13. Clustering

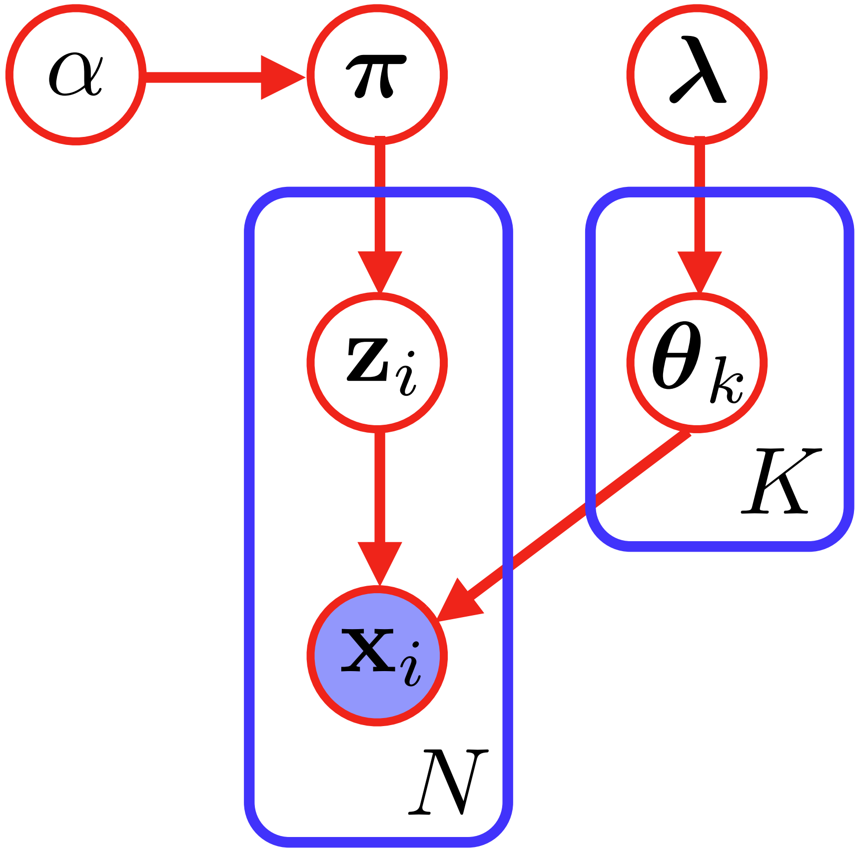
Motivation

- Supervised learning is good for interaction with humans, but labels from a supervisor are hard to obtain
- Clustering is **unsupervised** learning, i.e. it tries to lear only from the data
- Main idea: find a similarity measure and group similar data objects together
- Clustering is a very old research field, many approaches have been suggested
- Main problem in most methods: how to find a good number of clusters

Clustering using Mixture Models

- The full posterior of the Gaussian Mixture Model is

$$p(X, Z, \boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) = \underbrace{p(X \mid Z, \boldsymbol{\mu}, \Sigma)}_{\text{data likelihood (Gaussian)}} \underbrace{p(Z \mid \boldsymbol{\pi})}_{\text{correspondence prob. (Multinomial)}} \underbrace{p(\boldsymbol{\pi} \mid \alpha)}_{\text{mixture prior (Dirichlet)}} \underbrace{p(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{\lambda})}_{\text{parameter prior (Gauss-IW)}}$$



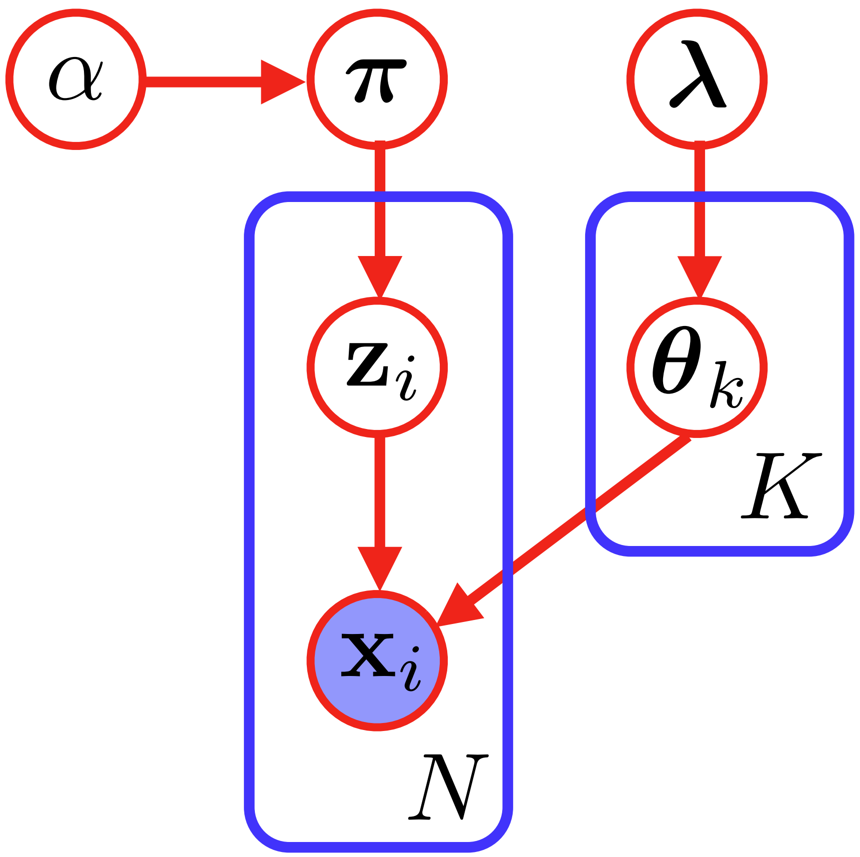
In this model, we use:

- $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$
- $\Sigma = (\Sigma_1, \ldots, \Sigma_K)$
- $(\boldsymbol{\mu}_k, \Sigma_k) = \boldsymbol{\theta}_k$

