Gaussian Mixture Models and **Structure Learning** in Bayesian Networks

Anthony Platanios

EM Algorithm Review

We begin with an arbitrary choice for our parameters and iterate over the following steps, until convergence

E Step: Estimate the values of the unobserved variables using the current parameters

M Step: Use the observed variables along with our estimates of the unobserved variables from the previous E step to compute a maximum likelihood estimate for our parameters and update them

Guaranteed to find a local maximum

EM Algorithm Review

Given a joint distribution $p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$ over observed variables $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$, parameterized by $\boldsymbol{\theta}$, we want to maximize $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$

Initialization Choose $\boldsymbol{\theta}^{\text{old}}$

E Step Calculate $p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}})$

M Step Solve $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}^{\text{old}}}\{\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\}$

Convergence If not converged, set $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$

Guaranteed to find a local maximum

EM Algorithm Review

So far we have introduced EM as an algorithm for performing inference in cases where we have **partially labeled data**

However, EM can be used in many more cases, including having **no labeled data at all**

We are now going to consider one such cases as an example

Unsupervised EM Example

Let's consider a case with no labeled data at all

Classification

e.g., Naive Bayes and Logistic Regression

Unsupervised EM Example

Let's consider a case with no labeled data at all

Classification → **Clustering**

We want to
learn the classes
themselves

Clustering

An instance of **unsupervised learning**

Examples

- Find interesting groups of patients in a hospital, faces in photos, webpages, etc.
- Find interesting topics that different documents talk about based on word distributions (i.e., **topic modeling**)

A way of doing that is using **mixtures of distributions**

Mixture of Distributions

We model the joint distribution of our observations using a mixture of multiple distributions — each observation comes from one of those distributions and the **distributions define our clusters**

$$p(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N) = \prod_{n=1}^{N} \sum_{k=1}^{K} p(\boldsymbol{x}^n \mid z^n) p(z^n = k)$$

Discrete (and **unobserved**) random variable that specifies which distribution each observation came from

Cool Fact: That is what happens in Naive Bayes too

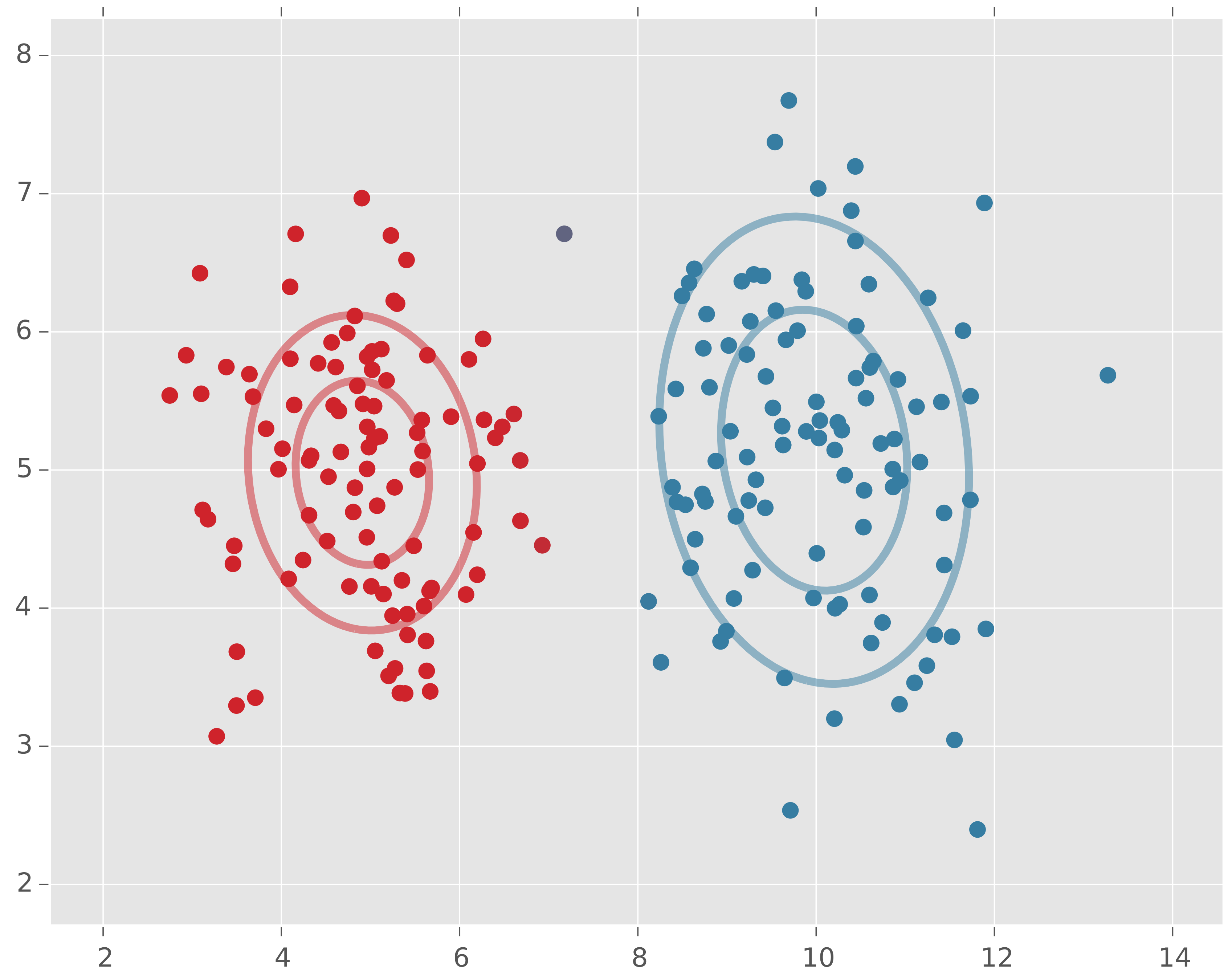
Gaussian Mixture Models (GMM)

We assume that each observation is generated in the following way:

1. Randomly choose a Gaussian distribution according to $p(z^n = k)$
2. Sample the observation from that Gaussian distribution — that Gaussian distribution defines $p(\boldsymbol{x}^n \mid z^n)$

What does this **look like** and
how do we **formalize** it?

GMM Example



GMM Formal Definition

Let's assume we have K clusters. We define a variable indicating which cluster each observation belongs to using a **one-hot vector**

$$\boldsymbol{z} = [0, 0, 0, \ldots, 0, 1, 0, \ldots, 0, 0]$$

$z_k = 1$ if and only if our observation belongs to cluster k

This will be the **unobserved latent variable** of our model, corresponding to what used to be a class for each observation (in classification problems)

GMM Formal Definition

We define the joint distribution as follows

$$p(\boldsymbol{x}^n, \boldsymbol{z}^n) = \underline{p(\boldsymbol{x}^n \mid \boldsymbol{z}^n)}p(\boldsymbol{z}^n)$$

$$p(\boldsymbol{x}^n \mid z_k^n = 1) = \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \quad \Rightarrow$$

$$p(\boldsymbol{x}^n \mid \boldsymbol{z}^n) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k^n}$$

Each cluster has its own Gaussian distribution and the exponent picks the one corresponding to the cluster to which the current observation belongs

GMM Formal Definition

We define the joint distribution as follows

$$p(\boldsymbol{x}^n, \boldsymbol{z}^n) = p(\boldsymbol{x}^n \mid \boldsymbol{z}^n)\underline{p(\boldsymbol{z}^n)}$$

$$p(z_k^n = 1) = \pi_k \quad \Rightarrow \quad p(\boldsymbol{z}^n) = \prod_{k=1}^{K} \pi_k^{z_k^n}$$

