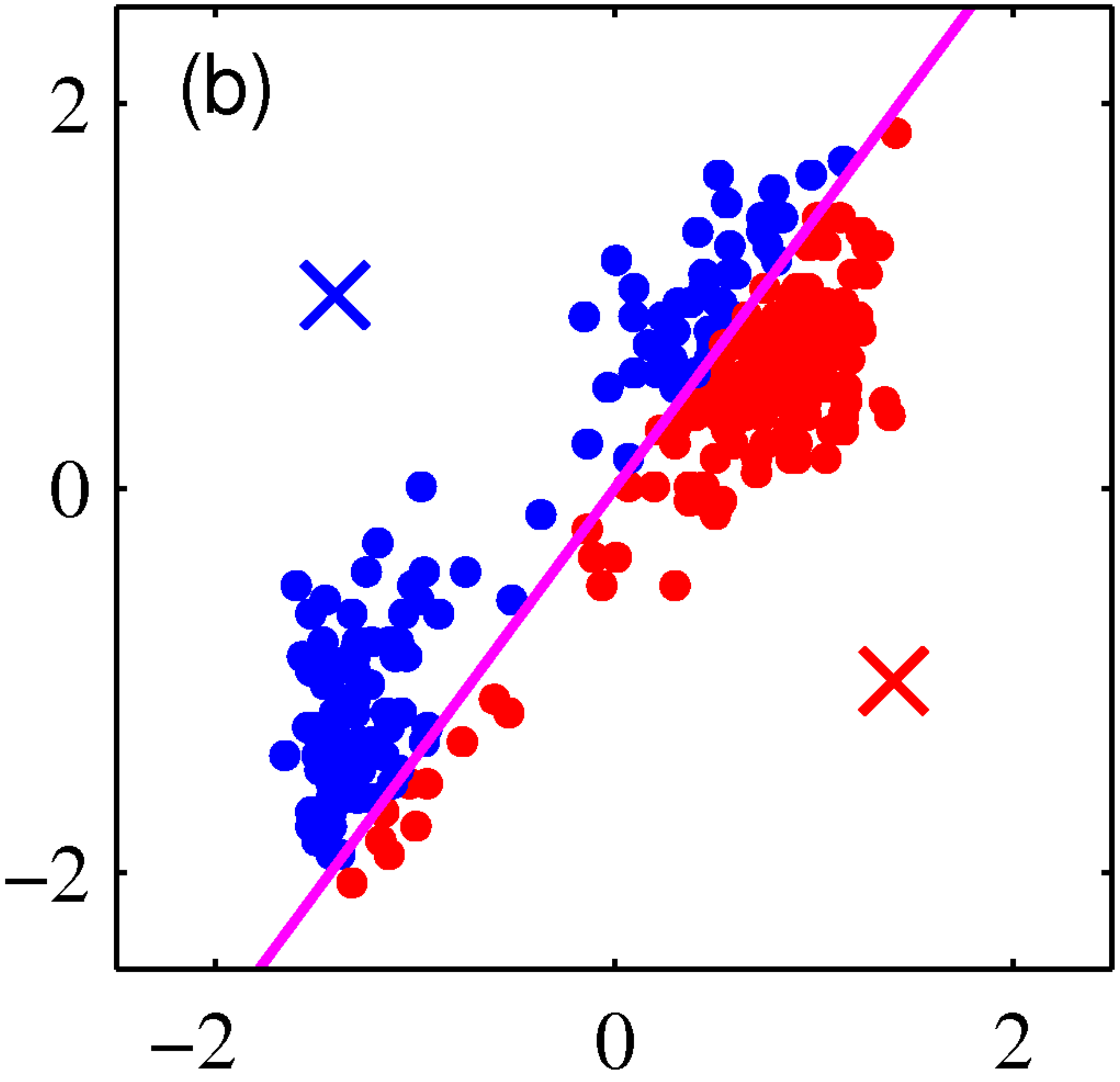


Figure 9.1 (Bishop)