Clustering using Mixture Models

- The full posterior of the Gaussian Mixture Model is

$$p(X, Z, \boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) = \underbrace{p(X \mid Z, \boldsymbol{\mu}, \Sigma)}_{\text{data likelihood (Gaussian)}} \underbrace{p(Z \mid \boldsymbol{\pi})}_{\text{correspondence prob. (Multinomial)}} \underbrace{p(\boldsymbol{\pi} \mid \alpha)}_{\text{mixture prior (Dirichlet)}} \underbrace{p(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{\lambda})}_{\text{parameter prior (Gauss-IW)}}$$



Given this model, we can create new samples:

1. Sample $\boldsymbol{\pi}, \boldsymbol{\theta}_k$ from priors
2. Sample corresp. $\mathbf{z}_i$
3. Sample data point $\mathbf{x}_i$

Clustering using Mixture Models

- The full posterior of the Gaussian Mixture Model is

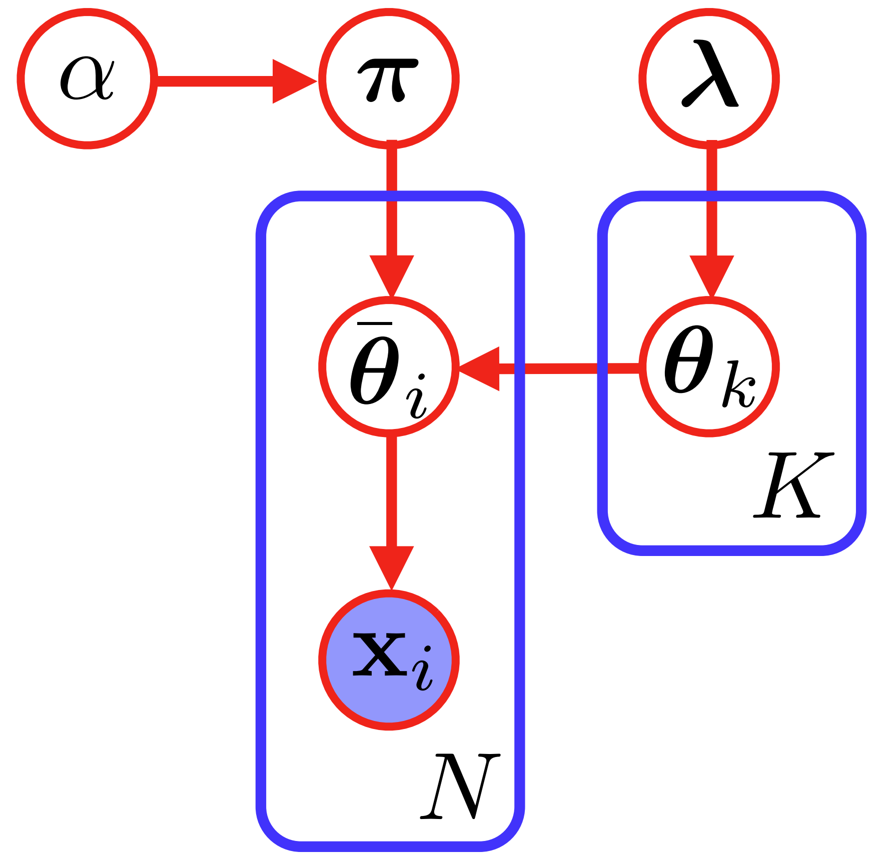
$$p(X, Z, \boldsymbol{\mu}, \Sigma, \boldsymbol{\pi}) = \underbrace{p(X \mid Z, \boldsymbol{\mu}, \Sigma)}_{\substack{\text{data likelihood} \\ \text{(Gaussian)}}} \underbrace{p(Z \mid \boldsymbol{\pi})}_{\substack{\text{correspondence} \\ \text{prob. (Multinomial)}}} \underbrace{p(\boldsymbol{\pi} \mid \alpha)}_{\substack{\text{mixture prior} \\ \text{(Dirichlet)}}} \underbrace{p(\boldsymbol{\mu}, \Sigma \mid \boldsymbol{\lambda})}_{\substack{\text{parameter prior} \\ \text{(Gauss-IW)}}}$$

An equivalent formulation of this model is this:

1. Sample $\boldsymbol{\pi}, \boldsymbol{\theta}_k$ from priors
2. Sample params $\bar{\boldsymbol{\theta}}_i$ from:

$$p(\bar{\boldsymbol{\theta}}_i \mid \boldsymbol{\pi}, \boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \delta(\boldsymbol{\theta}_k, \bar{\boldsymbol{\theta}}_i)$$

3. Sample data point $\mathbf{x}_i$



Clustering using Mixture Models

What is the difference in that model?

- there is one parameter $\bar{\boldsymbol{\theta}}_i$ for each observation $\mathbf{x}_i$
- intuitively: we first sample the location of the cluster and then the data that corresponds to it

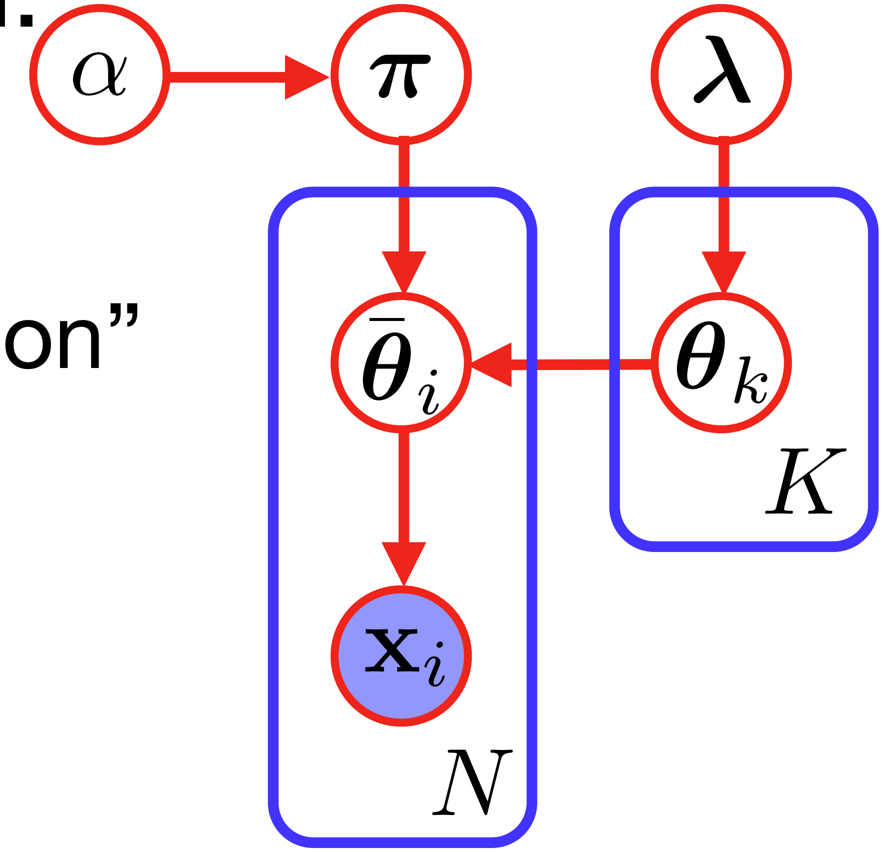
In general, we use the notation:

$$\boldsymbol{\pi} \sim \mathrm{Dir}(\frac{\alpha}{K}\mathbf{1})$$

$$\boldsymbol{\theta}_k \sim \mathrm{H}(\boldsymbol{\lambda})$$ “Base distribution”

$$\bar{\boldsymbol{\theta}}_i \sim \mathrm{G}(\boldsymbol{\pi}, \boldsymbol{\theta}_k)$$ where

$$G(\boldsymbol{\pi}, \boldsymbol{\theta}_k) = \sum_{k=1}^{K} \pi_k \delta(\boldsymbol{\theta}_k, \bar{\boldsymbol{\theta}}_i)$$



However: We need to know K

The Dirichlet Process

- So far, we assumed that K is known
- To extend that to infinity, we use a trick:

Definition: A Dirichlet process (DP) is a distribution over probability measures G, i.e. $G(\boldsymbol{\theta}) \geq 0$ and $\int G(\boldsymbol{\theta}) d\boldsymbol{\theta} = 1$. If for any partition $(T_1, \ldots, T_K)$ it holds:

$$(G(T_1), \ldots, G(T_K)) \sim \mathrm{Dir}(\alpha H(T_1), \ldots, \alpha H(T_K))$$

then G is sampled from a Dirichlet process.

Notation: $G \sim \mathrm{DP}(\alpha, H)$

where α is the **concentration parameter**
and H is the **base measure**

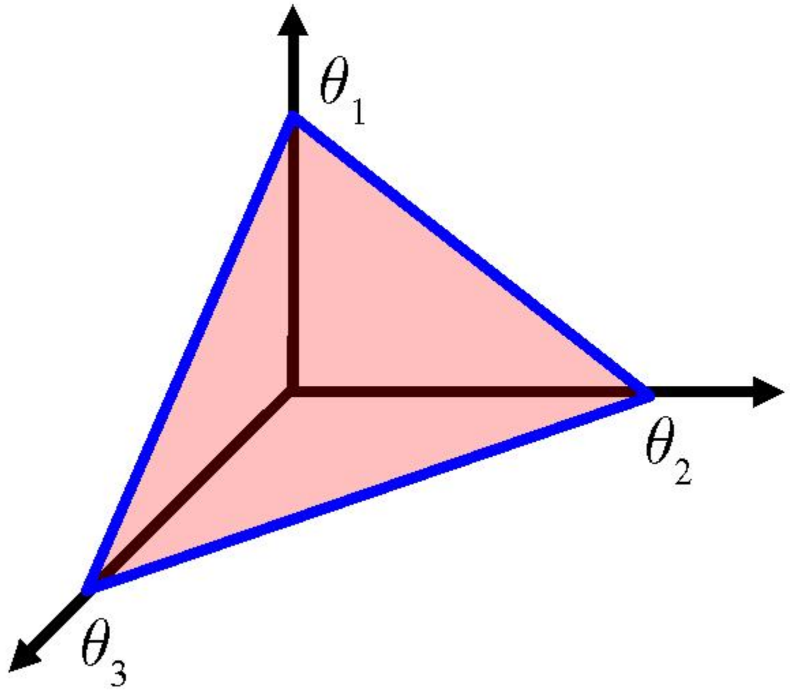
Repetition

- The Dirichlet distribution is defined as:

$$\mathrm{Dir}(\boldsymbol{\mu} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_K)} \prod_{k=1}^{K} \mu_k^{\alpha_k - 1} \qquad \alpha_0 = \sum_{k=1}^{K} \alpha_k$$

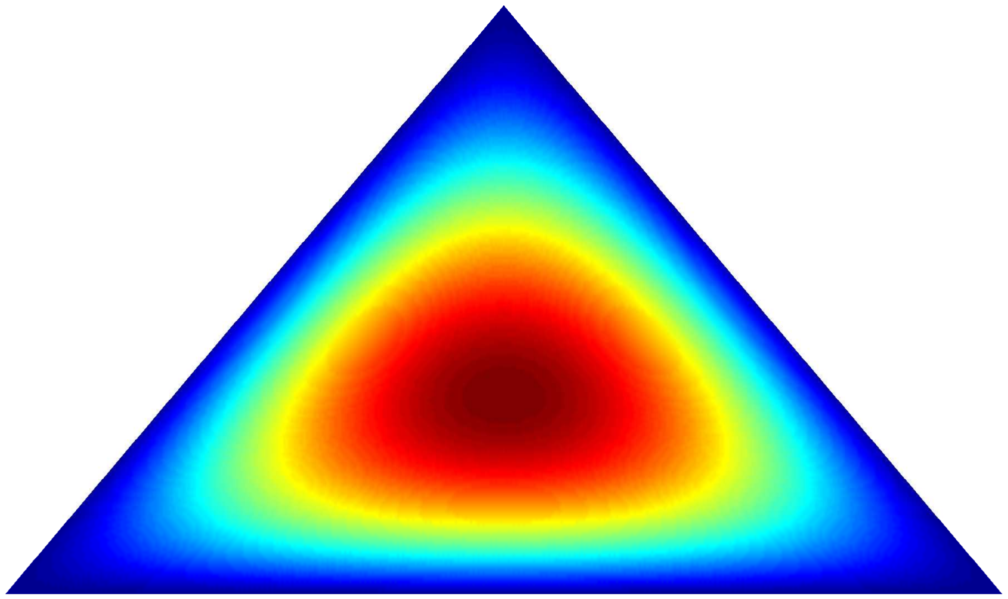
$$0 \leq \mu_k \leq 1 \qquad \sum_{k=1}^{K} \mu_k = 1$$