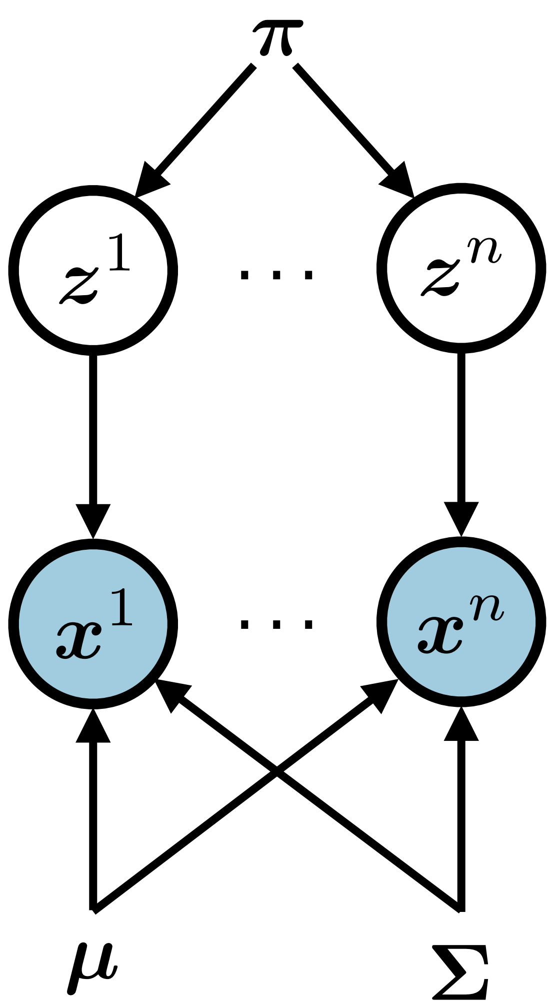
probability of the corresponding cluster

$$\sum_{k=1}^{K} \pi_k = 1$$

because only one cluster can be "active" at any time

GMM Formal Definition

What does our model look like?

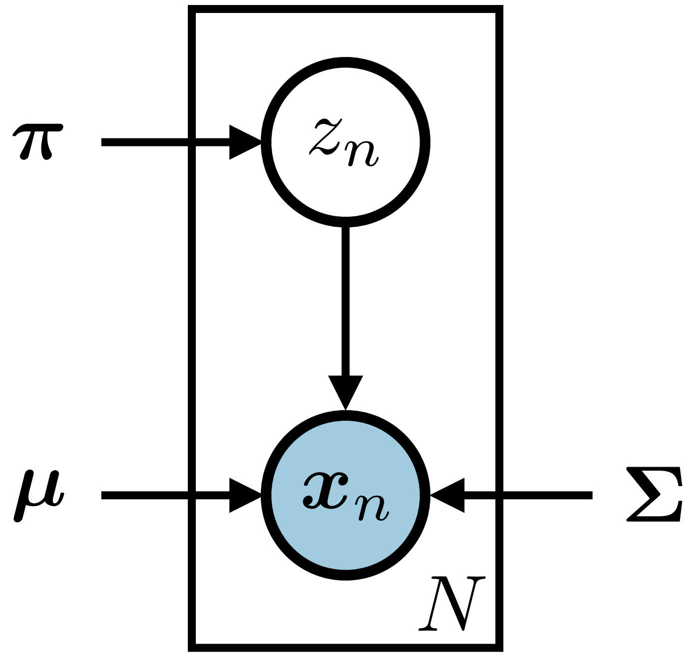


$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$$

GMM Formal Definition

What does our model look like?

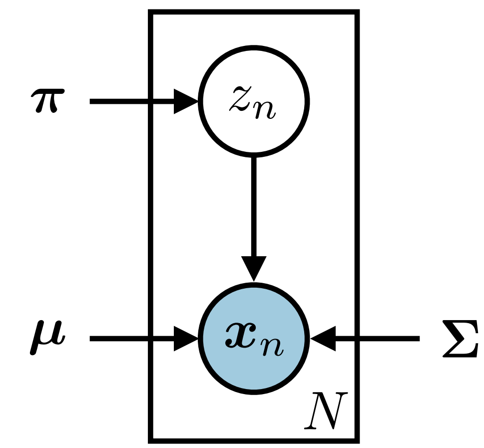


$$p(\boldsymbol{x} \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

$$\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$$

The plate notation simply means that we have N copies of the variables indexed by n

GMM - EM Algorithm



Remember these steps?

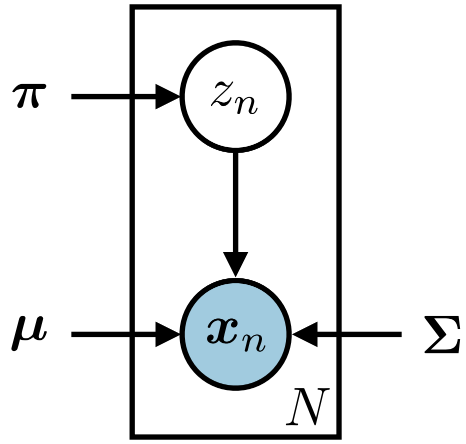
Initialization Choose $\boldsymbol{\theta}^{\mathrm{old}}$

E Step Calculate the expected value of the cluster assignments

M Step Calculate the maximum likelihood estimate of all parameters given the expected value of the cluster assignments

Convergence If not converged, set $\boldsymbol{\theta}^{\mathrm{old}} \leftarrow \boldsymbol{\theta}^{\mathrm{new}}$

GMM - EM Algorithm



Remember these steps?

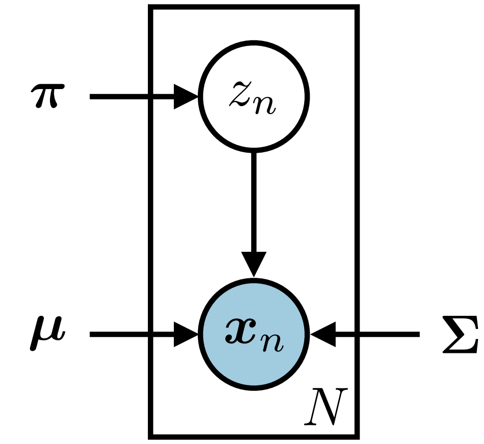
Initialization Choose $\boldsymbol{\theta}^{\text{old}}$

E Step Calculate $p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}})$

M Step Solve $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}^{\text{old}}} \{\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\}$

Convergence If not converged, set $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$

GMM - E Step



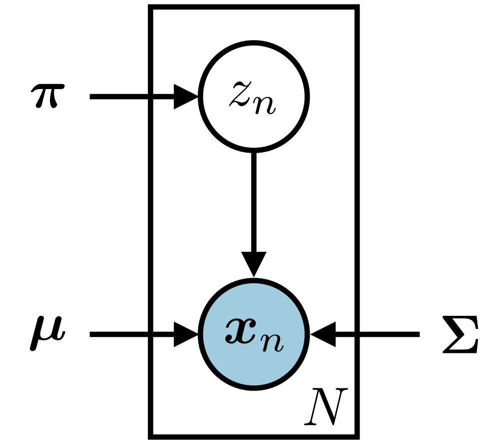
Calculate $p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}})$

$$\underbrace{p(z_k^n = 1 \mid \boldsymbol{x}^n, \boldsymbol{\theta})}_{r(z_k^n)} = \frac{p(z_k^n = 1)p(\boldsymbol{x}^n \mid z_k^n = 1)}{\sum_{k'=1}^K p(z_{k'}^n = 1)p(\boldsymbol{x}^n \mid z_{k'}^n = 1)}$$

$$= \frac{\pi_k \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^K \pi_{k'} \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$$

We can think of that quantity as the **responsibility** that the corresponding mixture component takes for “explaining” observation $\boldsymbol{x}^n$

GMM - M Step



Solve $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}}} \{\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\}$