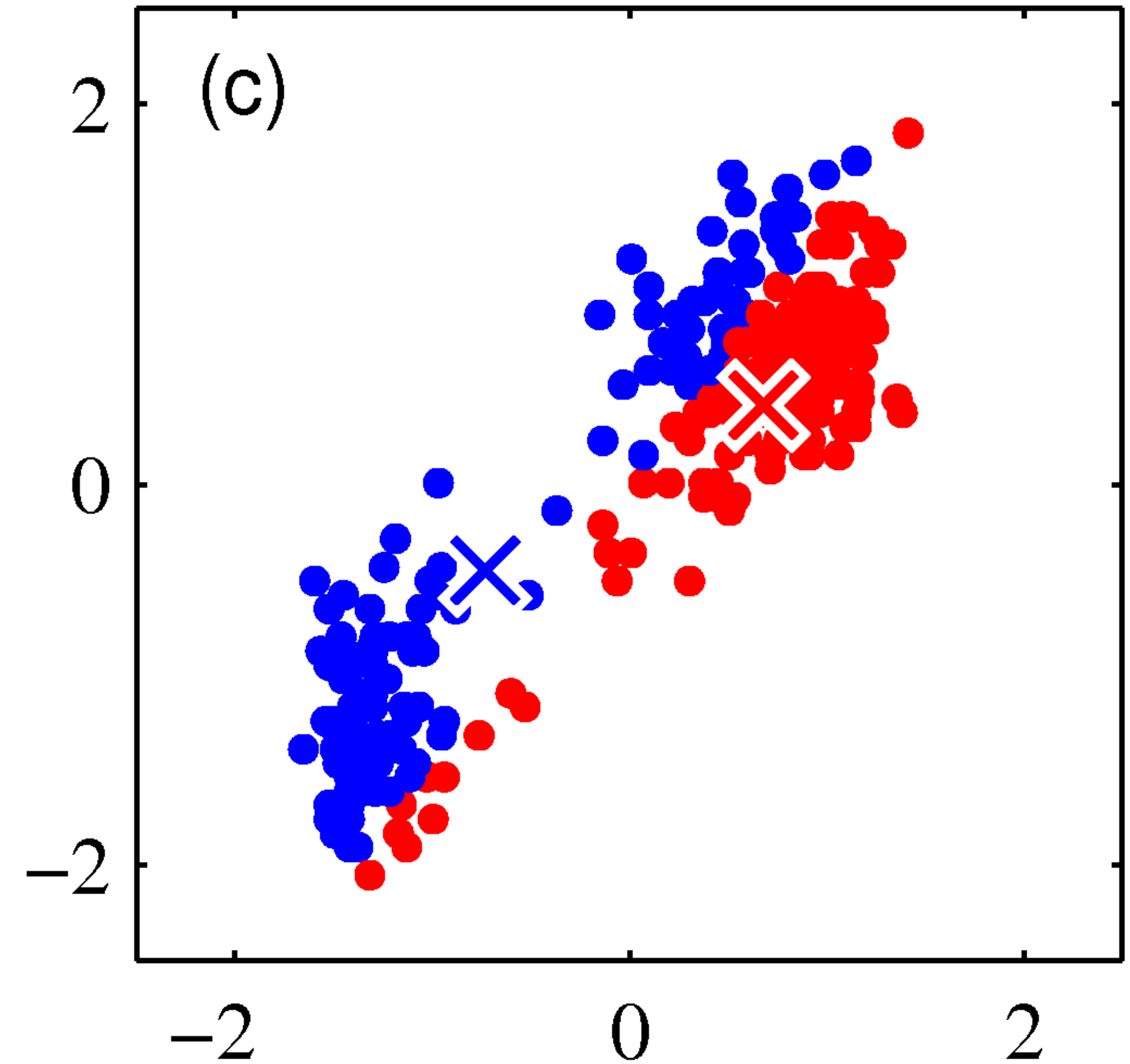
Example: K-means

- Iterative step 1
  - Assign each point to its closest means

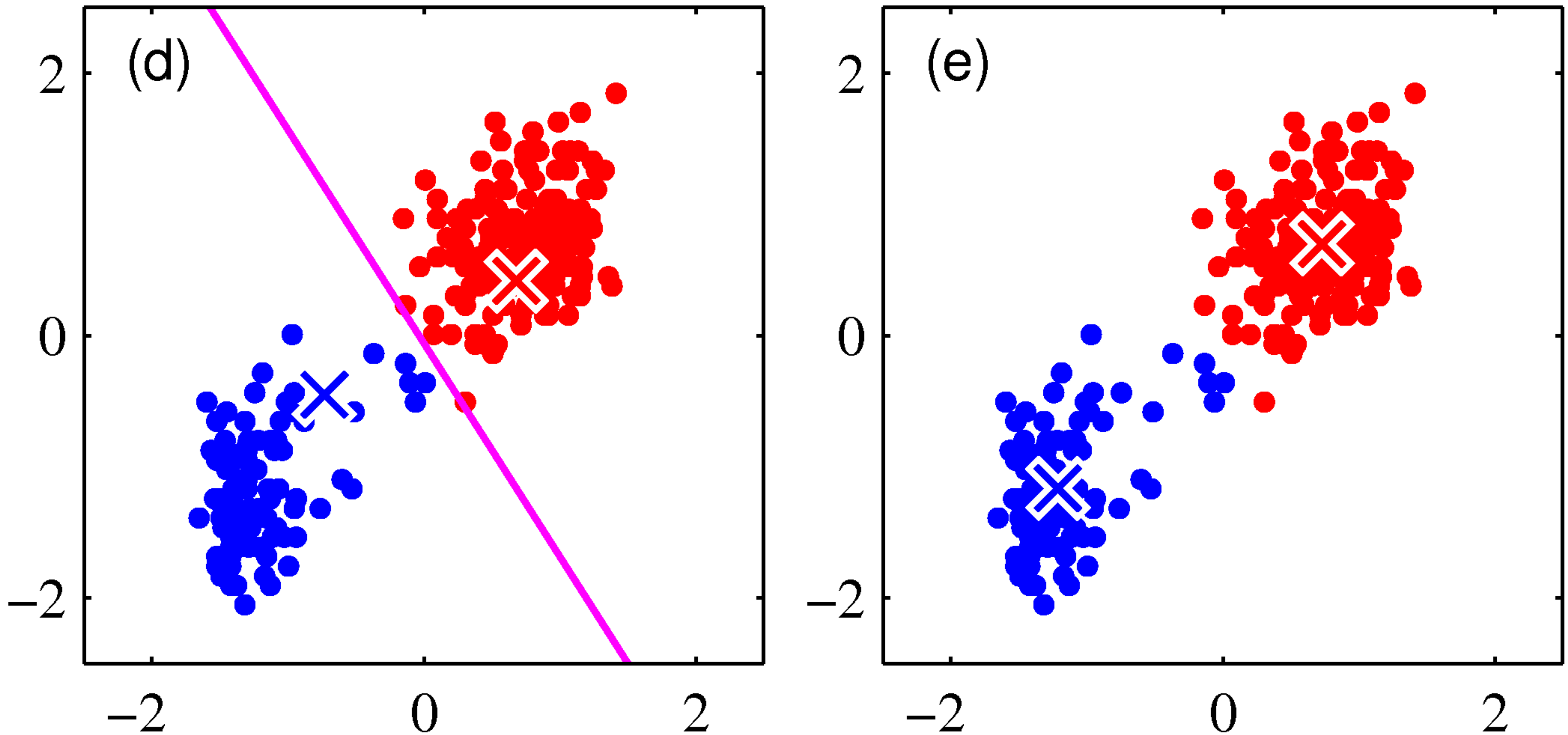


Example: K-means

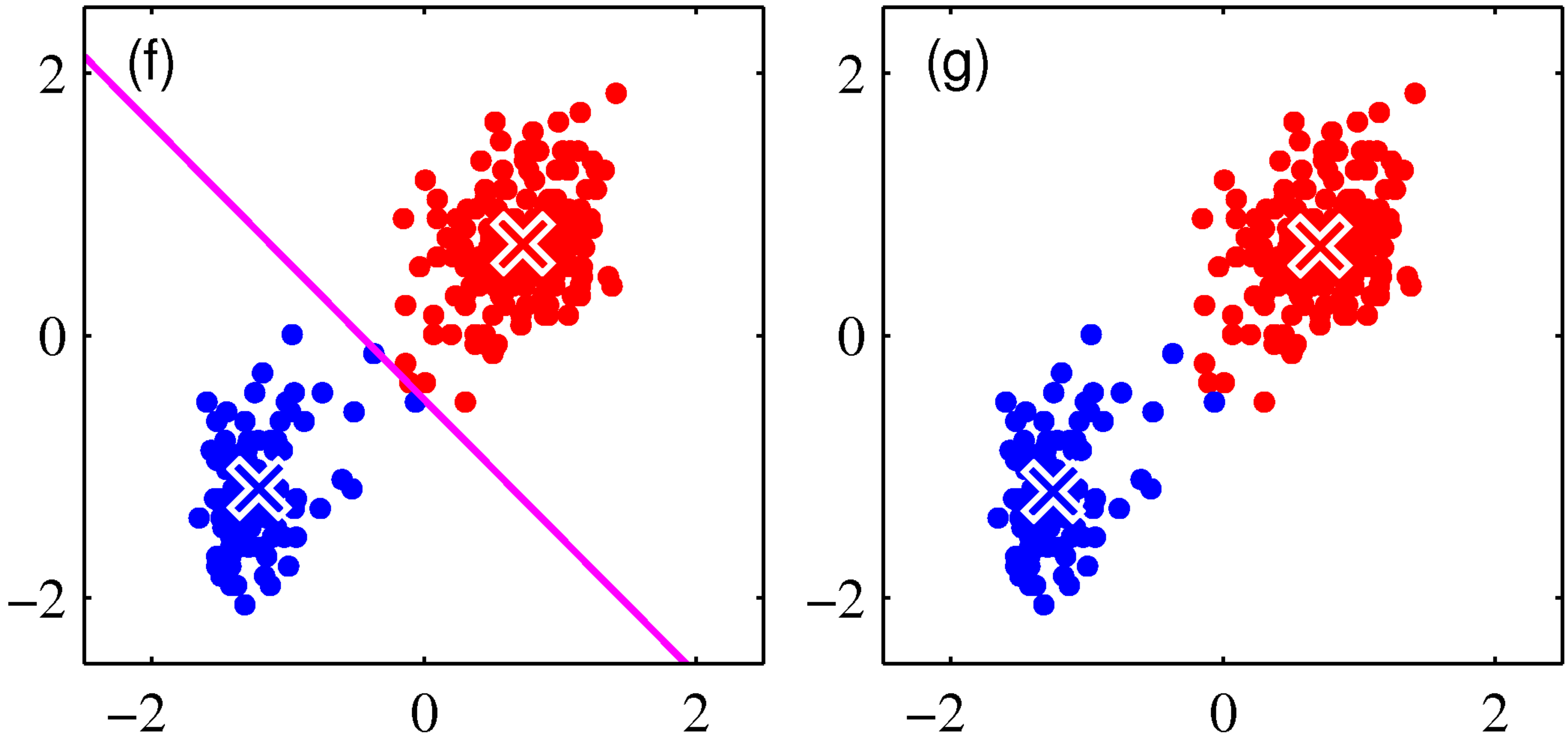
- Iterative step 1
  - Update cluster means based on the new points



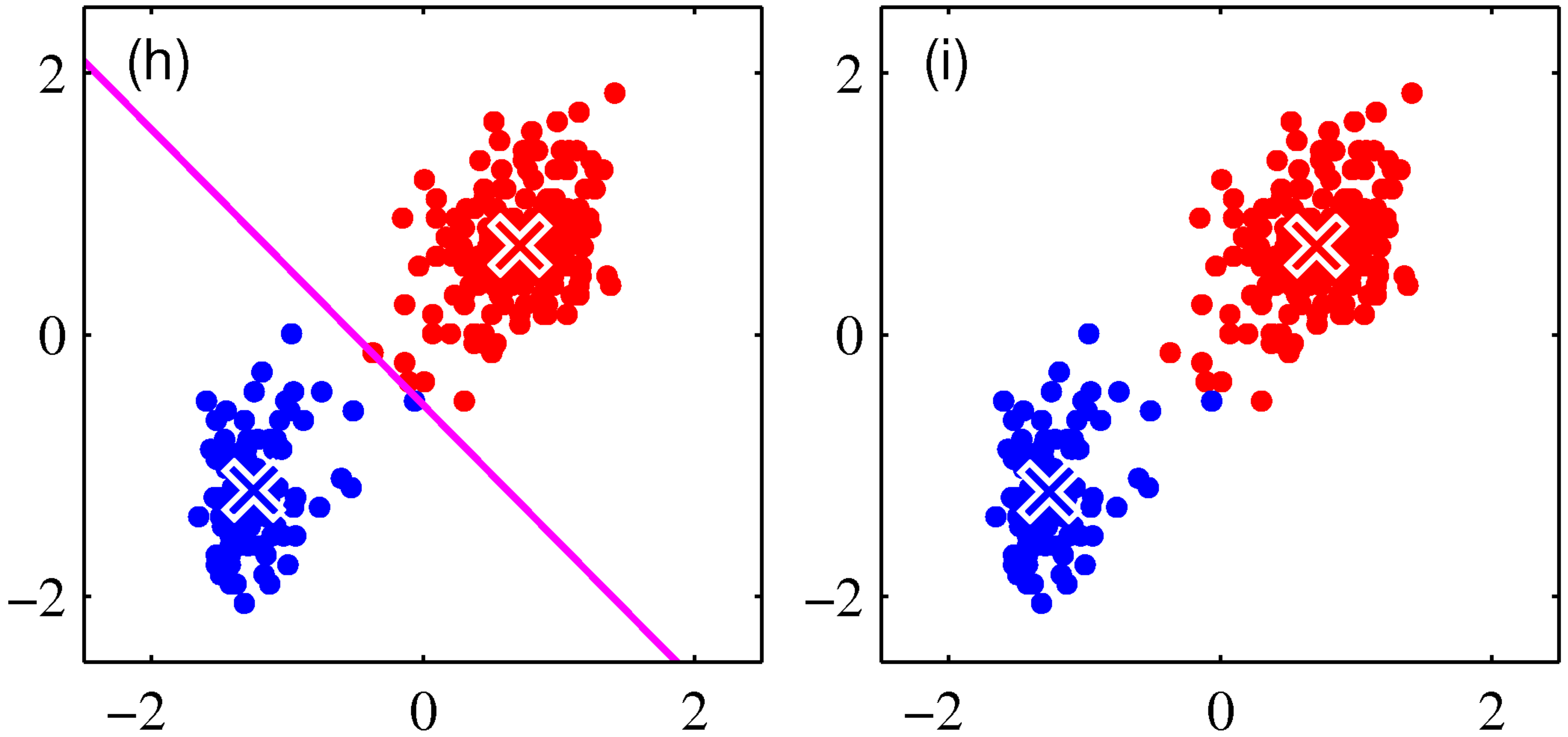
Example: K-means



Example: K-means



Example: K-means



Converged so stop!