- It is the conjugate prior for the multinomial distribution
- There, the parameter $\boldsymbol{\alpha}$ can be interpreted as the effective number of observations for every state

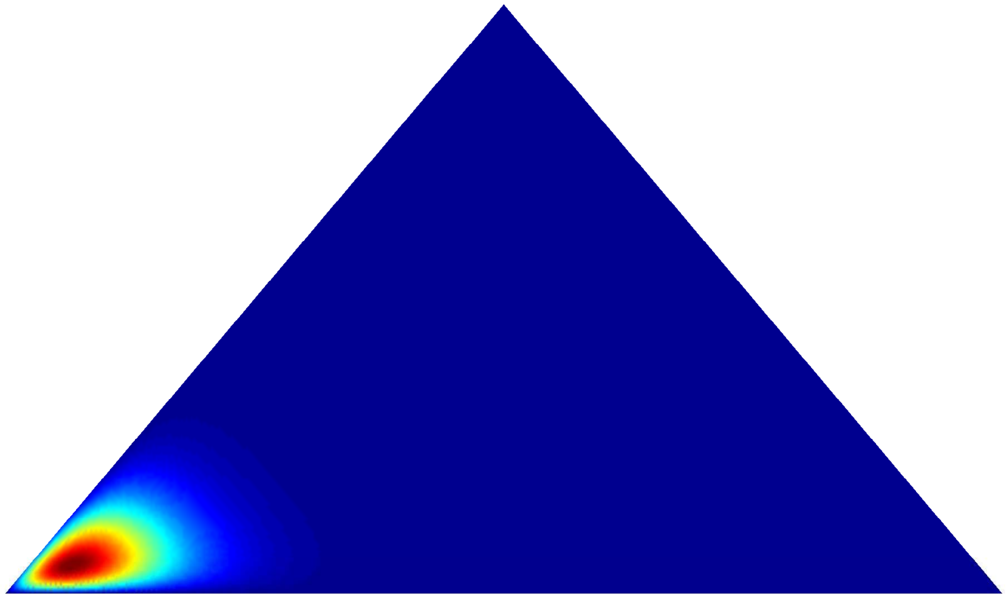


The simplex for K=3

Some Examples

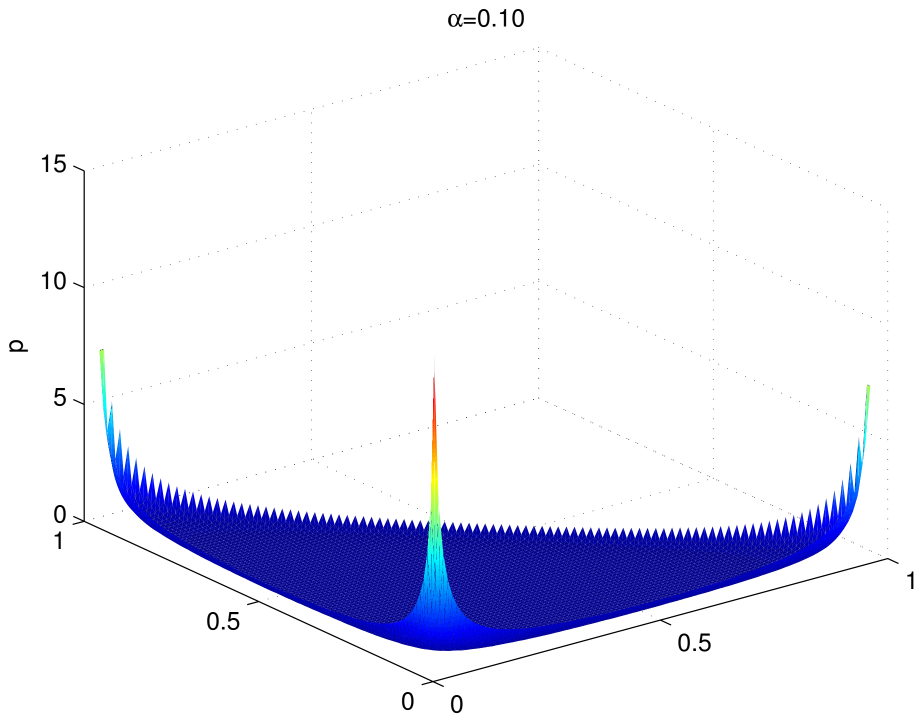


$\alpha = (2, 2, 2)$



$\alpha = (20, 2, 2)$

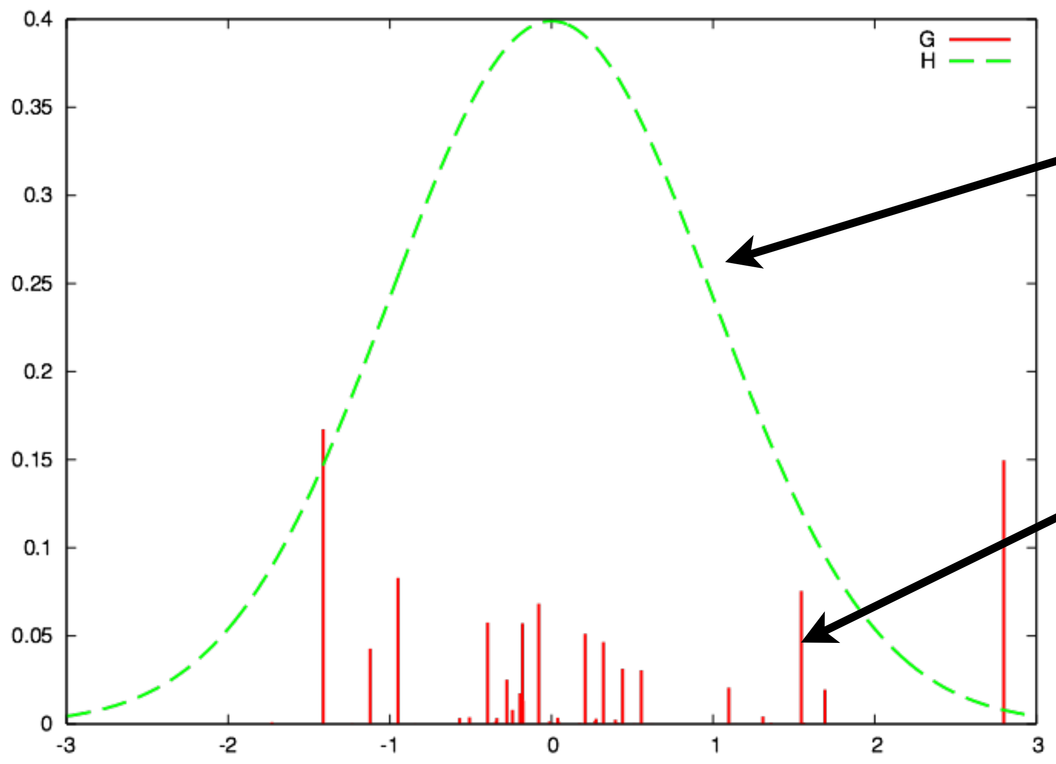
- α_0 controls the strength of the distribution ("peakedness")
- α_k control the location of the peak