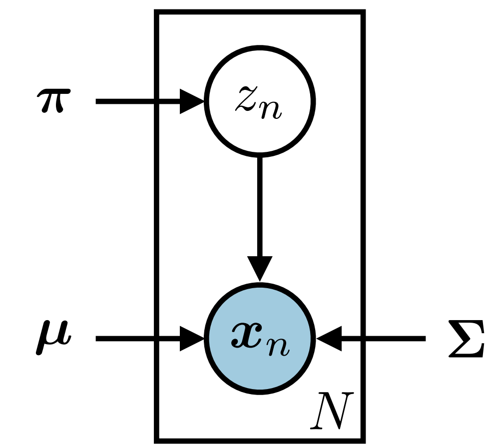
$$p(\boldsymbol{x}^n \mid \boldsymbol{z}^n) = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k^n} \qquad p(\boldsymbol{z}^n) = \prod_{k=1}^{K} \pi_k^{z_k^n}$$

$$p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \prod_{n=1}^{N} \prod_{k=1}^{K} \left[\pi_k \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]^{z_k^n}$$

$$\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_k^n \log \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + z_k^n \log \pi_k$$

$$\mathbb{E}_{\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}}} \{\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\} = \sum_{n=1}^{N} \sum_{k=1}^{K} r(z_k^n) \log \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + r(z_k^n) \log \pi_k$$

GMM - M Step



Solve $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z}|\boldsymbol{x},\boldsymbol{\theta}^{\text{old}}} \{\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\}$

$$\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \underbrace{\sum_{n=1}^{N} \sum_{k=1}^{K} r(z_k^n) \log \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + r(z_k^n) \log \pi_k}_{\mathcal{L}(\boldsymbol{\theta})}$$

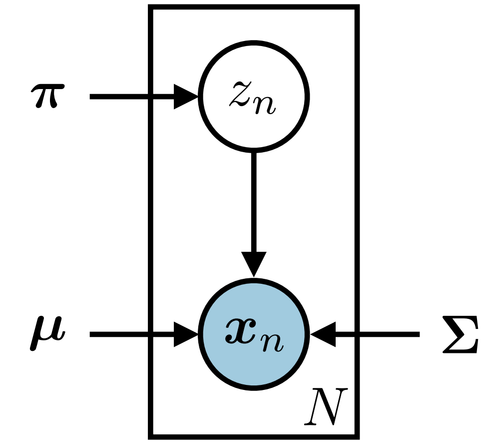
$$\sum_{k=1}^{K} \pi_k = 1$$

We also have a constraint and we will use a trick called a **Lagrange multiplier** to make sure it is satisfied while optimizing the likelihood. We will change our maximization objective to the following, for some λ

$$f(\boldsymbol{\theta}) = \mathcal{L}(\boldsymbol{\theta}) - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

Penalty for violating the constraint

GMM - M Step



$$f(\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r(z_k^n) \log \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + r(z_k^n) \log \pi_k - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

Solve for π_k

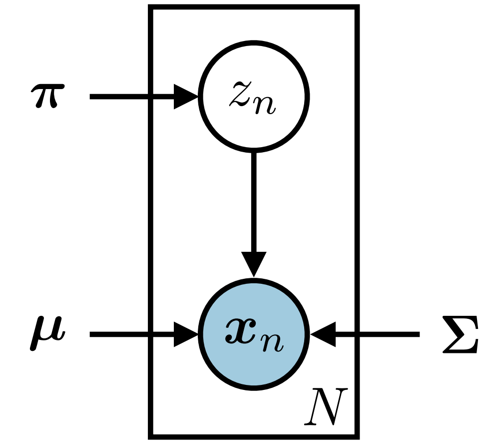
$$\frac{\partial f(\boldsymbol{\theta})}{\partial \lambda} = 1 - \sum_{k=1}^{K} \pi_k = 0 \Rightarrow \sum_{k=1}^{K} \pi_k = 1$$

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \pi_k} = \frac{\sum_{n=1}^{N} r(z_k^n)}{\pi_k} - \lambda = 0 \Rightarrow \pi_k = \frac{\sum_{n=1}^{N} r(z_k^n)}{\lambda} = \frac{N_k}{\lambda}$$

Effective number of samples for which this mixture component is **responsible** for

$$\lambda = \sum_{k=1}^{K} \sum_{n=1}^{N} r(z_k^n) = N \longrightarrow \pi_k = \frac{N_k}{N}$$

GMM - M Step



$$f(\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} r(z_k^n) \log \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + r(z_k^n) \log \pi_k - \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

Solve for $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$

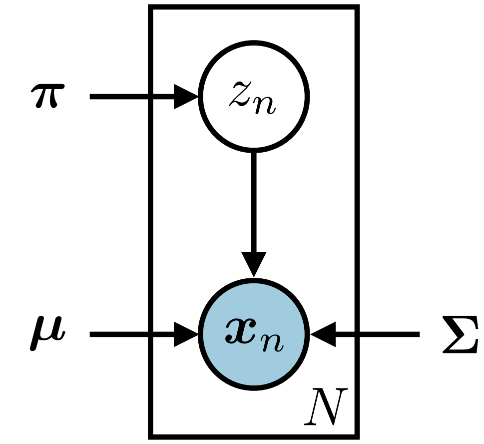
$$\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\mu}_k} = \sum_{n=1}^{N} r(z_k^n) \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}^n - \boldsymbol{\mu}_k) = 0 \Rightarrow \boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r(z_k^n) \boldsymbol{x}^n$$

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \boldsymbol{\Sigma}_k} = \frac{1}{2} \sum_{n=1}^{N} r(z_k^n) \boldsymbol{\Sigma}_k^{-1} \left[(\boldsymbol{x}^n - \boldsymbol{\mu}_k)(\boldsymbol{x}^n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} - 1 \right] = 0 \Rightarrow$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} r(z_k^n) (\boldsymbol{x}^n - \boldsymbol{\mu}_k)(\boldsymbol{x}^n - \boldsymbol{\mu}_k)^\top$$

Note that if the clusters were observed, then all the responsibilities would be indicator functions

GMM - EM Algorithm



Initialization Choose initial values for $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$

E Step Compute $r(z_k^n) = \dfrac{\pi_k \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\boldsymbol{x}^n \mid \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'})}$

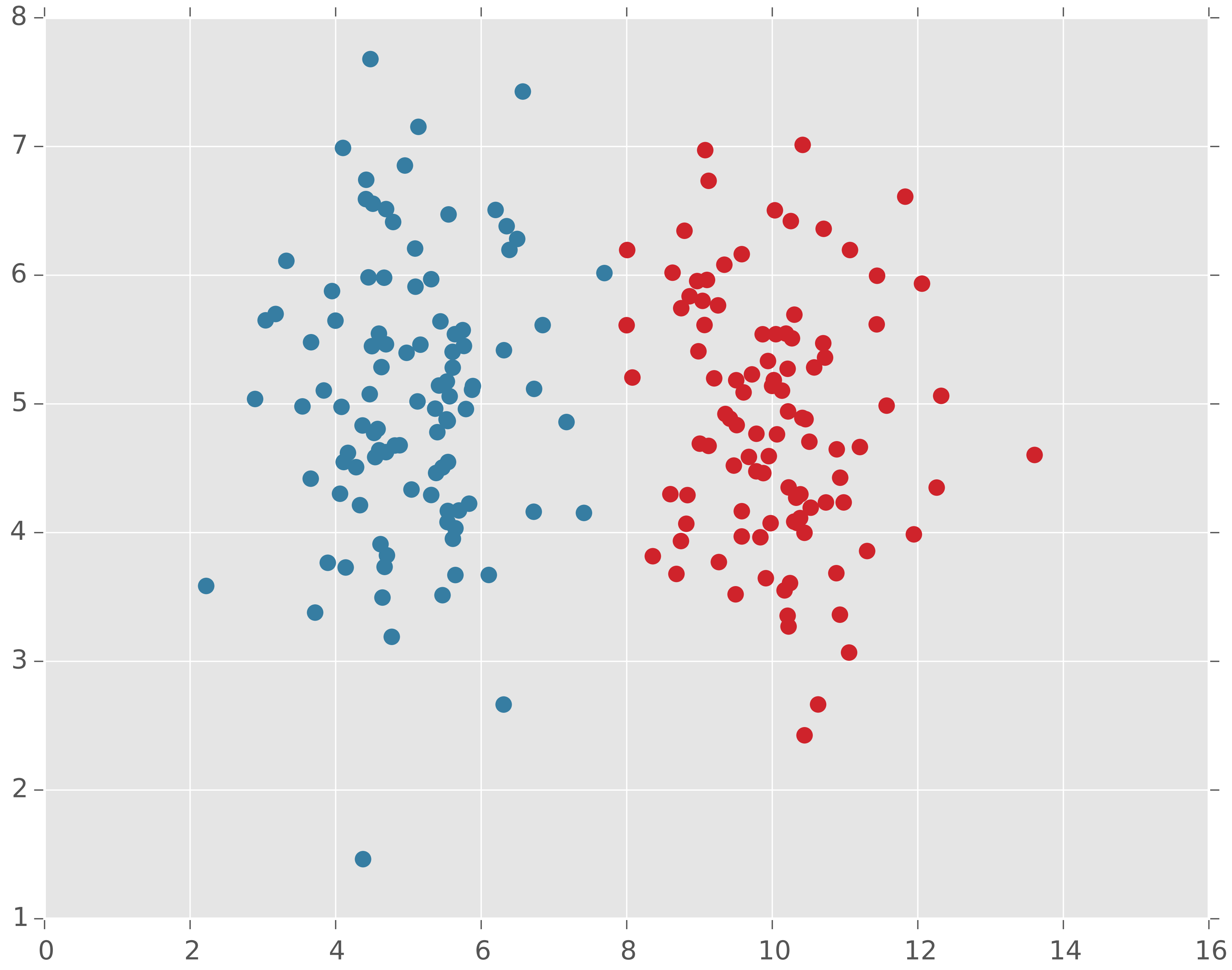
M Step Compute $\pi_k = \dfrac{N_k}{N}$ $\boldsymbol{\mu}_k = \dfrac{1}{N_k} \sum_{n=1}^{N} r(z_k^n) \boldsymbol{x}^n$