Example: K-means for Segmentation

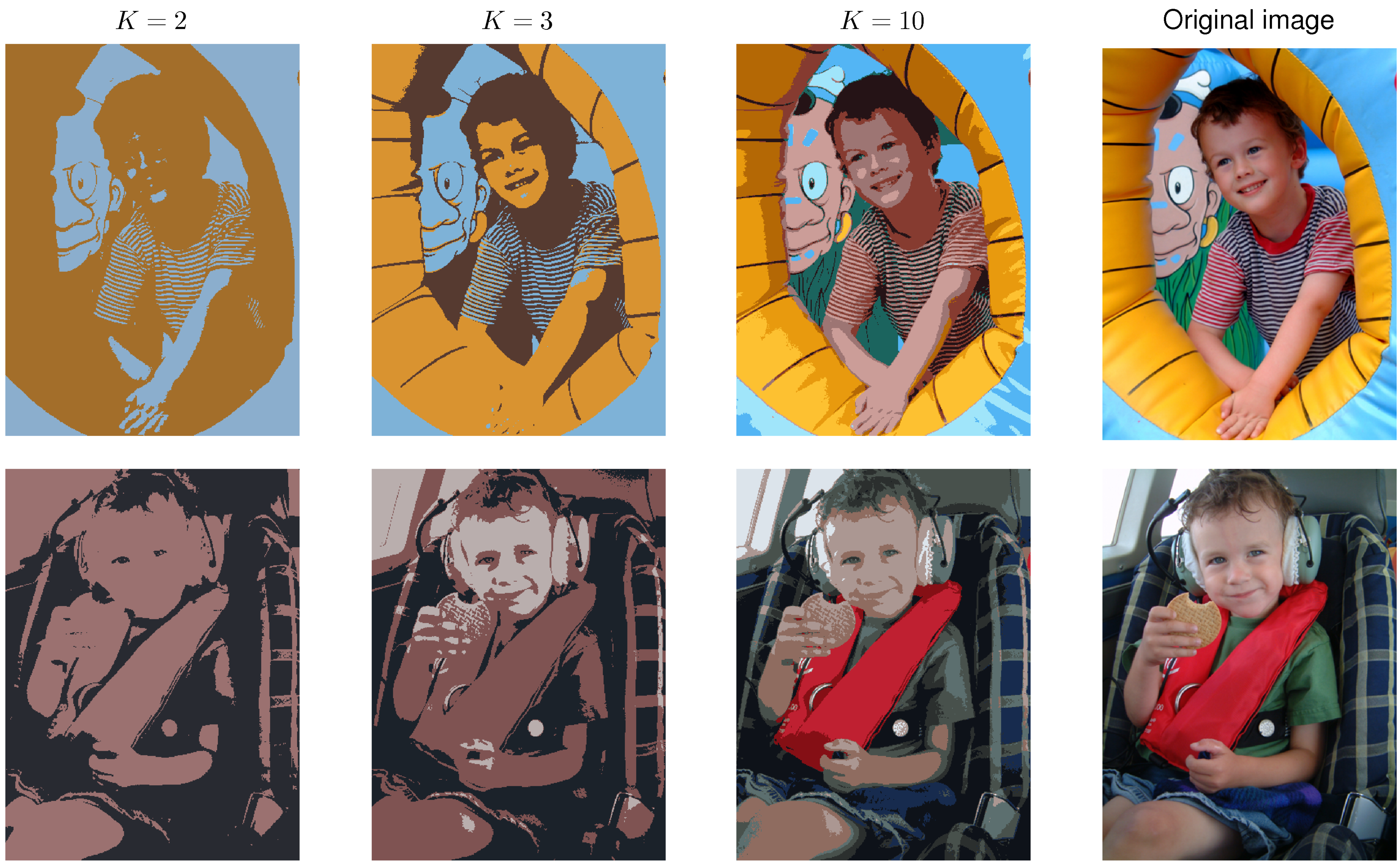


Figure 9.3 (Bishop)

K-Means: Optimization

- Minimize the distance of each point to the mean of the cluster/partition that contains it

$$\min_{S_1,\cdots,S_k} \sum_{i=1}^{k} \sum_{j \in S_i} ||\mathbf{x}_j - \boldsymbol{\mu}_i||_2^2$$

- Exactly minimizing this problem is NP-hard even for k = 2
- Solve via block coordinate descent / alternating minimization
- Not convex function — can get stuck in local minima

K-Means: Optimization

- Objective

$$\min_{S_1, \cdots, S_k} \sum_{i=1}^{k} \sum_{j \in S_i} ||\mathbf{x}_j - \boldsymbol{\mu}_i||_2^2$$

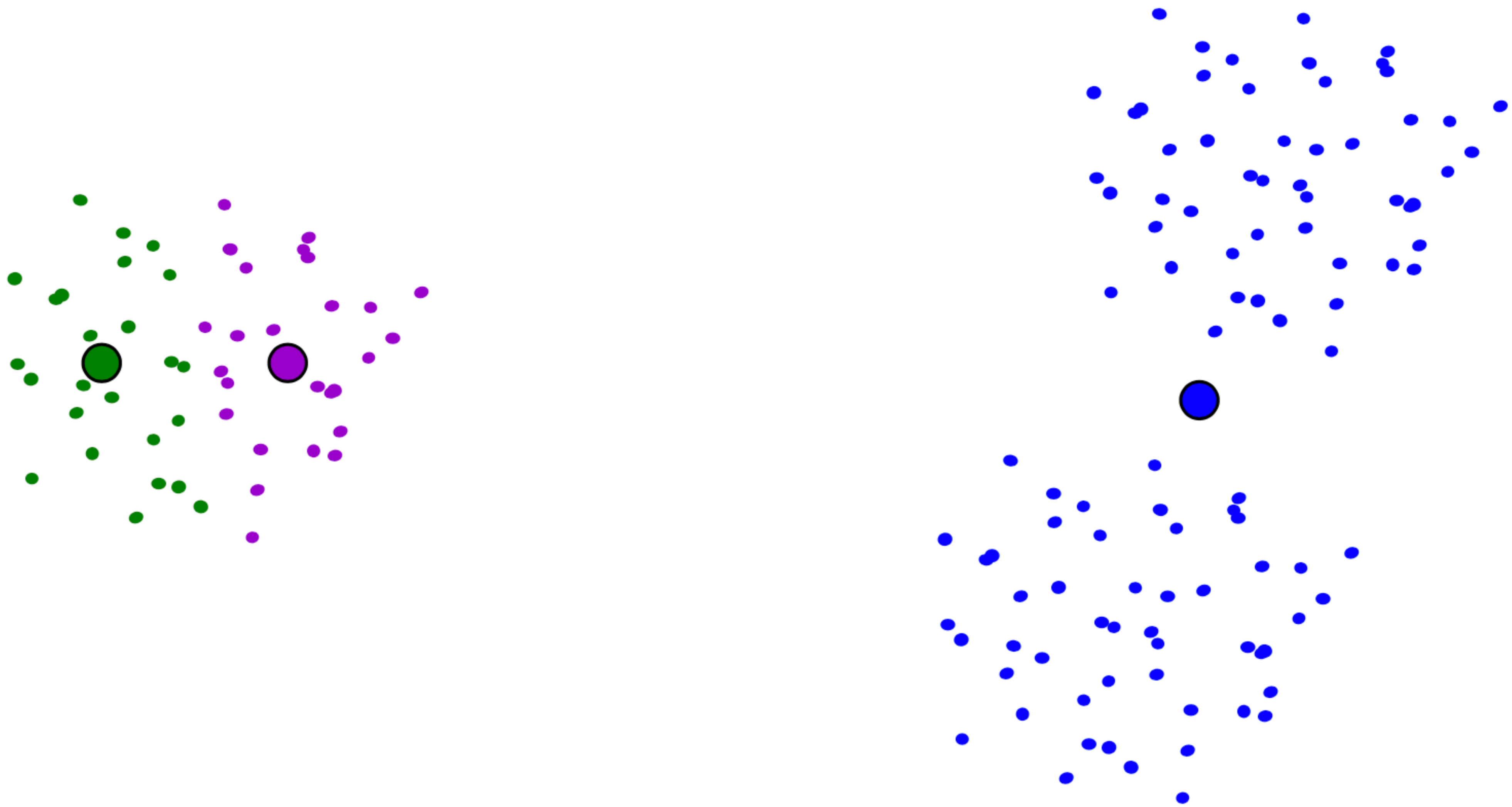
- Step 1: fix means, optimize assignments

$$C_j = \text{argmin}_i ||\mathbf{x}_j - \boldsymbol{\mu}_i||_2^2 \Rightarrow f(\mathbf{x}, S, \boldsymbol{\mu}) \text{ decreases}$$

- Step 2: fix assignment, optimize means

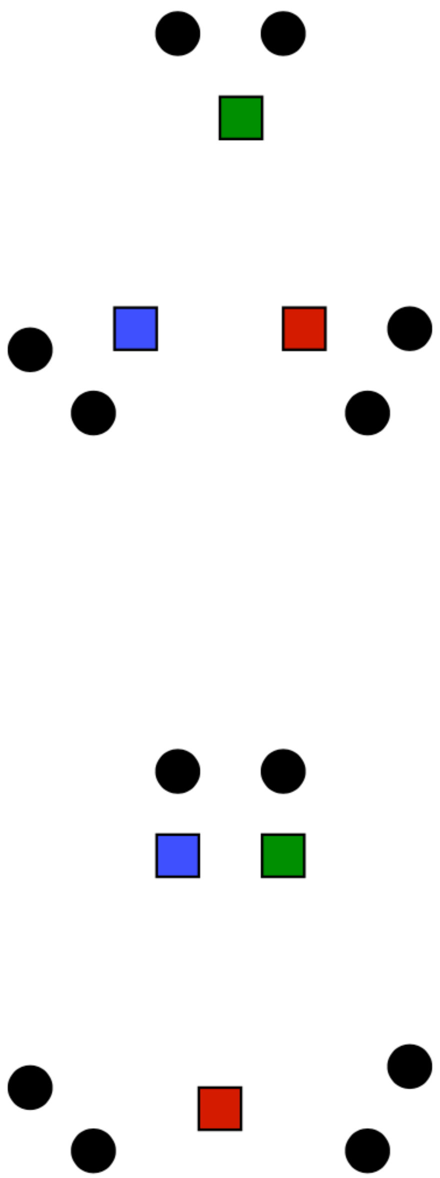
$$\min_{\mu} \sum_{i=1}^{k} \sum_{j \in S_i} ||\mathbf{x}_j - \boldsymbol{\mu}_i||_2^2 \Rightarrow \mu_i = \frac{1}{S_i} \sum_{j \in S_i} \mathbf{x}$$

k-Means: Local Optima



K-means: Intialization

- K-means algorithm is a heuristic
  - Requires initial means
  - What could go wrong?