$\alpha = (0.1, 0.1, 0.1)$

Intuitive Interpretation

- Every sample from a Dirichlet distribution is a vector of K positive values that sum up to 1, i.e. the sample itself is a finite distribution
- Accordingly, a sample from a Dirichlet process is an infinite (but still discrete!) distribution



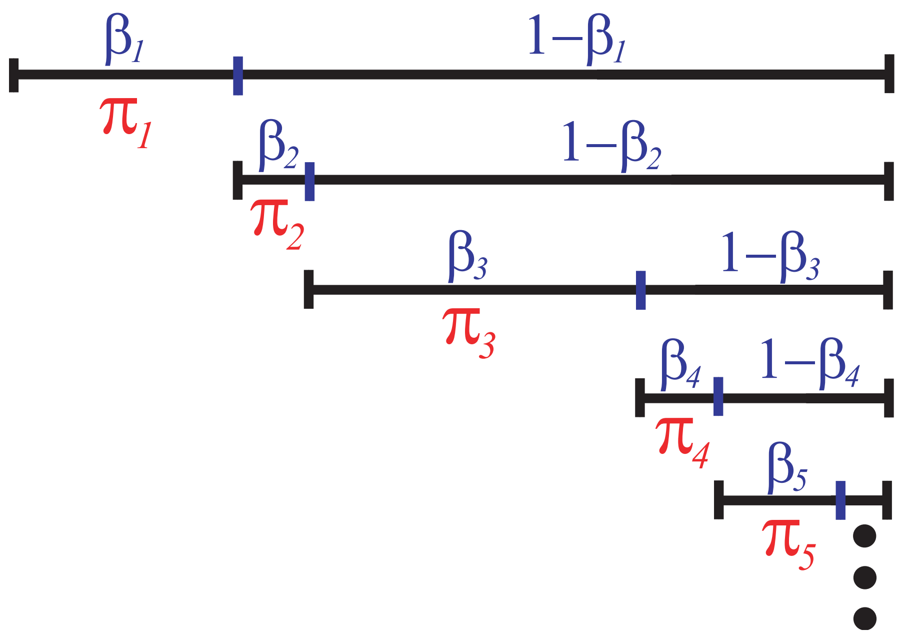
Base distribution (here Gaussian)

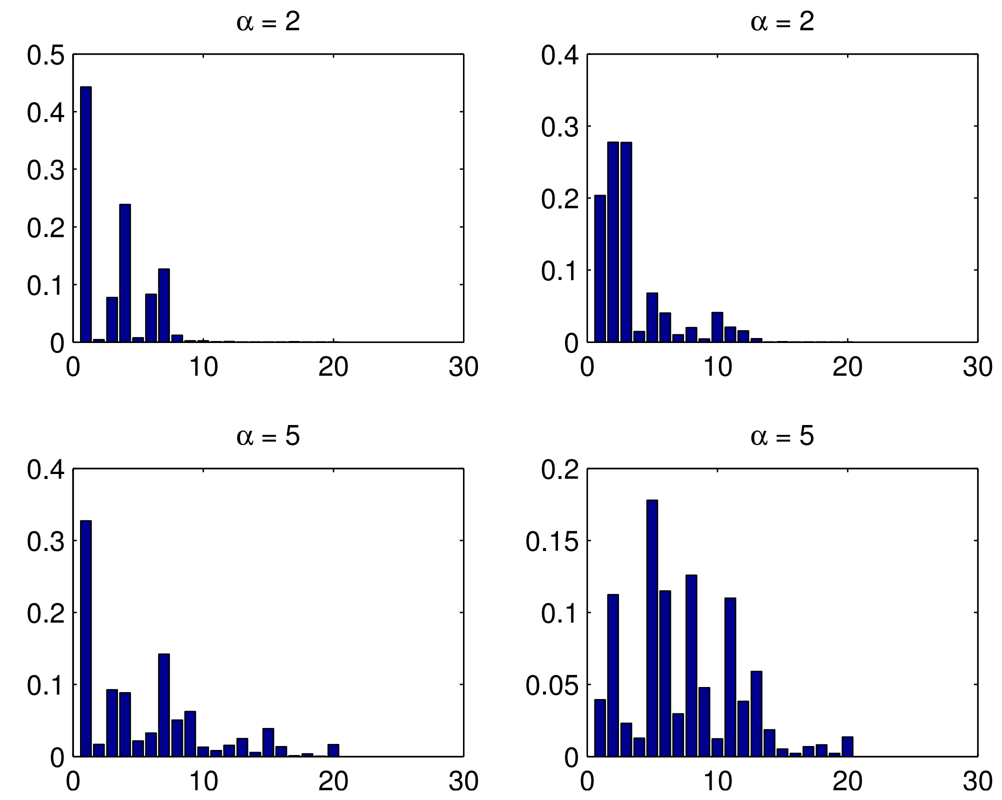
Infinitely many samples (sum up to 1)