$$N_k = \sum_{n=1}^{N} r(z_k^n)$$

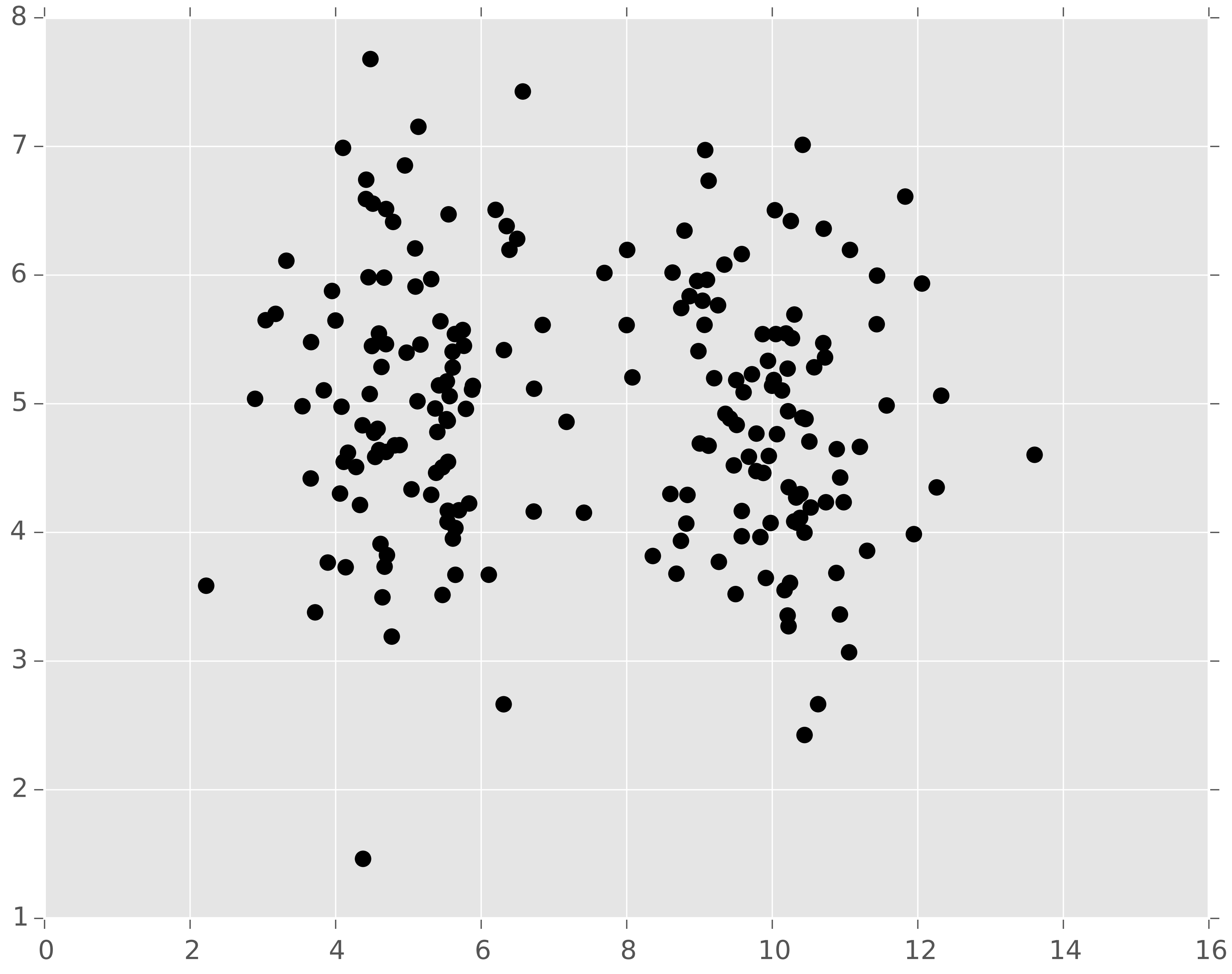
$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^{N} r(z_k^n)(\boldsymbol{x}^n - \boldsymbol{\mu}_k)(\boldsymbol{x}^n - \boldsymbol{\mu}_k)^\top$$