Various schemes to prevent this: initialization heuristics, variance-based split/merge

k-Means: Initialization

K-means: Choice of K

- How to pick “best” k?
- Want to find k to pick out interesting clusters, but not to overfit data points
    - Large k doesn’t necessarily mean we will get interesting clusters
    - Small k can result in large clusters than can be broken down futher

K-means: Properties

- Guaranteed to converge in a finite number of iterations
    - Not to global optimum
- Running time (per iteration):
    - Assign data points to closest cluster center: O(kN)
    - Change cluster center to average of assigned points: O(N)

Hierarchical Clustering

Hierarchical Clustering

- K-means clustering requires K to be specified — what if we want it to be flexible?
- K-means results depends heavily on initialization of cluster centers — what if we want consistent results?
- Hierarchical clustering produces consistent results without needing initial starting positions using just pairwise dissimilarities between points

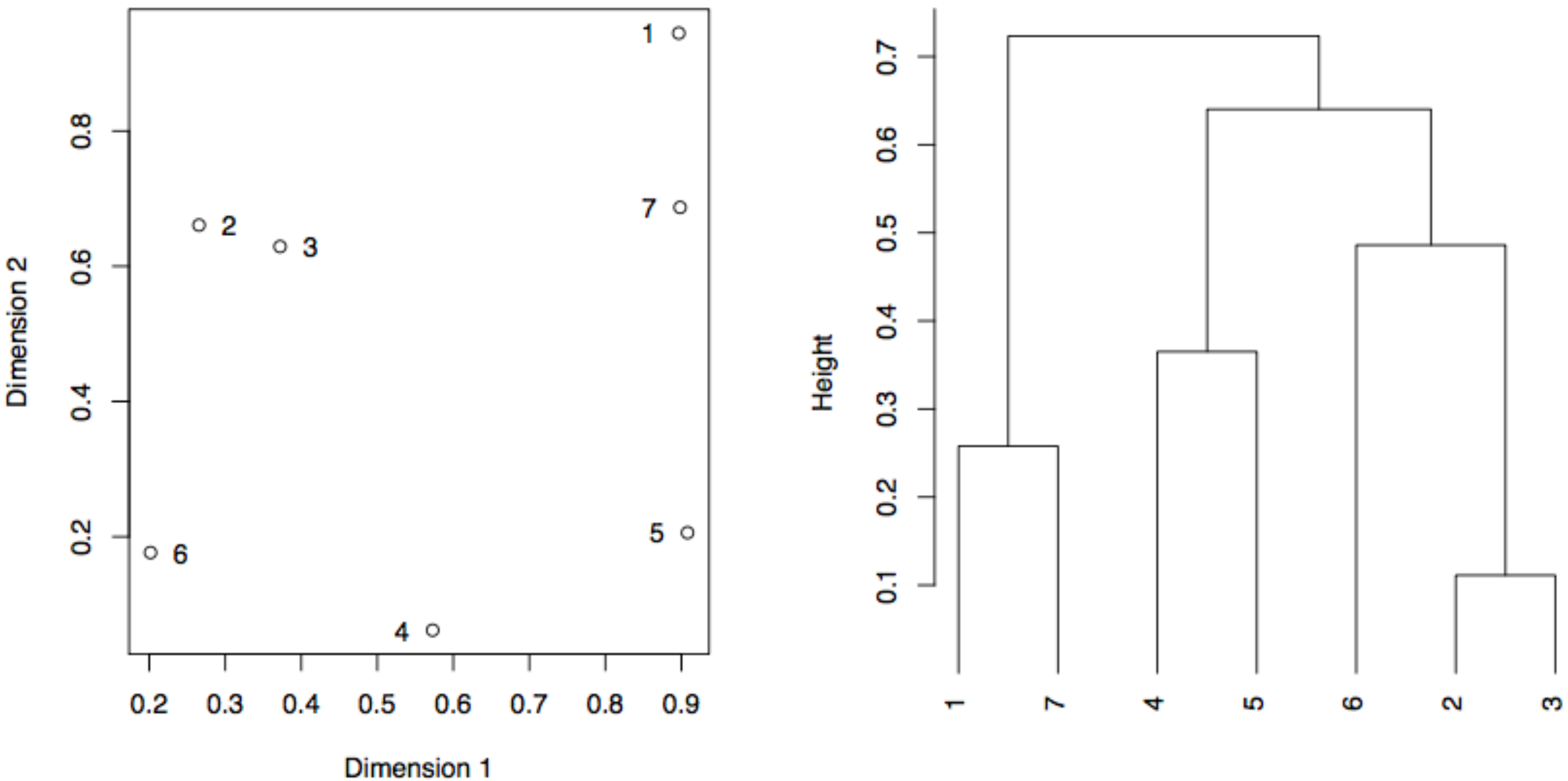
Hierarchical Clustering: Algorithms

- Agglomerative: bottom up
  - Start with all points in their own group
  - Merge two groups that have the smallest dissimilarity until there is one cluster
- Divisive: top-down
  - Start with all points in one cluster
  - Split group into two resulting in biggest dissimilarity until each point in own group

Dendrogram

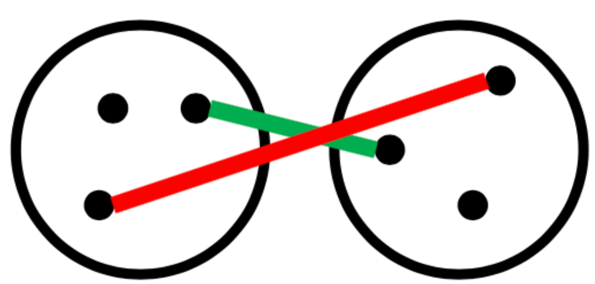
- Convenient graphic to display the hierarchical sequence of clustering assignments
- A tree where
  - Each node represents a group
  - Each leaf node contains a single point
  - Root node contains whole data set
  - Each internal node has two children

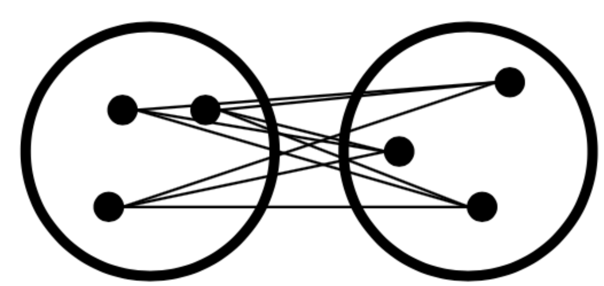
Example: Dendrogram



Linkage

- Linkage: Function d(G, H) takes two groups G and H and returns a dissimilarity score between them
- Choice of linkage determines how we measure dissimilarity between group of points
- Given a particular linkage — merge groups such that d(G,H) is smallest


Closest / farthest pair


Average of all pairs

Linkage: Types

Linkage	Description	Equation
Single	Minimal inter-cluster dissimilarity (smallest dissimilarity between two points in G and H)	$\min_{i \in G, j \in H} d_{ij}$
Complete	Maximal inter-cluster dissimilarity (largest dissimilarity between two points in G and H)	$\max_{i \in G, j \in H} d_{ij}$
Average	Mean inter-cluster dissimilarity (average dissimilarity between two points in G and H)	$\frac{1}{\lvert G \rvert \lvert H \rvert} \sum_{i \in G, j \in H} d_{ij}$
Ward	Minimize total within-cluster variance	Lance-Williams algorithm

Example: Linkage

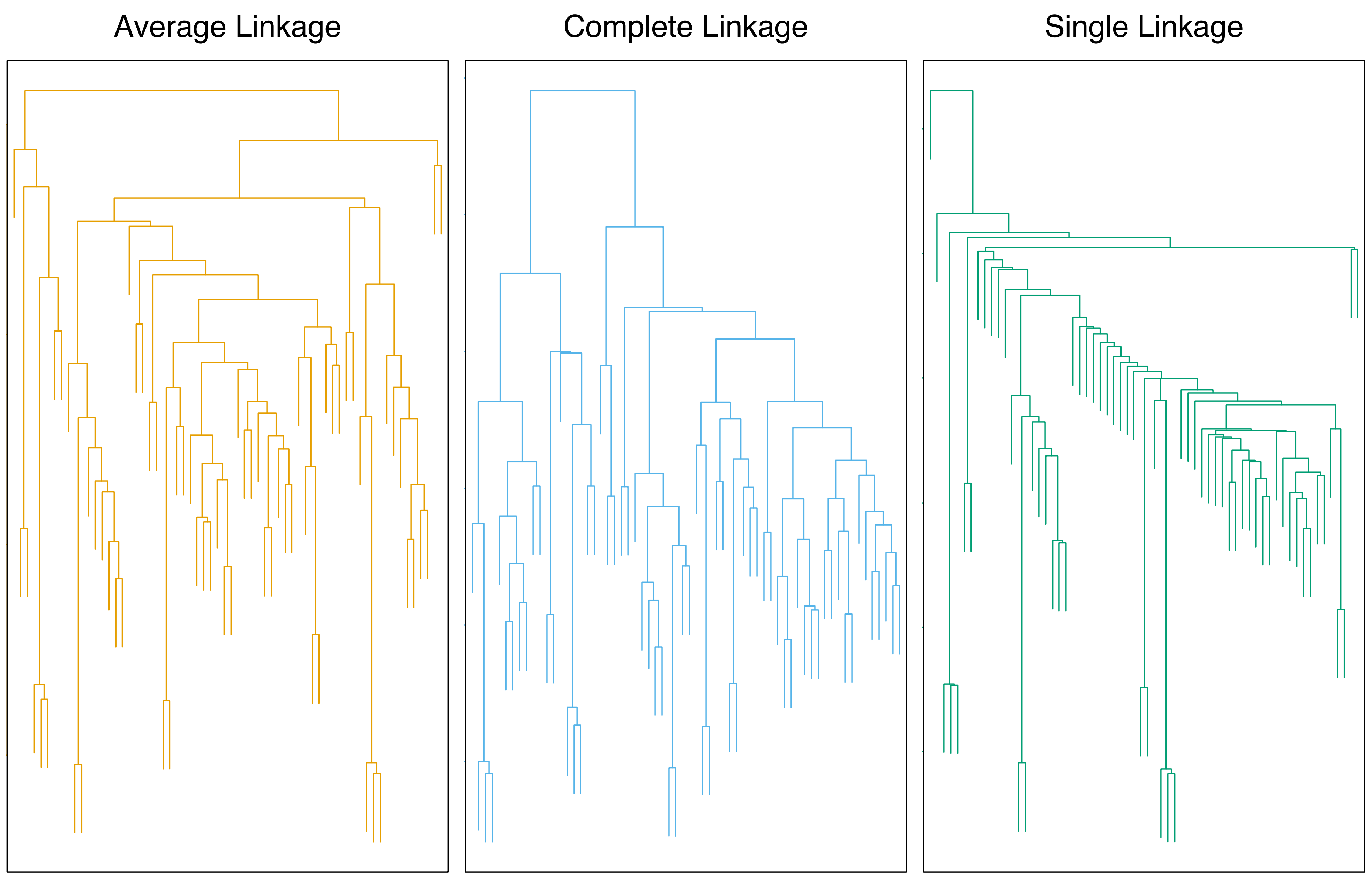


Figure 14.13 (Hastie et al.)