Construction of a Dirichlet Process

- The Dirichlet process is only defined **implicitly**, i.e. we can test whether a given probability measure is sampled from a DP, but we can not yet construct one.
- A DP can be constructed using the “stick-breaking” analogy:
  - imagine a stick of length 1
  - we select a random number β between 0 and 1 from a Beta-distribution
  - we break the stick at $\pi = \beta$ * length-of-stick
  - we repeat this infinitely often

The Stick-Breaking Construction





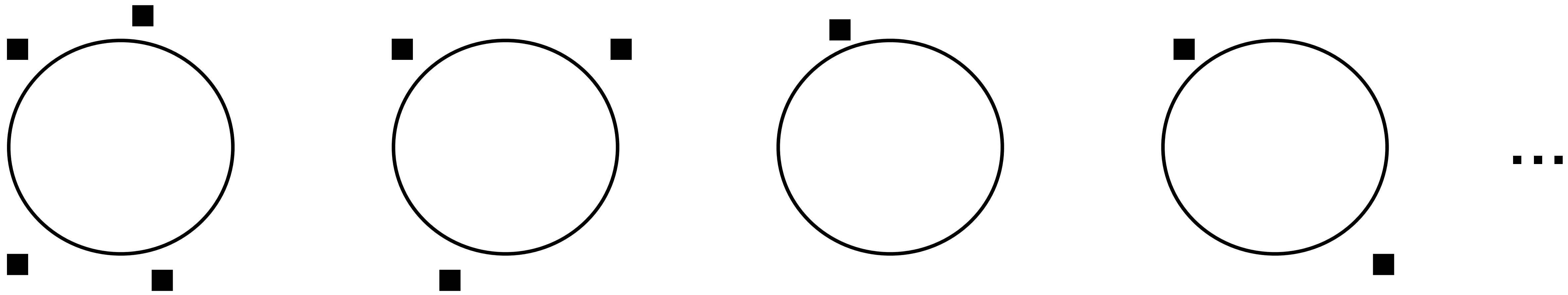
- formally, we have

$$\beta_k \sim \mathrm{Beta}(1, \alpha) \qquad \pi_k = \beta_k \prod_{l=1}^{k-1}(1-\beta_l) = \beta_k(1 - \sum_{l=1}^{k-1}\pi_l)$$

- now we define

$$G(\boldsymbol{\theta}) = \sum_{k=1}^{\infty} \pi_k \delta(\boldsymbol{\theta}_k, \boldsymbol{\theta}) \qquad \boldsymbol{\theta}_k \sim H \qquad \text{then:} \quad G \sim \mathrm{DP}(\alpha, H)$$

The Chinese Restaurant Process



- Consider a restaurant with infinitely many tables
- Everytime a new customer comes in, he sits at an **occupied table** with probability **proportional to the number of people** sitting at that table, but he may choose to sit on a **new** table with **decreasing** probability as more customers enter the room.

The Chinese Restaurant Process

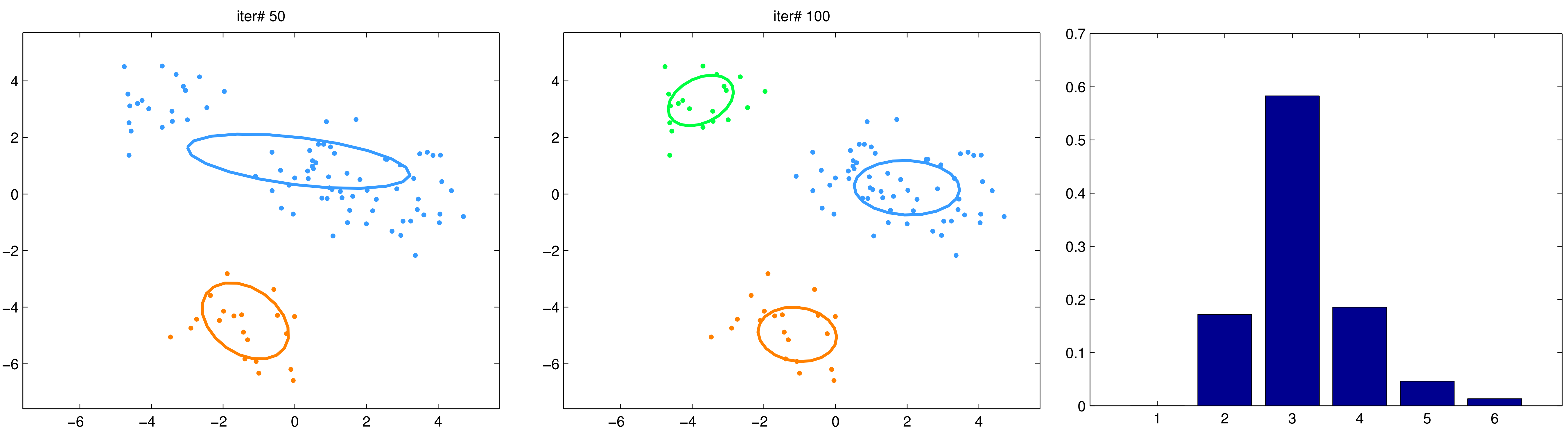
- It can be shown that the probability for a new customer is

$$p(\bar{\boldsymbol{\theta}}_{N+1} = \boldsymbol{\theta} \mid \bar{\boldsymbol{\theta}}_{1:N}, \alpha, H) = \frac{1}{\alpha + N} \left(\alpha H(\boldsymbol{\theta}) + \sum_{k=1}^{K} N_k \delta(\bar{\boldsymbol{\theta}}_k, \boldsymbol{\theta}) \right)$$

- This means that currently occupied tables are more likely to get new customers (**rich get richer**)
- The number of occupied tables grows logarithmically with the number of customers

The DP for Mixture Modeling

- Using the stick-breaking construction, we see that we can extend the mixture model clustering to the situation where K goes to infinity
- The algorithm can be implemented using Gibbs sampling