Convergence Iterate until the log-likelihood converges

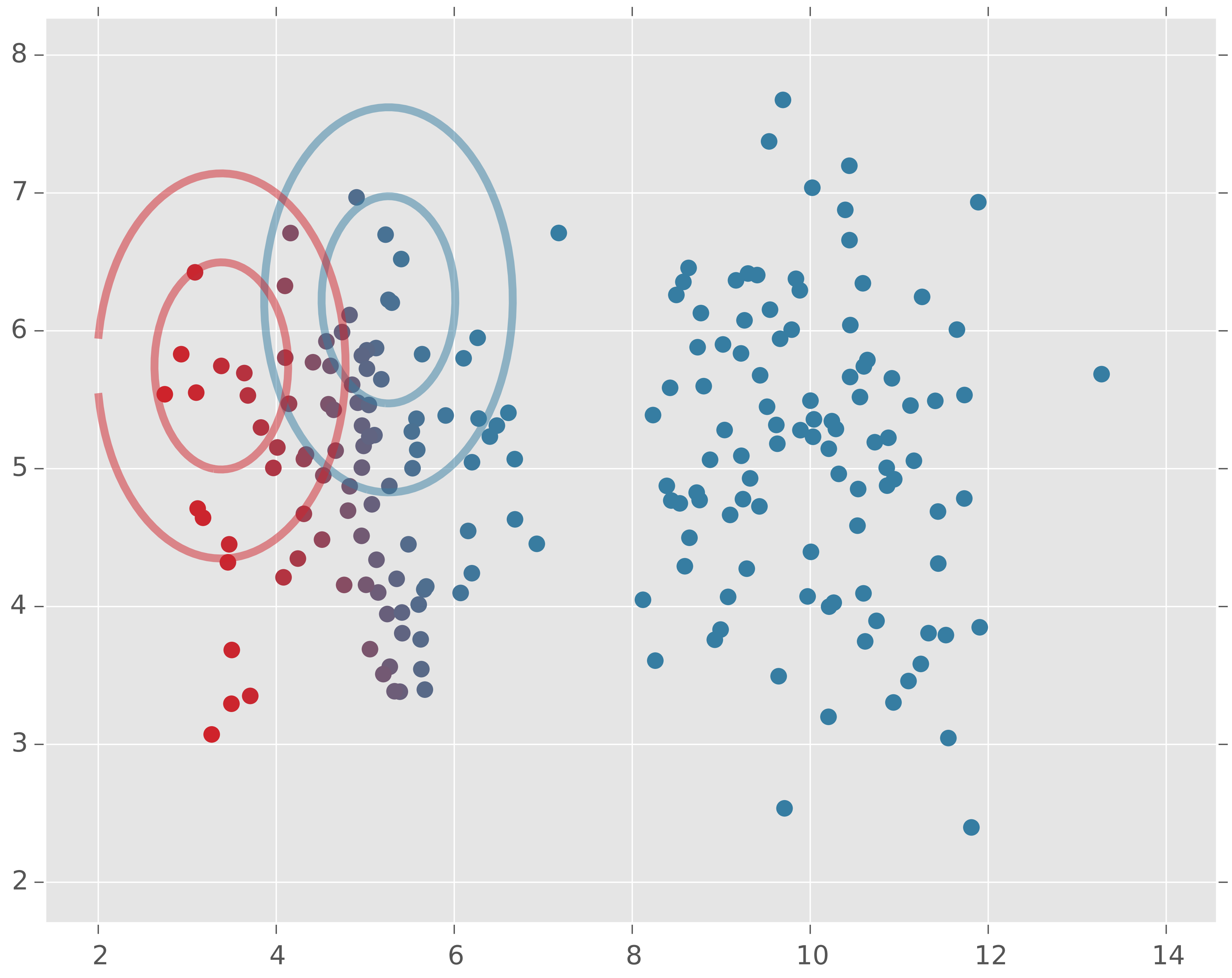
GMM Example - Data Set



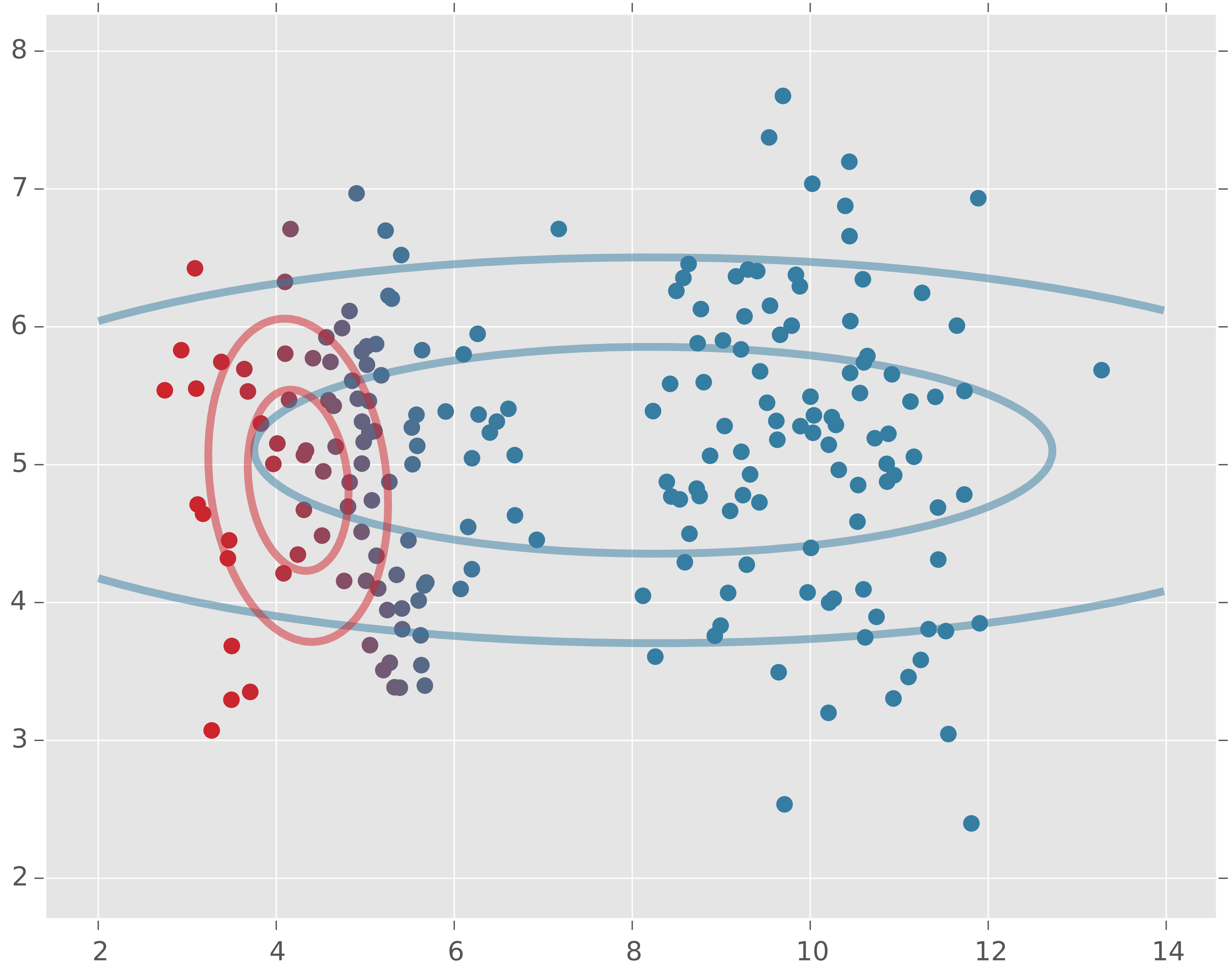
GMM Example - Data Set



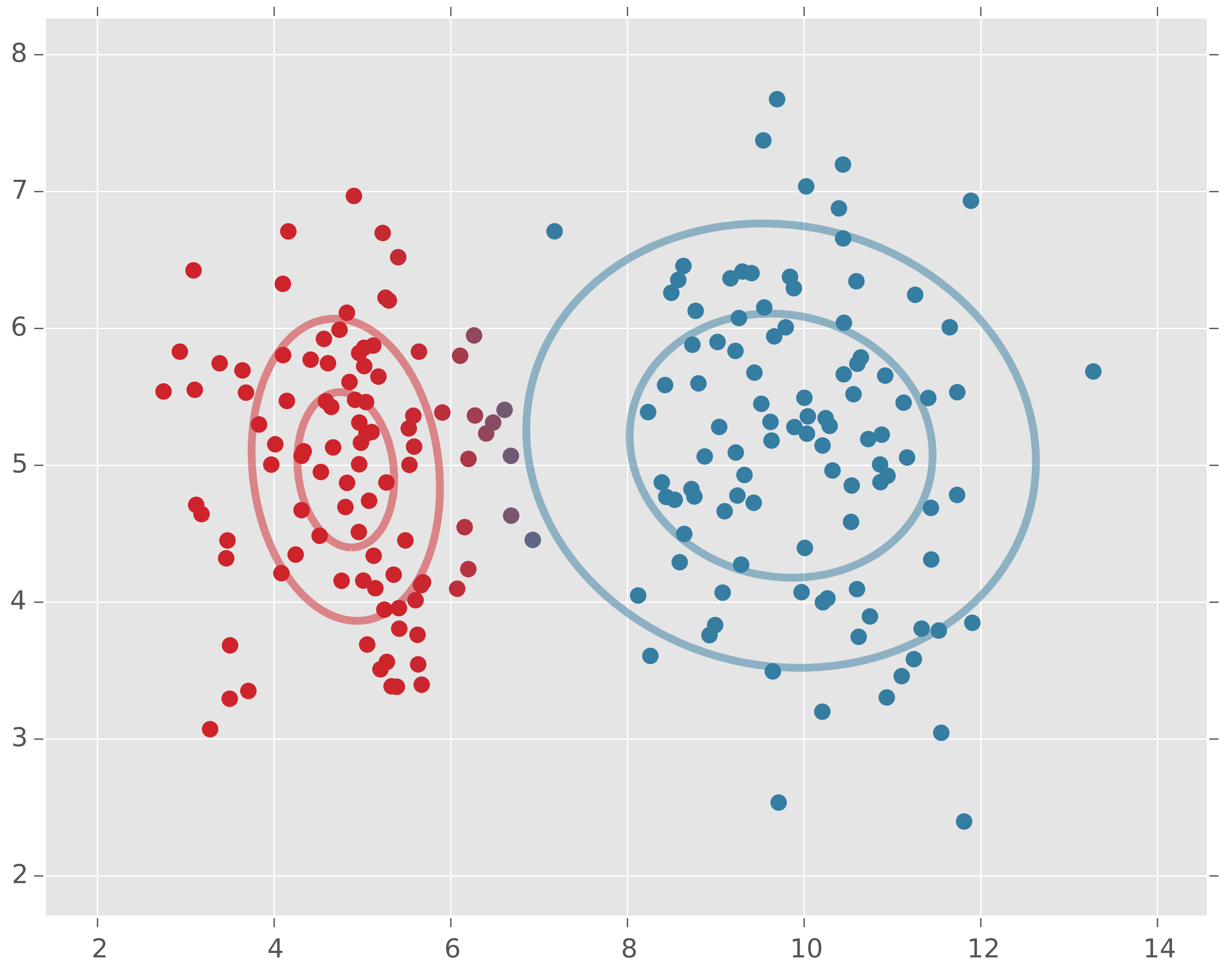