Linkage: Practical Considerations

- Single linkage suffers from chaining: Clusters can be too spread out and not compact enough
- Complete linkage suffers from crowding: Clusters are compact but not far enough apart

Linkage: Practical Considerations

- Average linkage balances both: Clusters tend to be relatively compact and far apart
  - Less interpretability when tree is cut at length h
  - Results can change with monotone increasing transformation of dissimilarites

Revisiting K-Means

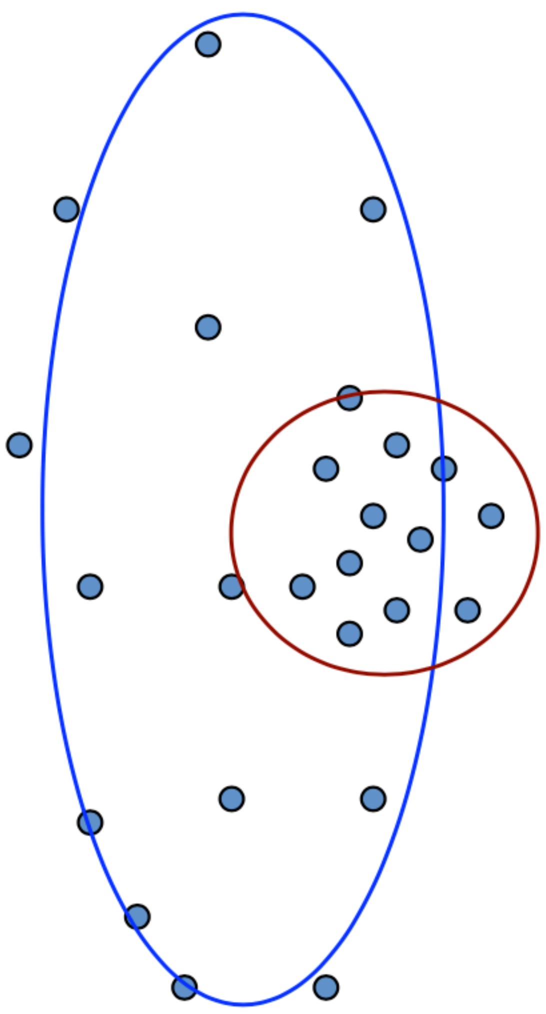
- Assumes that each instance is given a "hard" assignment to exactly one cluster
- Does not allow in cluster membership or for any instance to belong to more than one cluster
    - What if a data point lies roughly midway between two cluster centers?
- Soft clustering: Gives probabilities that an instance belongs to a set of clusters

Probabilistic Clustering

- Use probabilistic model: Allows overlaps, clusters of different sizes, etc
- Generative model: Can tell generative story from the data

$$P(Y)P(X|Y)$$

- How to estimate parameters without labels?



Mixture Models

Finite Mixture Models

- Mixture model:

$$\boldsymbol{\theta} = \{\lambda_1, \cdots, \lambda_K, \theta_1, \cdots, \theta_K\}$$

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \lambda_k p_k(\mathbf{x}|z_k, \theta_k)$$

Note: Each point is assumed to be generated from 1 mixture component

  - Mixture components: $p_k(\mathbf{x}|z_k, \theta_k)$
  - Binary indicator variables: $\mathbf{z} = (z_1, \cdots, z_K)$
  - Mixture weights: $\lambda_k = p(z_k)$, $\sum_{k=1}^{K} \lambda_k = 1$

Finite Mixture Model: Membership

- Membership weight vector w expresses uncertainty about which of the K components generated the point

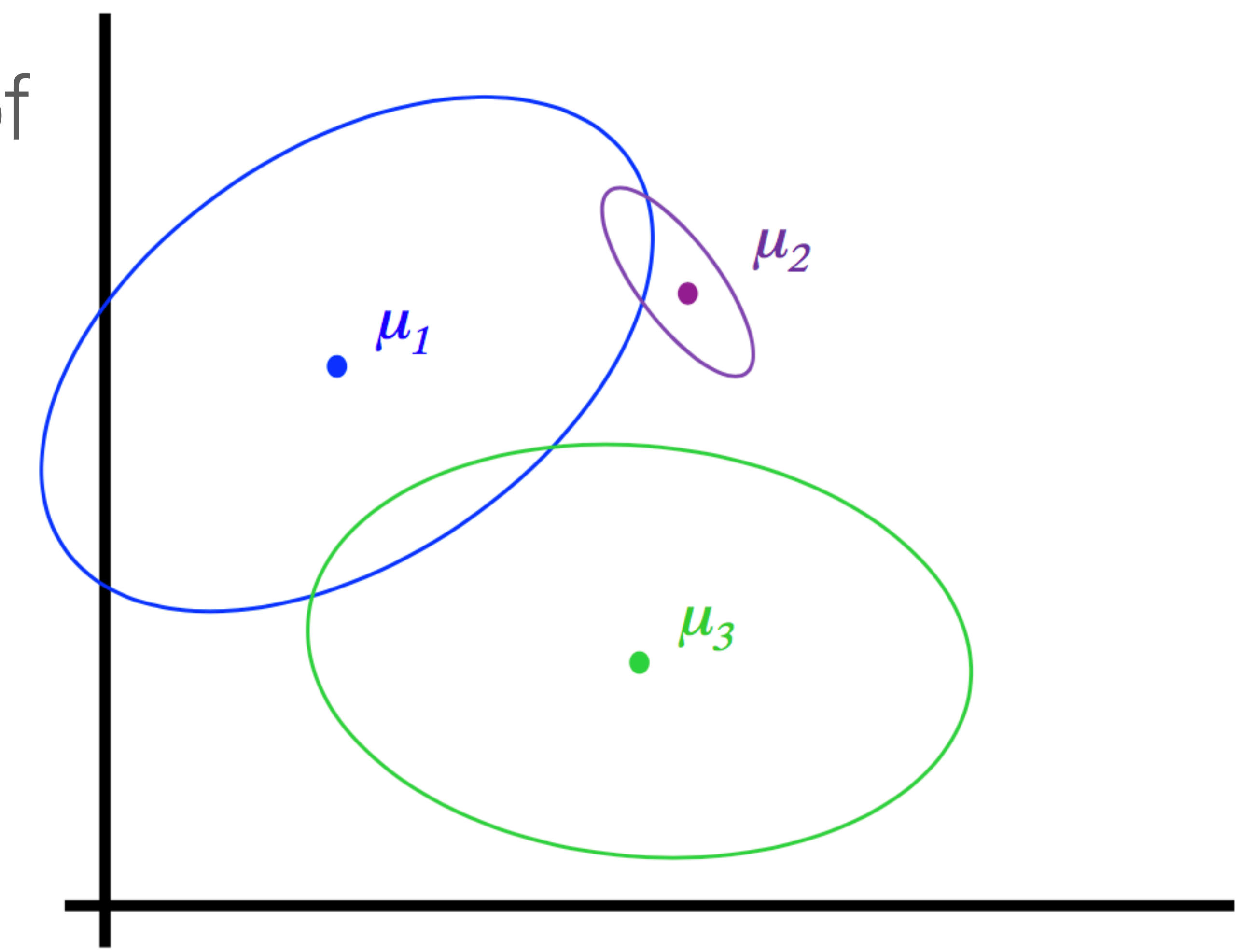
$$w_{ik} = p(z_{ik}|\mathbf{x}_i, \boldsymbol{\theta}) = \frac{\lambda_k p_k(\mathbf{x}_i|z_k, \theta_k)}{\sum_{m=1}^{K} \lambda_m p_m(\mathbf{x}_i|z_m, \theta_m)}$$

Gaussian Mixture Models (GMMs)