Affinity Propagation

- Often, we are only given a **similarity matrix for** the data points
- The idea of Affinity Propagation is to determine cluster centers (“exemplars”) that explain other data points in an optimal way
- This is similar to k-medoids, but the algorithm is more robust against local minima
- **Idea:** each data point must choose another data point as its exemplar; some points will choose themselves as exemplar
- The number of clusters is then found automatically

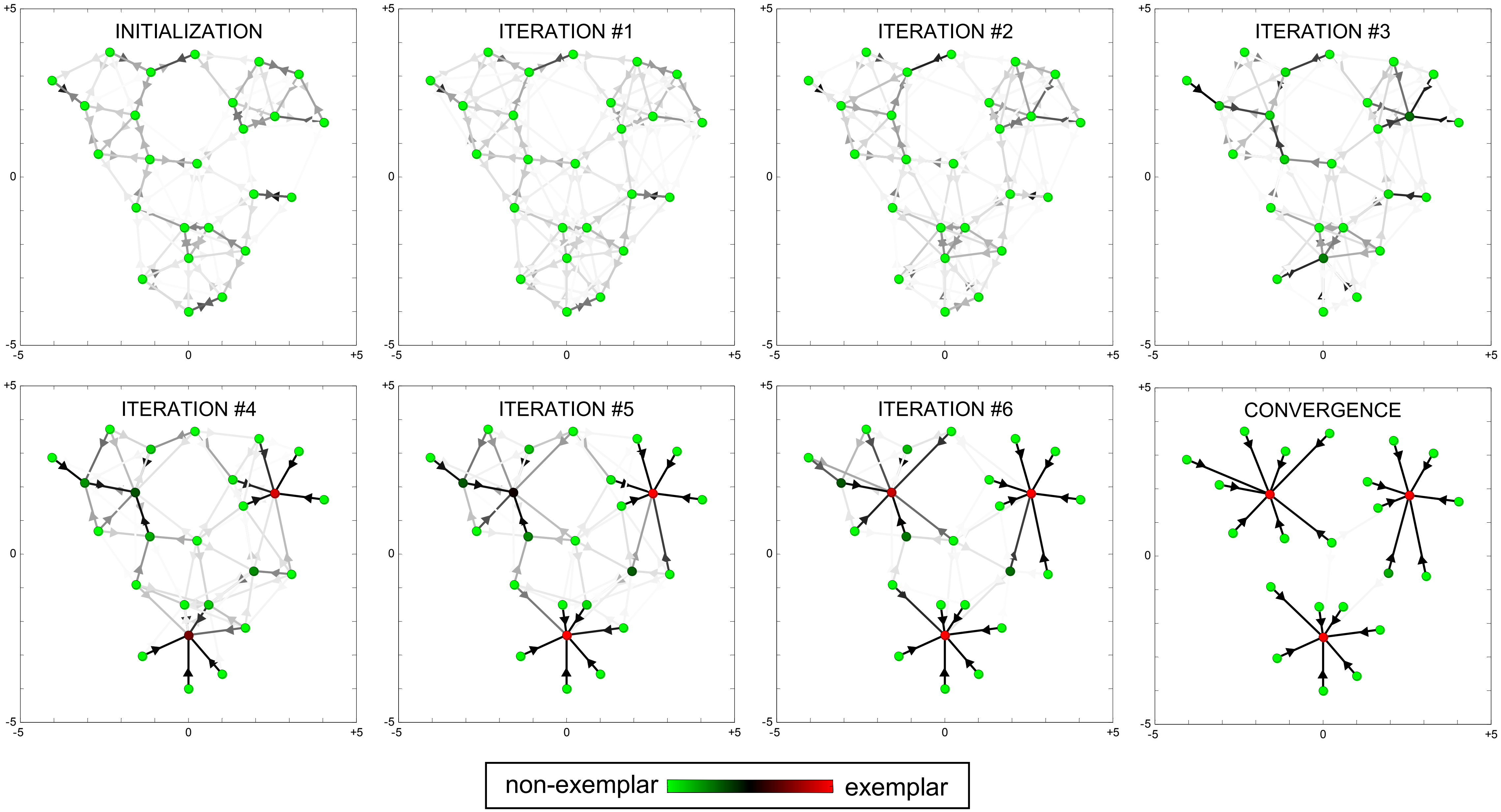
Affinity Propagation

- Input: similarity values s(i,j)
- Initialize the responsibilities r(i,j), and the availabilities a(i,j) to 0
- do until convergence:
  - recompute the responsibilities:
    $$r(i,j) = s(i,j) - \max_{j' \neq j}\{a(i,j') + s(i,j')\}$$
  - recompute the availabilities:
    $$a(i,j) = \min\left\{0, r(j,j) + \sum_{i' \notin \{i,j\}} \max\{0, r(i',j)\}\right\}$$
- the j that maximizes r(i,j) + a(i,j) is the exemplar of i

Affinity Propagation

- Intuitively:
  - responsibility measures how much i thinks that j would be a good exemplar
  - availability measures how strongly j things it should be an exemplar for i
- The algorithm can be shown to be equivalent to max-product loopy belief propagation
- Convergence is not guaranteed, but with “damping” oscillations can be avoided
- The number of clusters can be controlled by the “self-similarity”

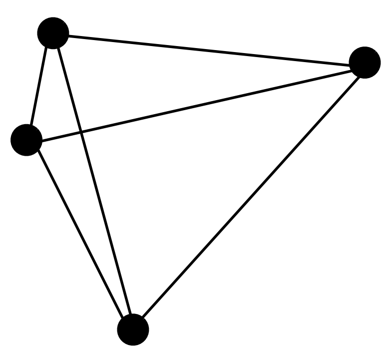
Affinity Propagation



- Colours: how much each point wants to be an exemplar
- Edge strengths: how much a point wants to belong to a cluster

Spectral Clustering

- Consider an undirected graph that connects all data points
- The edge weights are the similarities ("closeness")
- We define the weighted degree d_i of a node as the sum of all outgoing edges



$$W = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}$$

$$d_i = \sum_{j=1}^{N} w_{ij}$$

$$D = \begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & d_3 & \\ & & & d_4 \end{bmatrix}$$

Spectral Clustering

- The Graph Laplacian is defined as:

$$L = D - W$$

- This matrix has the following properties:
  - the 1 vector is eigenvector with eigenvalue 0