GMM Example - Iteration 1



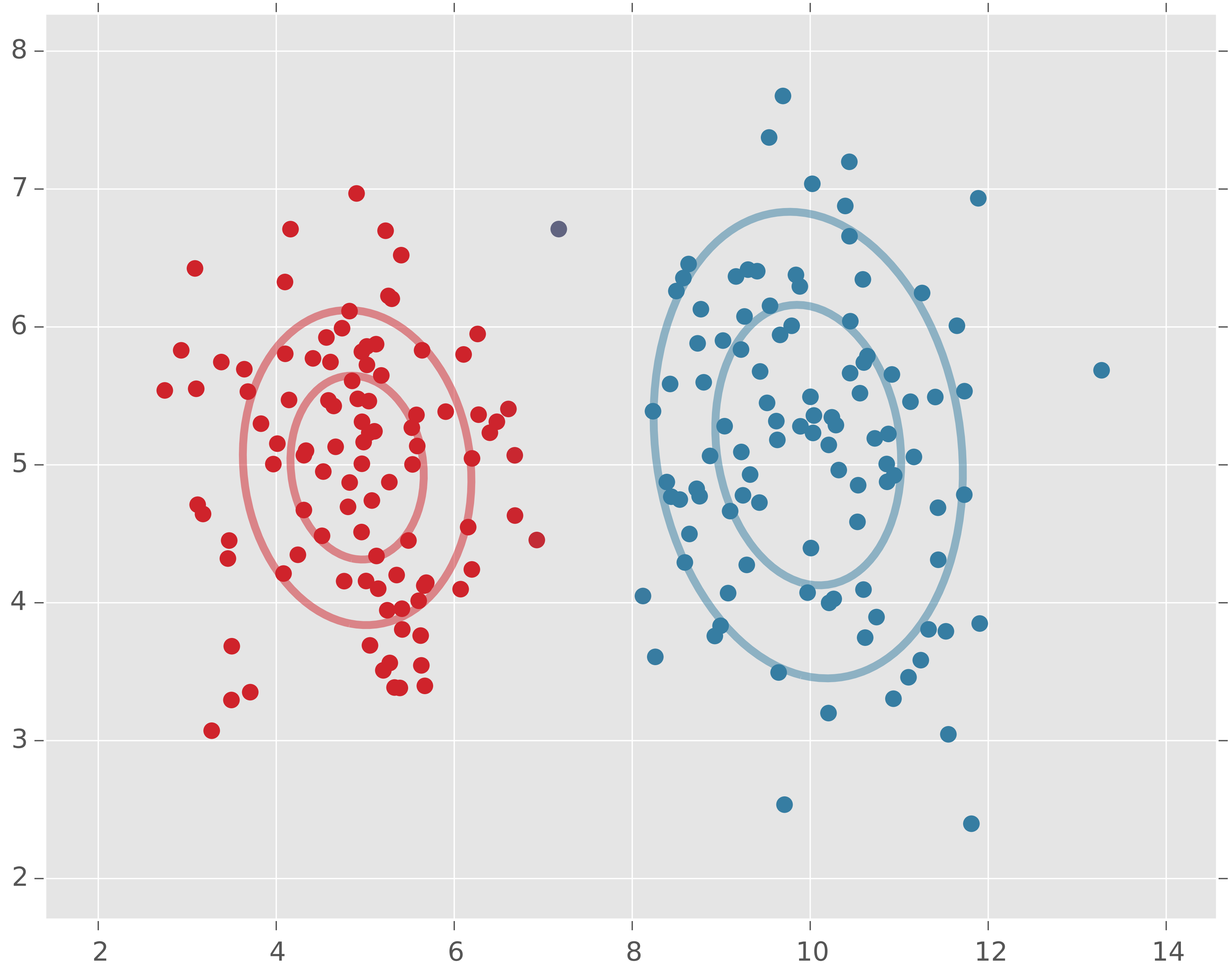
GMM Example - Iteration 2



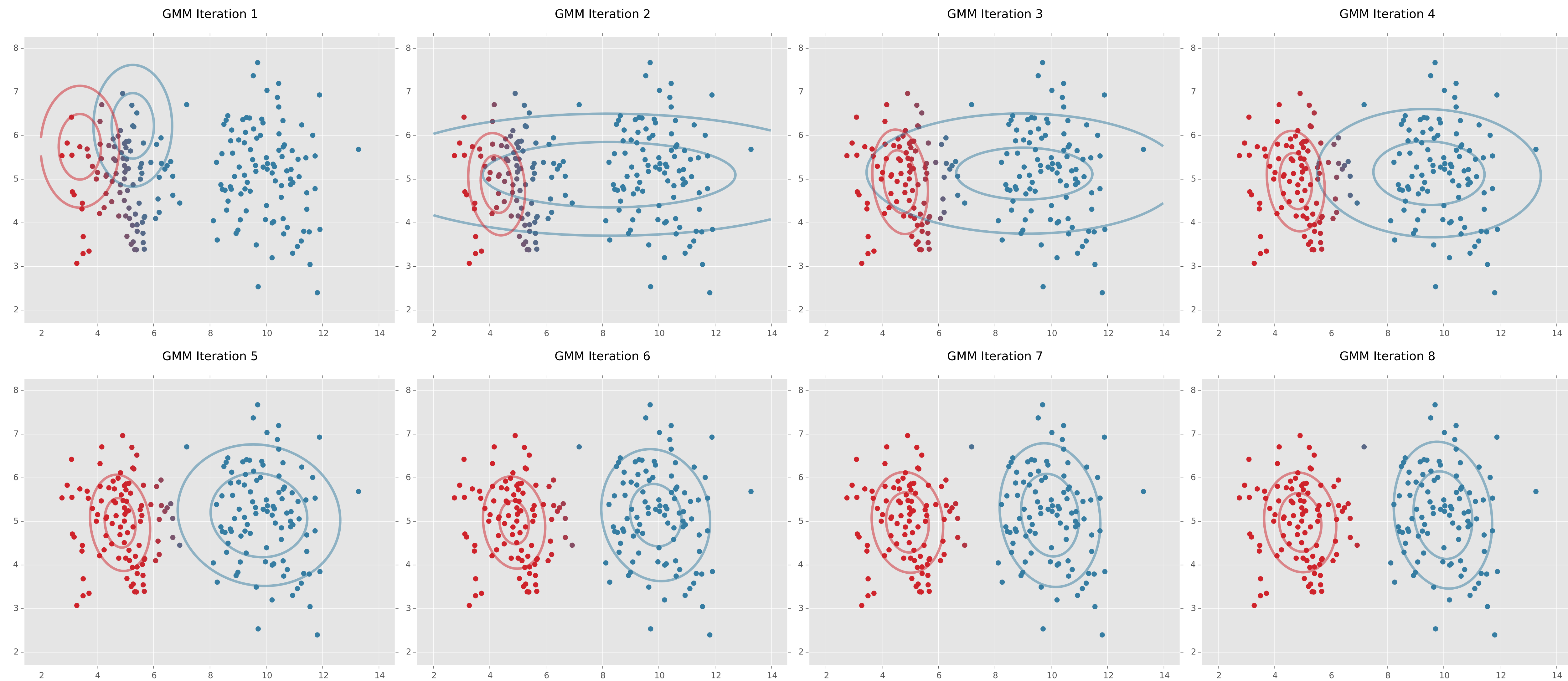
GMM Example - Iteration 5

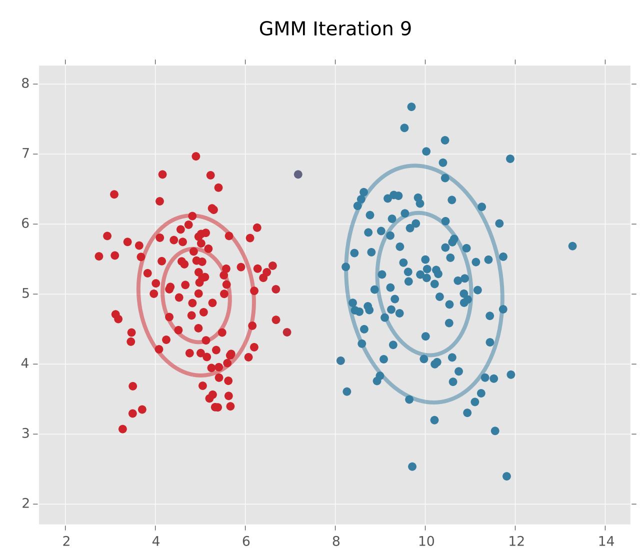