- Cluster by fitting a mixture of k Gaussians to the data

- Each components is a multivariate Gaussian with parameters

$$\theta_k = \mu_k, \Sigma_k$$



Example: Simulated Data

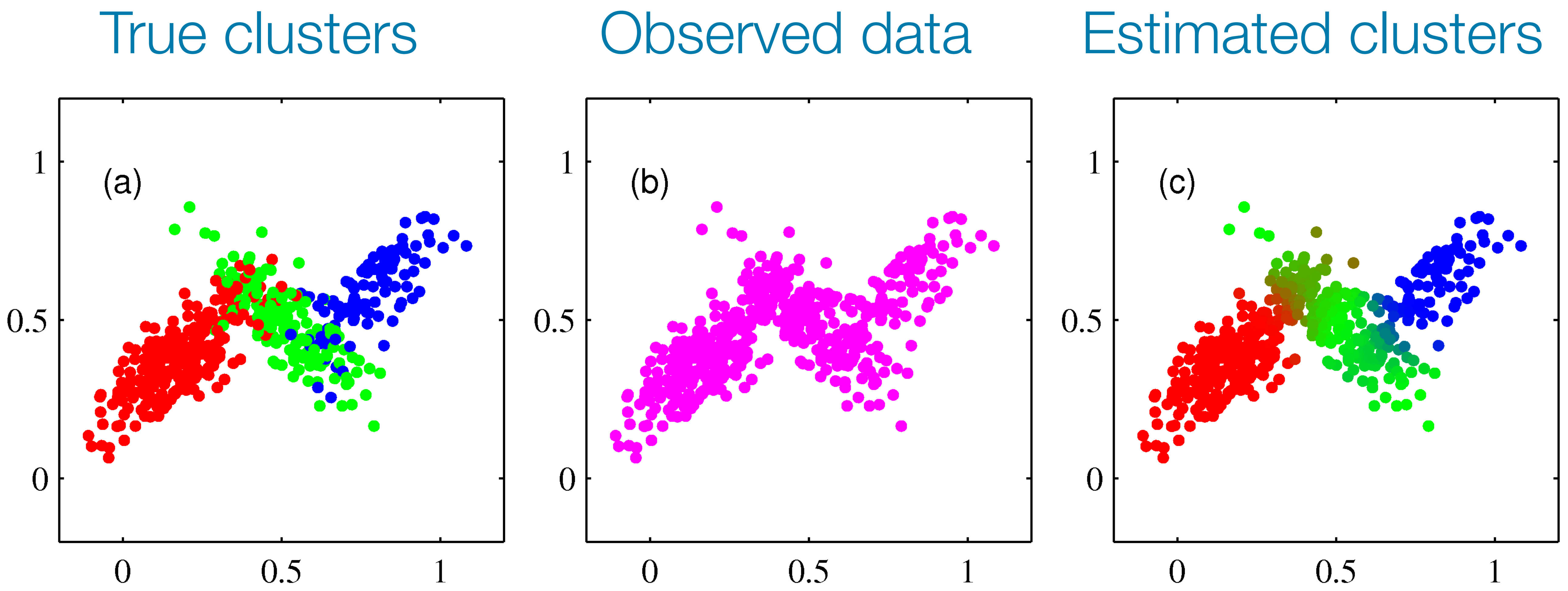
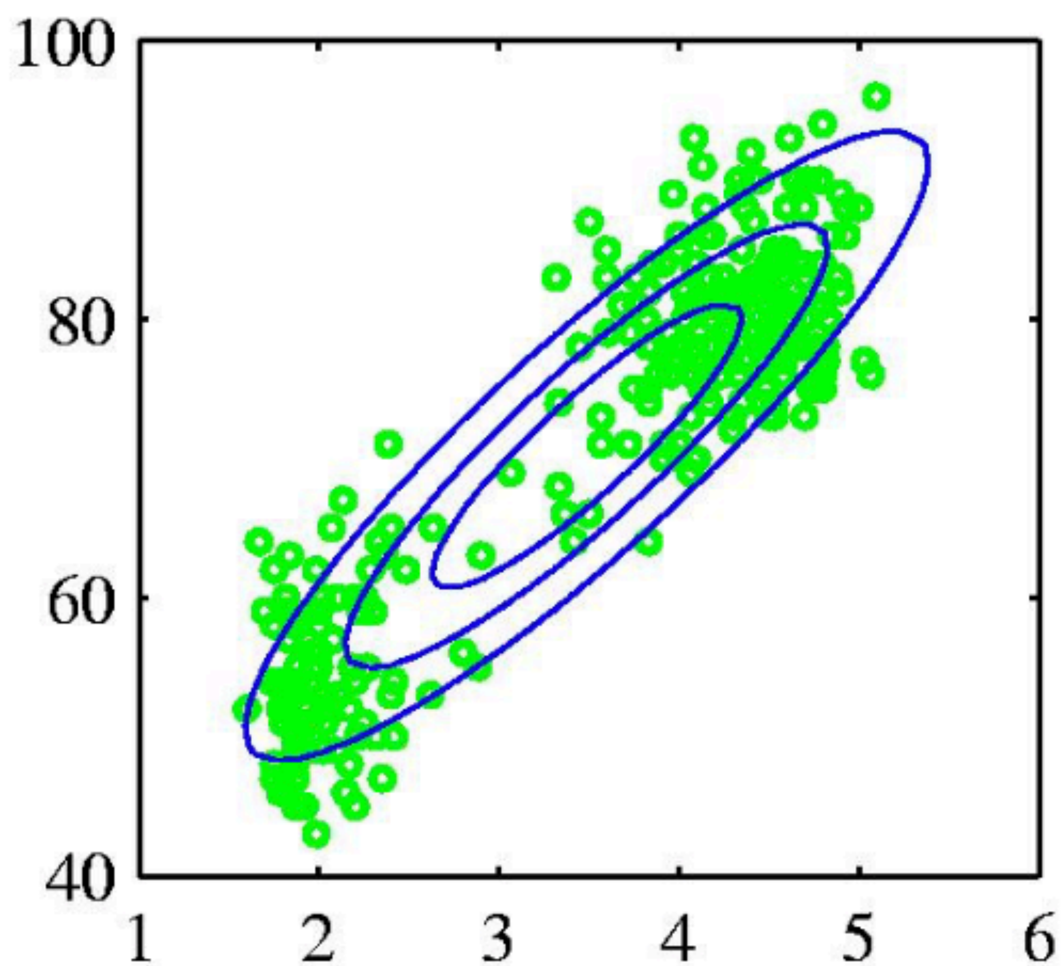
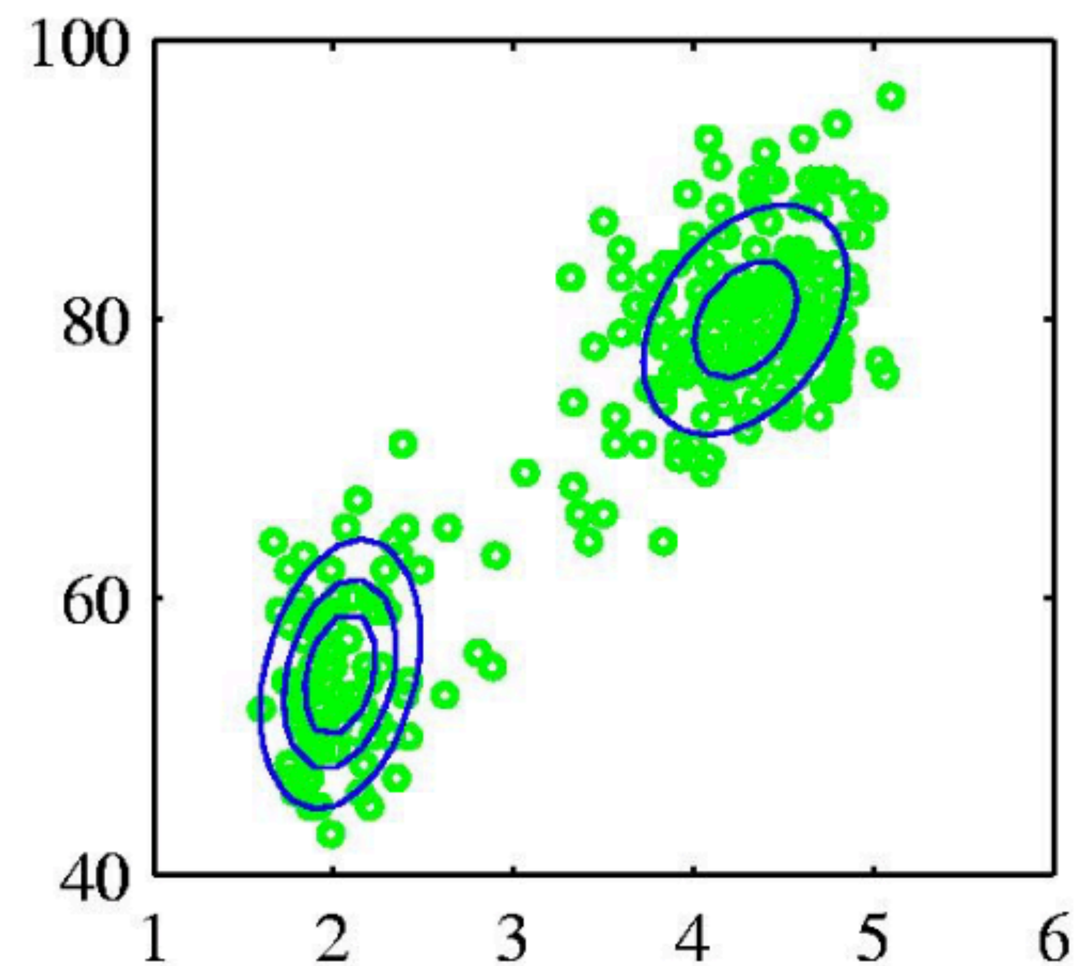


Figure 9.5 (Bishop)

Example: Old Faithful



Single Gaussian



Mixture of two Gaussians

GMM: Learning

- How can we learn the parameters?
- Supervised case: Straightforward — group data based on labels and compute the mean and the covariance from the training data
- Unsupervised case: Differentiating the MLE objective based on the joint probability distribution is difficult to solve

$$\text{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\mathbf{x}_i|\boldsymbol{\theta}) = \text{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{N} \sum_{k=1}^{K} p_k(\mathbf{x}_i|z_k, \theta_k)p(z_k)$$

Expectation Maximization (EM)

EM Algorithm: Idea

- Start with random parameters
- E-step: Find a class for each example based on expectation
  - Each example will be given a vector of probabilities
- M-step: Estimate the parameters of the model using the maximum likelihood method (supervised learning setting)
- Iterate until convergence

EM: E-Step

- Compute w_{ik} for all data points indexed by i and all mixture components indexed by k

$$w_{ik} = p(z_{ik}|\mathbf{x}_i, \boldsymbol{\theta}) = \frac{\lambda_k p_k(\mathbf{x}_i|z_k, \theta_k)}{\sum_{m=1}^{K} \lambda_m p_m(\mathbf{x}_i|z_m, \theta_m)}$$