Spectral Clustering

- The Graph Laplacian is defined as:

$$L = D - W$$

- This matrix has the following properties:
  - the 1 vector is eigenvector with eigenvector 0
  - the matrix is symmetric and positive semi-definite

Spectral Clustering

- The Graph Laplacian is defined as:

$$L = D - W$$

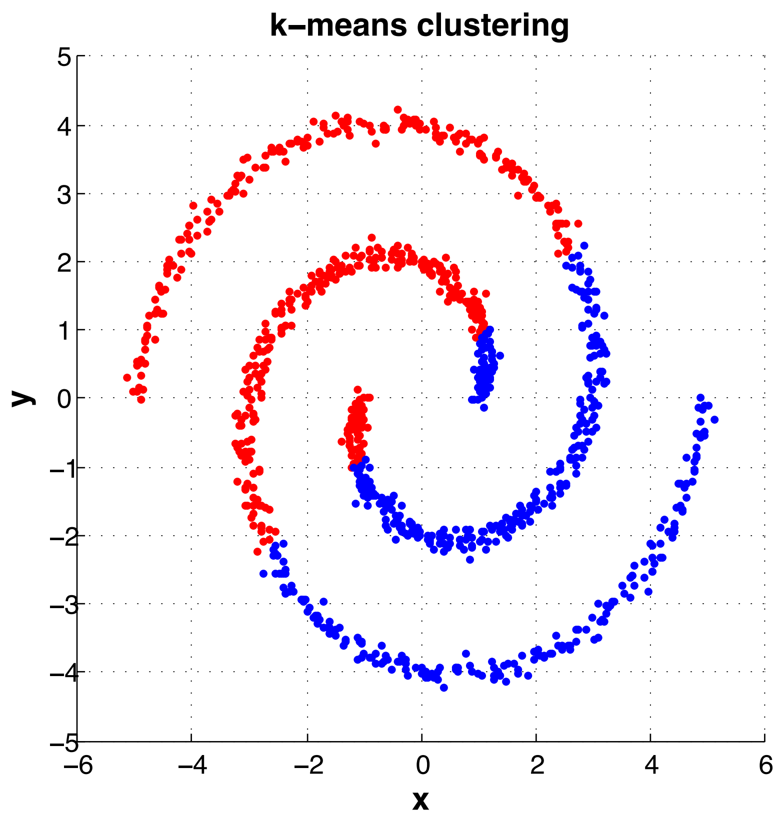
- This matrix has the following properties:
  - the 1 vector is eigenvector with eigenvector 0
  - the matrix is symmetric and positive semi-definite
- With these properties we can show:

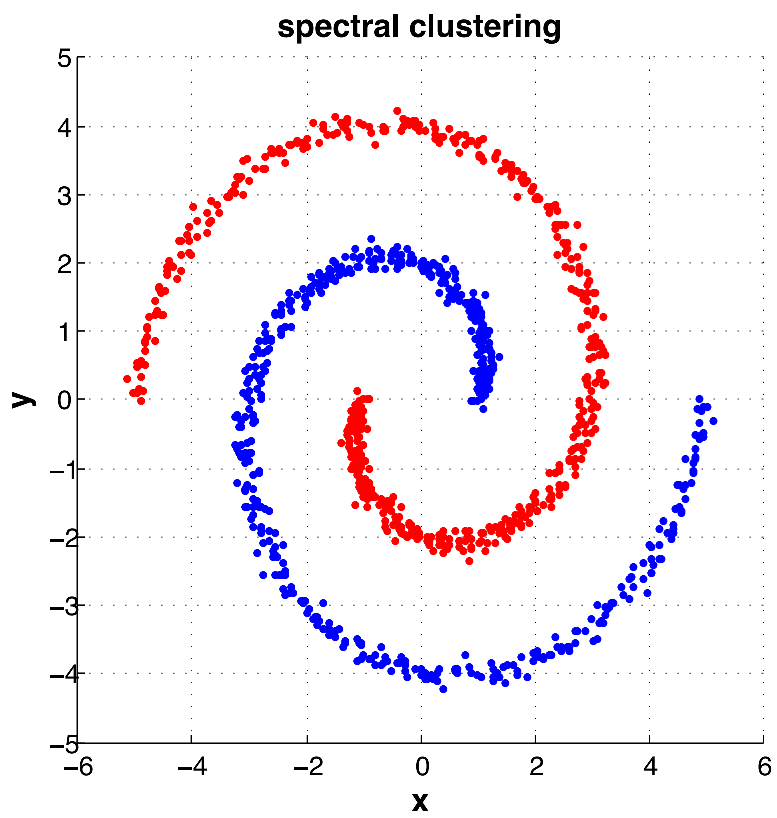
Theorem: The set of eigenvectors of L with eigenvalue 0 is spanned by the indicator vectors $\mathbf{1}_{A_1}, \ldots, \mathbf{1}_{A_K}$, where A_k are the K connected components of the graph.

The Algorithm

- Input: Similarity matrix W
- Compute L = D - W
- Compute the eigenvectors that correspond to the K smallest eigenvalues
- Stack these vectors as columns in a matrix U
- Treat each row of U as a K-dim data point
- Cluster the N rows with K-means clustering
- The indices of the rows that correspond to the resulting clusters are those of the original data points.

An Example





- Spectral clustering can handle complex problems such as this one
- The complexity of the algorithm is $O(N^3)$, because it has to solve an eigenvector problem
- But there are efficient variants of the algorithm

Further Remarks

- To account for nodes that are highly connected, we can use a normalized version of the graph Laplacian
- Two different methods exist:
  - $L_{rw} = D^{-1}L = I - D^{-1}W$
  - $L_{sym} = D^{-\frac{1}{2}}LD^{-\frac{1}{2}} = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$
- These have similar eigenspaces than the original Laplacian L
- Clustering results tend to be better than with the unnormalized Laplacian

Summary

- Several Clustering methods:
  - Dirichlet process mixture model does not require the number of clusters to be known; full Bayesian
  - Affinity Propagation: iterative approach where exemplars are determined as cluster centers
  - Spectral clustering uses the graph Laplacian and performs an eigenvector analysis
  - Hierarchical approaches can be bottom-up or top-down