GMM Example - Converged



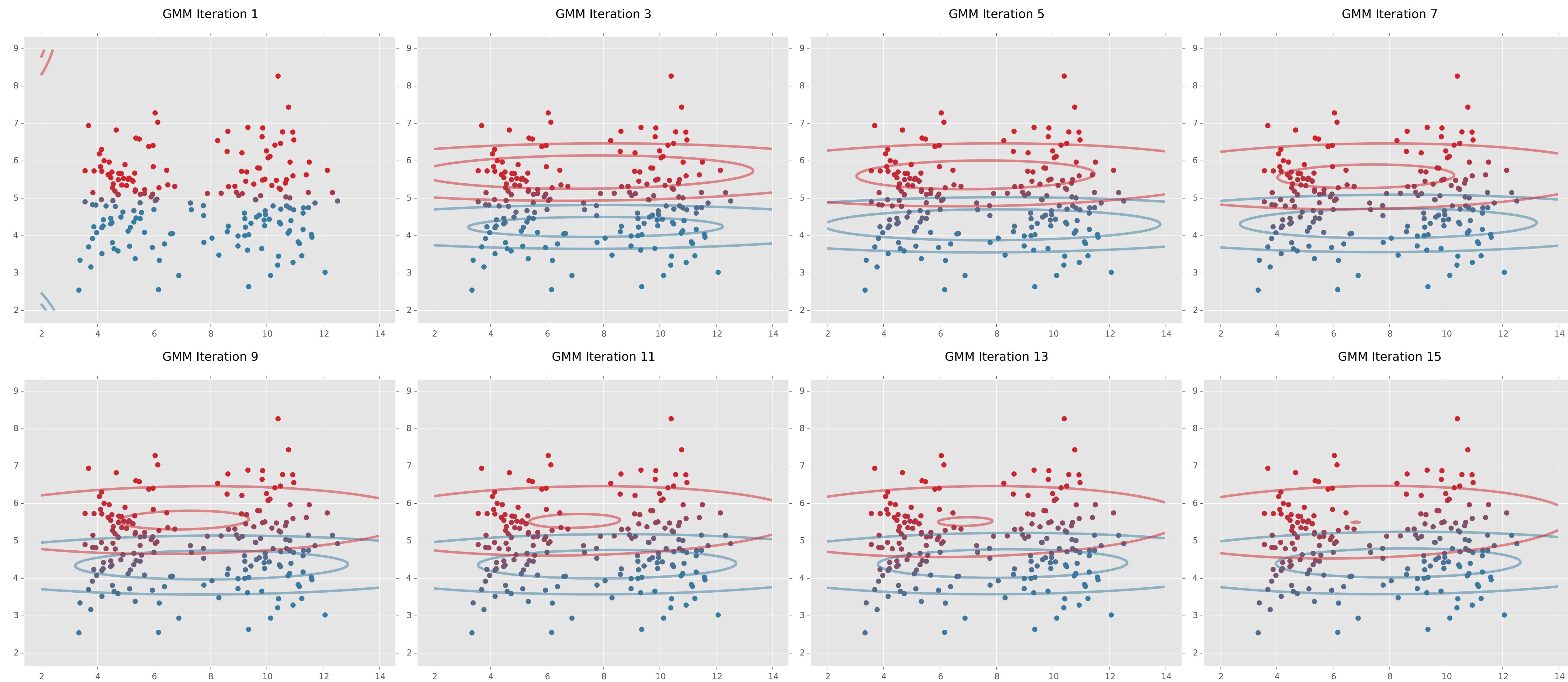
GMM Example - Summary

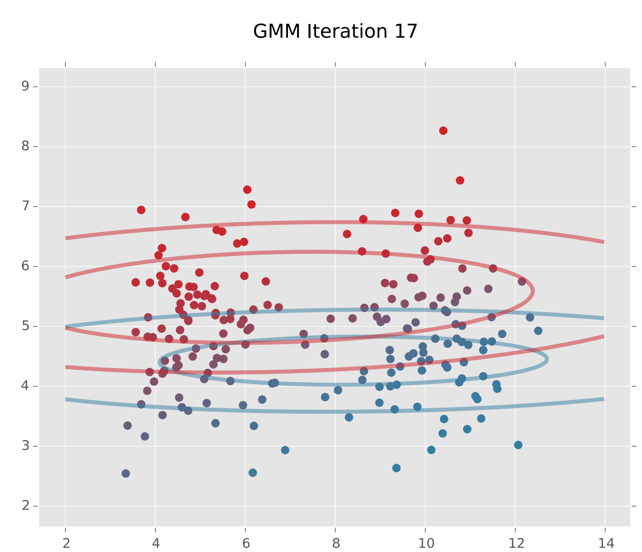




Pretty good! However, initialization matters…remember that EM only guarantees a **local maximum**