EM: M-Step

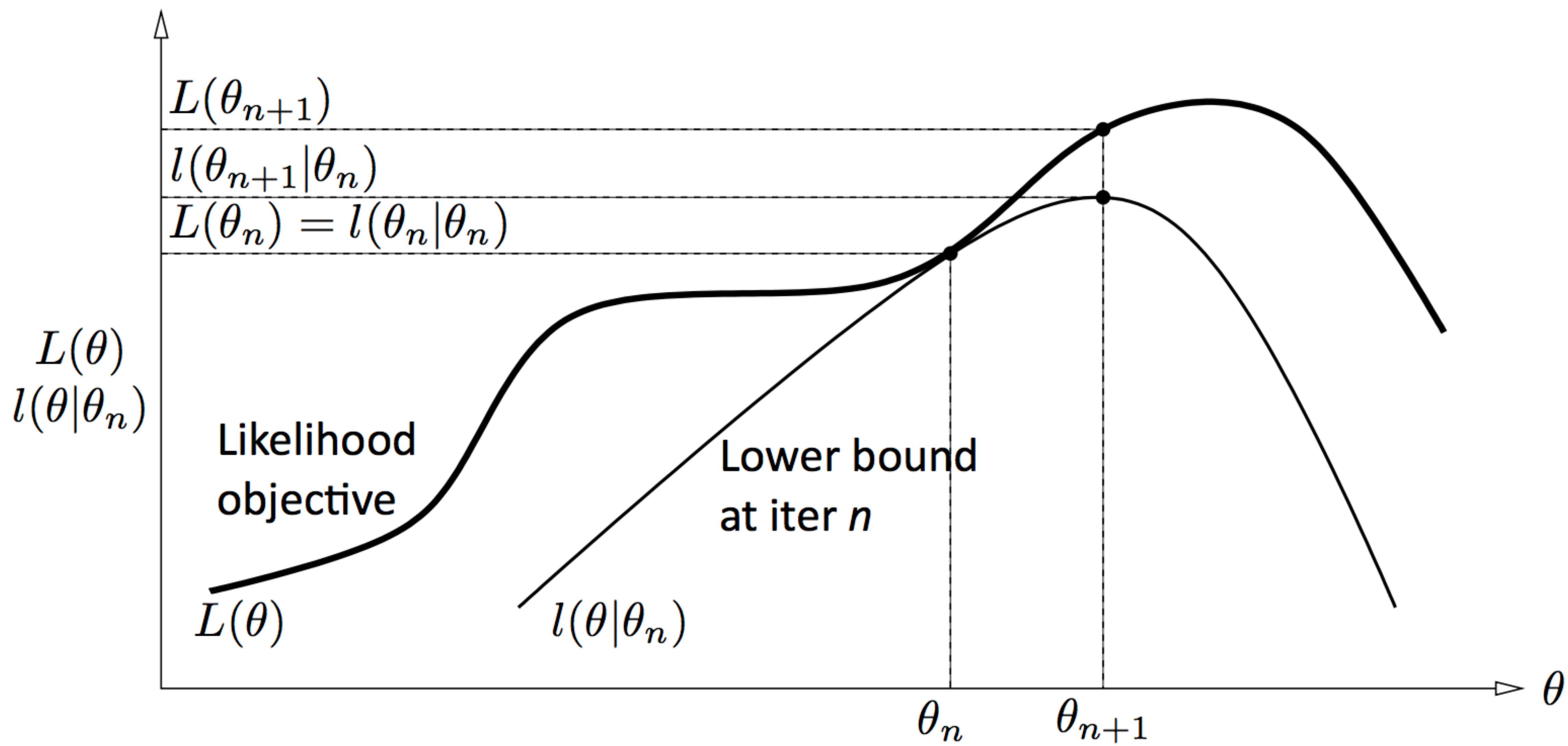
- Re-estimate the parameters using the “weighted” estimates

$$N_k = \sum_{i=1}^{N} w_{ik}, \;\; \lambda_k = \frac{N_k}{N}$$

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N} w_{ik} \mathbf{x}_i$$

$$\Sigma_k = \frac{1}{N_k} \sum_{i=1}^{N} w_{ik} (\mathbf{x}_i - \mu_k)(\mathbf{x}_i - \mu_k)^\top$$

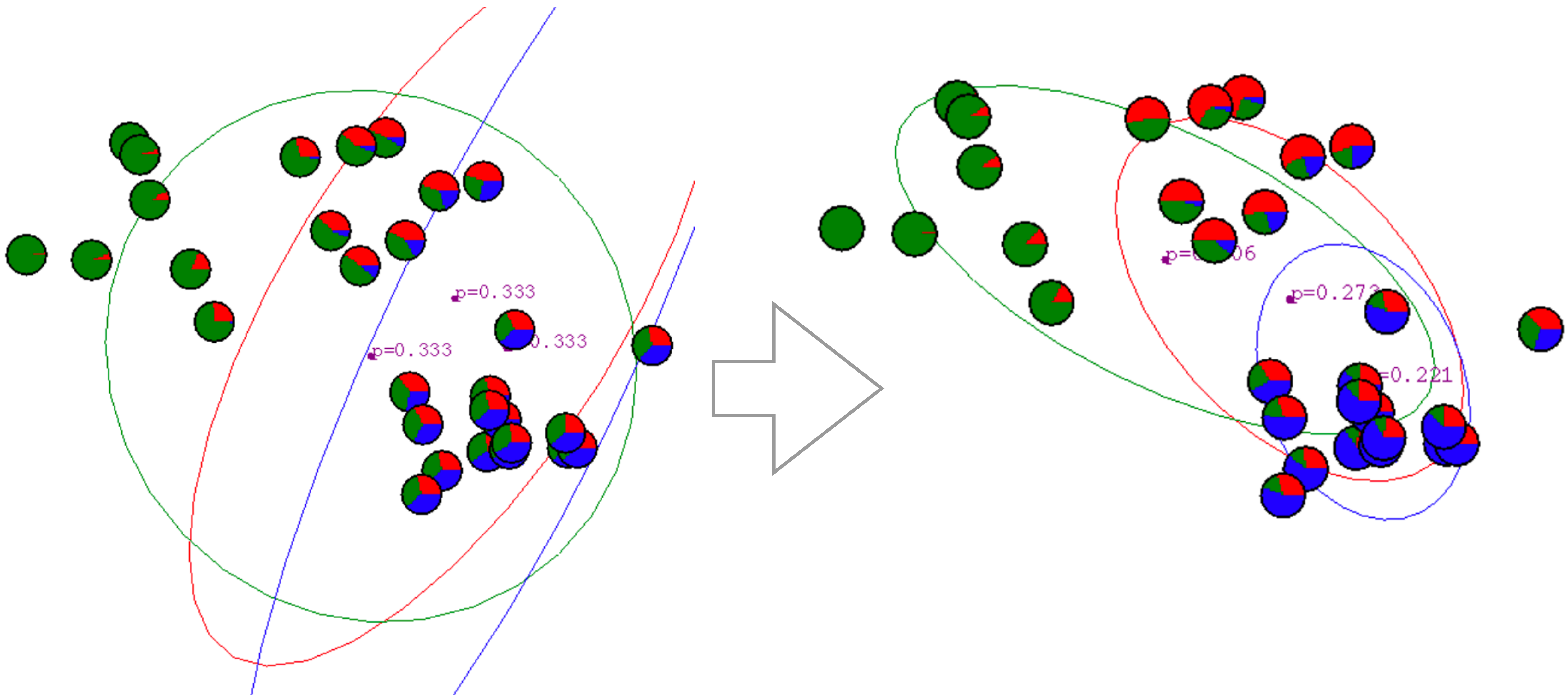
EM: Pictorially



Example: GMM

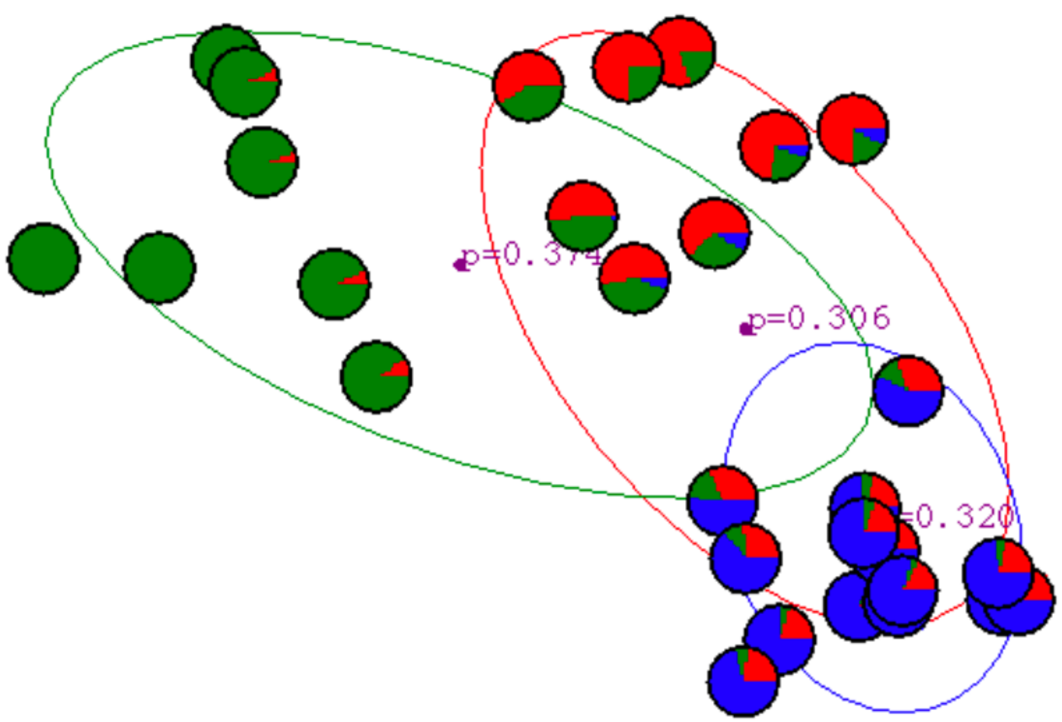
Start

Iteration 1

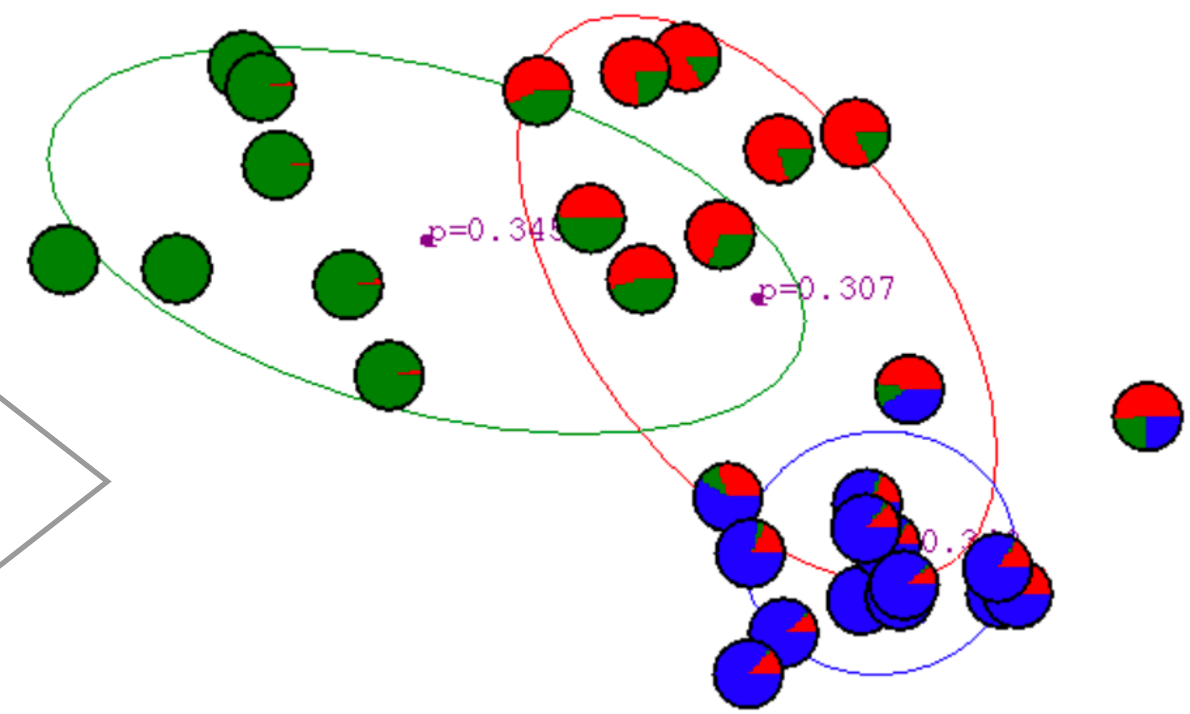


Example: GMM

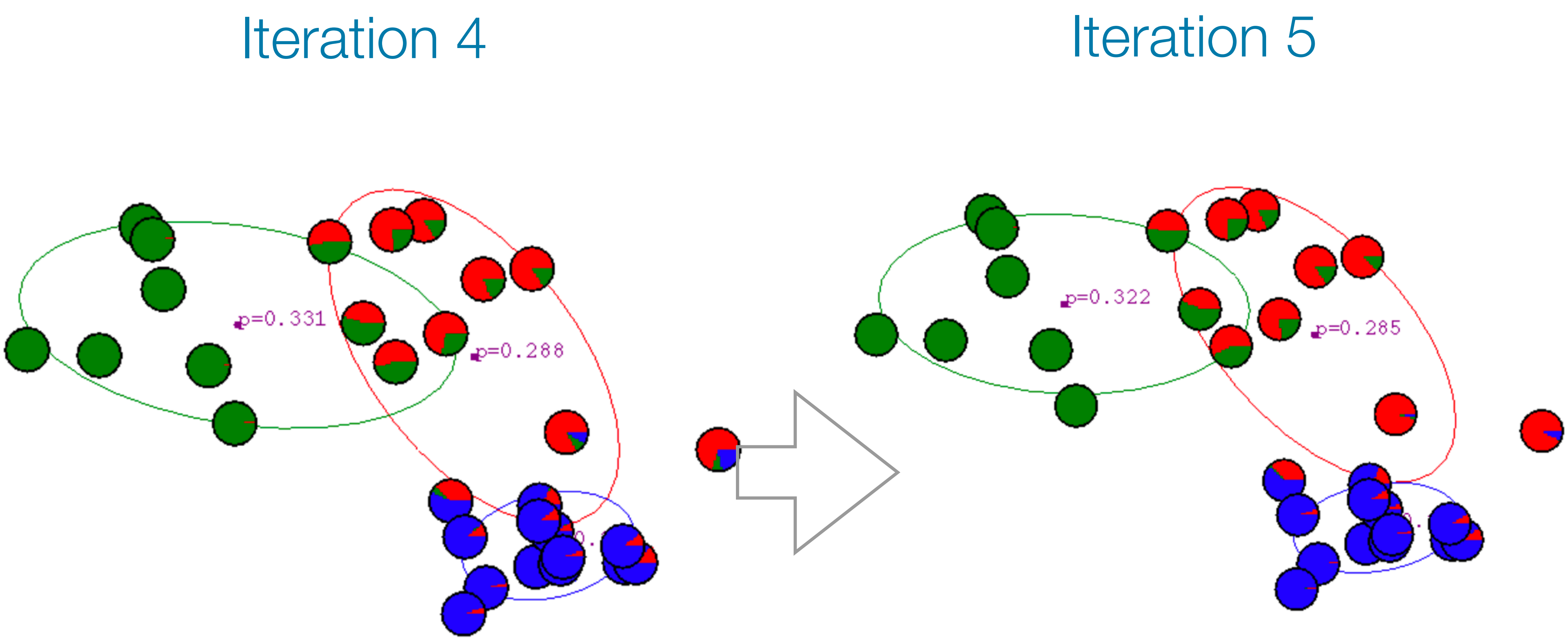
Iteration 2



Iteration 3

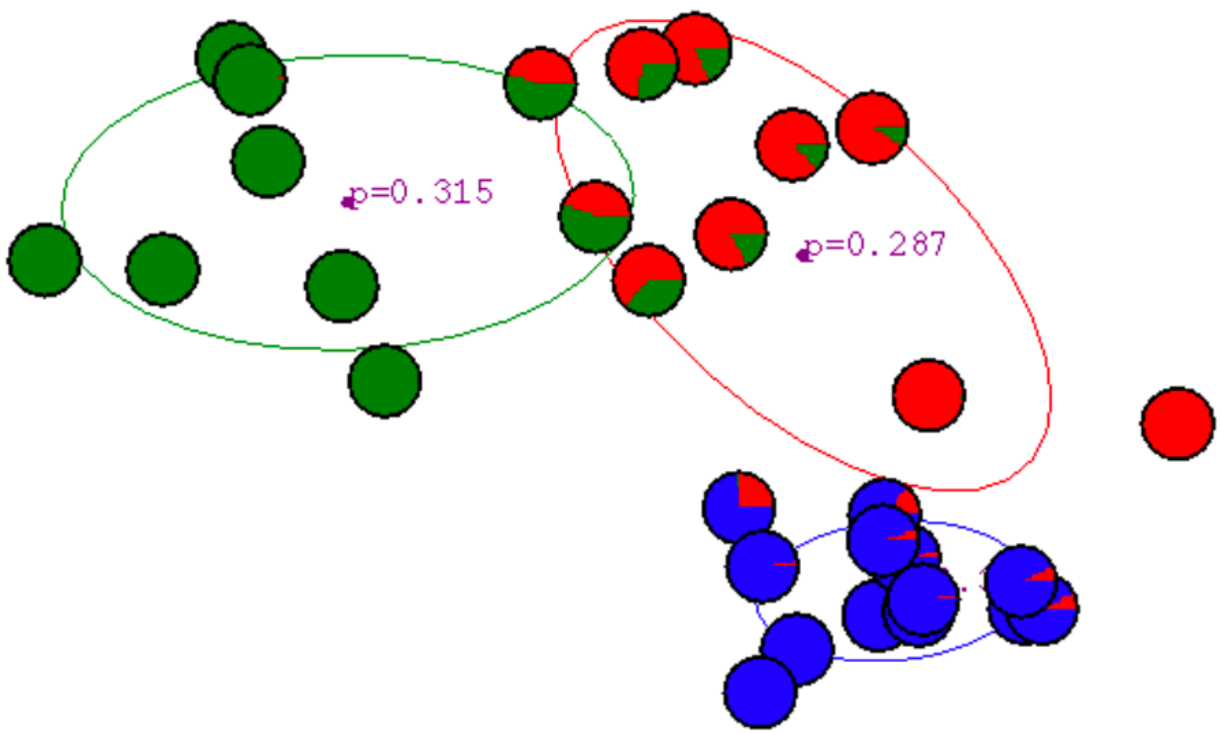


Example: GMM

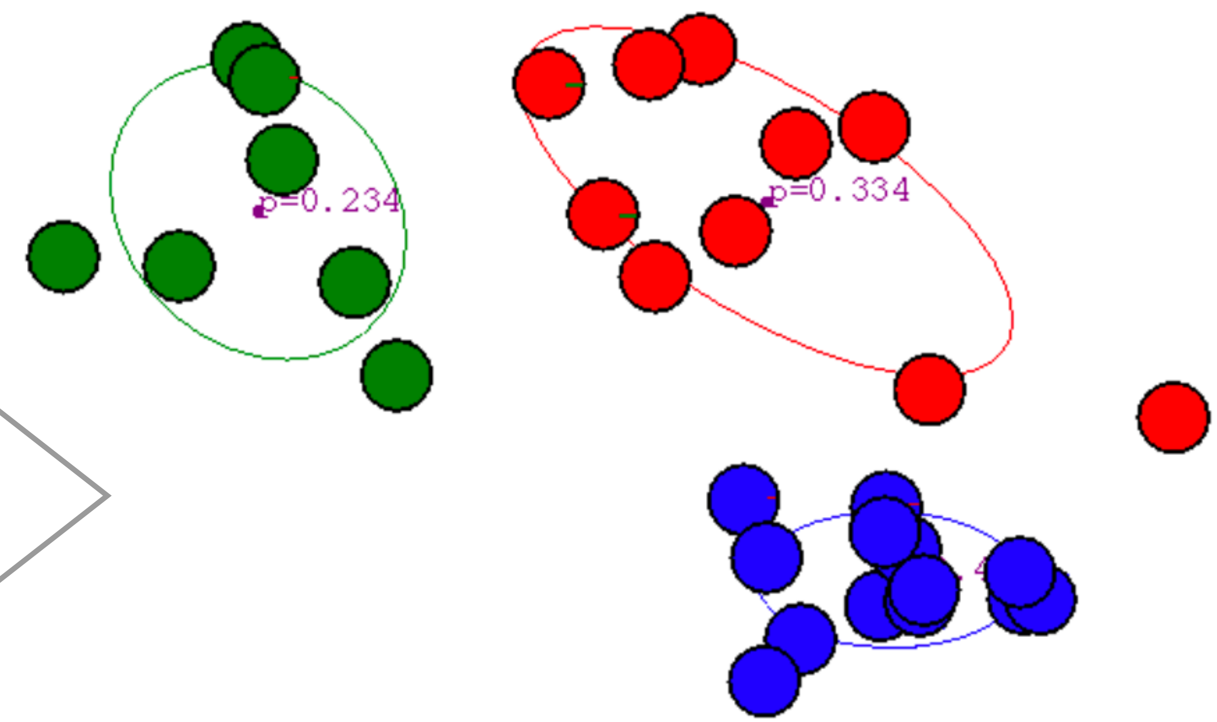


Example: GMM

Iteration 6



Iteration 20



EM: Properties

- Converges to local minima
  - Each iteration improves the log-likelihood
  - Proof is the same as K-means
- Hard assignments —> equivalent to K-means