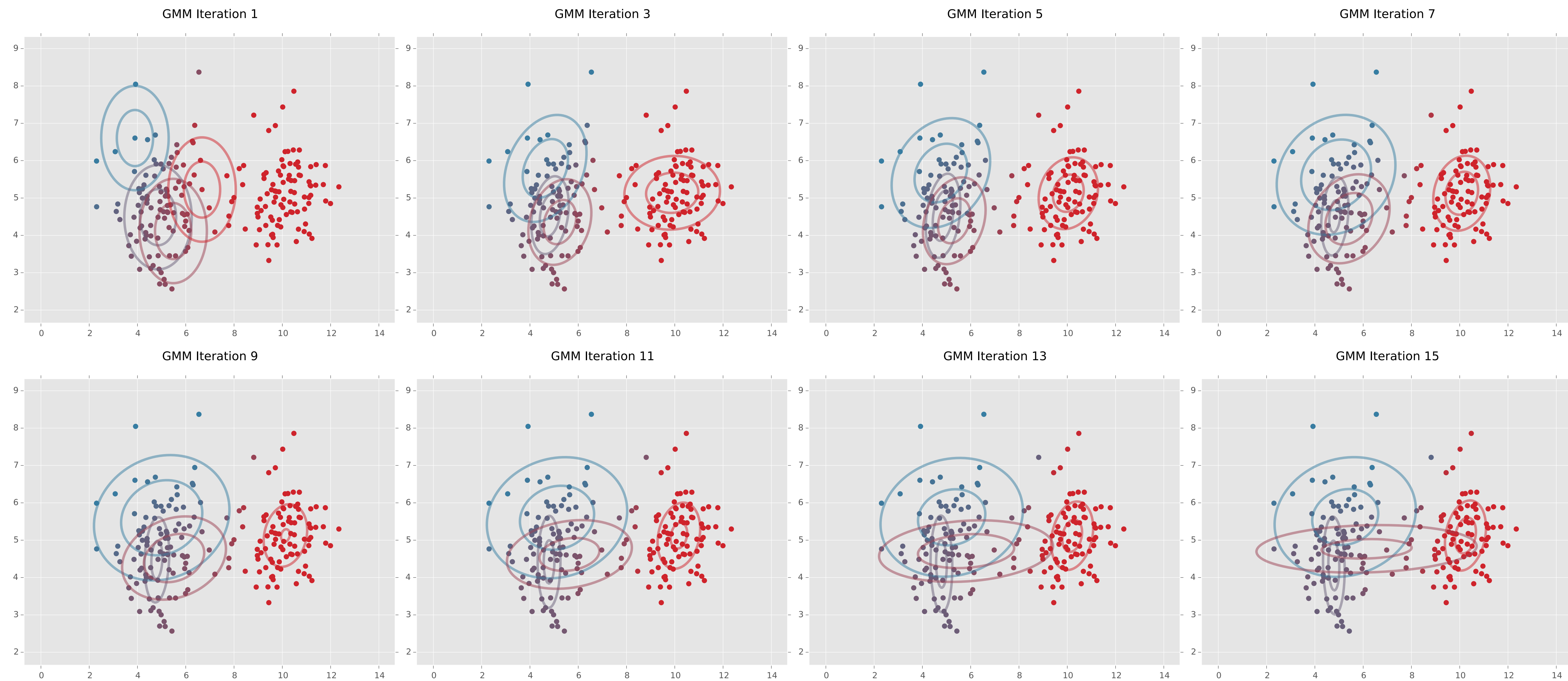
GMM Example - Bad Initialization

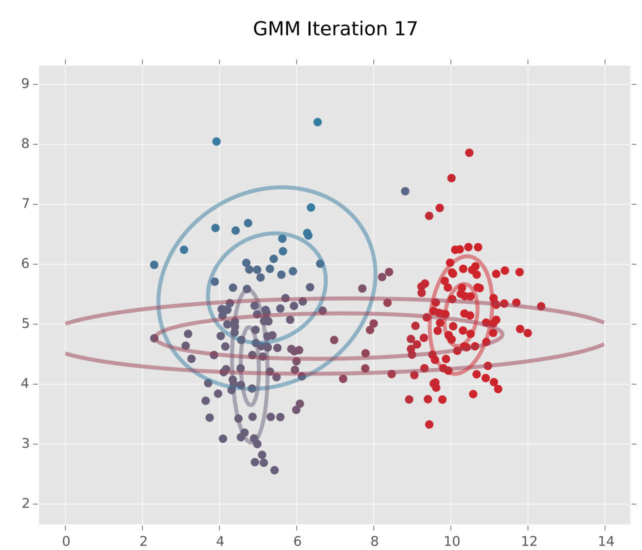




However, there are ways
to deal with that…
e.g., **k-means++**

GMM Example - Number of Clusters





However, there are ways to deal with that too…
e.g., **nonparametric models**

A Small Variation to GMM

For GMMs we had the following form

$$p(\boldsymbol{x}^n, \boldsymbol{z}^n) = p(\boldsymbol{x}^n \mid \boldsymbol{z}^n)\underline{p(\boldsymbol{z}^n)}$$

$$p(z_k^n = 1) = \pi_k \quad \Rightarrow \quad p(\boldsymbol{z}^n) = \prod_{k=1}^{K} \pi_k^{z_k^n}$$

probability of the corresponding cluster

$$\sum_{k=1}^{K} \pi_k = 1$$

because only one cluster can be “active” at any time

A Small Variation to GMM

What if we change it to this?

$$p(\boldsymbol{x}^n, \boldsymbol{z}^n) = p(\boldsymbol{x}^n \mid \boldsymbol{z}^n)\underline{p(\boldsymbol{z}^n)}$$

$$p(z_k^n = 1) = \mathbb{1}_{\{n = \arg\min_{n'} \|\boldsymbol{x}^{n'} - \boldsymbol{\mu}_k\|_2\}} \quad \Rightarrow \quad p(\boldsymbol{z}^n) = \prod_{k=1}^{K} \pi_k^{z_k^n}$$

The expectation is now simply equal to this indicator and the E step of EM simply sets the cluster of an observation to the cluster with mean closest to that observation

A Small Variation to GMM

If we also **fix the covariance matrix to be the identity matrix**, the we get the following algorithm

Initialization Initialize the cluster means arbitrarily

E Step Compute $r(z_k^n) = \mathbb{1}_{\{n = \arg\min_{n'} \|\boldsymbol{x}^{n'} - \boldsymbol{\mu}_k\|_2\}}$

M Step Compute $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r(z_k^n) \boldsymbol{x}^n$ $N_k = \sum_{n=1}^{N} r(z_k^n)$

Convergence Iterate until the means converge

k-Means Algorithm

If we also **fix the covariance matrix to be the identity matrix**, the we get the following algorithm

Initialization Initialize the cluster means arbitrarily

E Step Compute $r(z_k^n) = \mathbb{1}_{\{n = \arg\min_{n'} \|\boldsymbol{x}^{n'} - \boldsymbol{\mu}_k\|_2\}}$

M Step Compute $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^{N} r(z_k^n) \boldsymbol{x}^n \quad N_k = \sum_{n=1}^{N} r(z_k^n)$

Convergence Iterate until the means converge

This is just another way to view the famous k-means clustering algorithm from an EM perspective

EM Algorithm Recap

Given a joint distribution $p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})$ over observed variables $\boldsymbol{x}$ and latent variables $\boldsymbol{z}$, parameterized by $\boldsymbol{\theta}$, we want to maximize $p(\boldsymbol{x} \mid \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$

Initialization Choose $\boldsymbol{\theta}^{\text{old}}$

E Step Calculate $p(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}})$

M Step Solve $\boldsymbol{\theta}^{\text{new}} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{\theta}^{\text{old}}} \{\log p(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{\theta})\}$

Convergence If not converged, set $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$

Guaranteed to find a local maximum

EM Algorithm Recap

EM can be used in any probabilistic model and not just Bayesian networks — undirected graphical models are an example

Another **approximate inference** method for probabilistic models is **variational inference** and it is actually a generalization of EM

Bayesian Network **Structure Learning**

Learning structure is not that easy

- In general requires lots of data (can **overfit** easily)
- **Huge search space** — we use priors to constrain it

But there exist some algorithms for certain special cases (e.g., **Chow-Liu** for **tree structures**)

Chow-Liu Algorithm

Finds the **"best" tree-structured** Bayesian network

We have the following random variables

Let the true distribution be

$$p(\boldsymbol{X}) = p(X_1, \ldots, X_N)$$

Let our tree-structured approximate distribution be

$$q(\boldsymbol{X}) = q(X_1, \ldots, X_N)$$

Chow-Liu finds the $q(\boldsymbol{X})$ that minimizes KL divergence with $p(\boldsymbol{X})$

$$\mathrm{KL}(p(\boldsymbol{X}) \,||\, q(\boldsymbol{X})) \triangleq \sum_{\boldsymbol{x}} p(\boldsymbol{X} = \boldsymbol{x}) \log \frac{p(\boldsymbol{X} = \boldsymbol{x})}{q(\boldsymbol{X} = \boldsymbol{x})}$$

Chow-Liu Algorithm

Notice that

$$q(\boldsymbol{x}) = \prod_{i=1}^{N} p(x_i \mid \mathrm{Pa}(x_i))$$

Tree Structure

$$\begin{aligned}
\mathrm{KL}(p(\boldsymbol{X}) \mid\mid q(\boldsymbol{X})) &= \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} \\
&= \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(\boldsymbol{x}) - \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log q(\boldsymbol{x}) \\
&= -H(\boldsymbol{X}) - \sum_{i=1}^{N} \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(x_i \mid \mathrm{Pa}(x_i)) \\
&= -H(\boldsymbol{X}) - \sum_{i=1}^{N} \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log p(x_i) + \sum_{i=1}^{N} \sum_{\boldsymbol{x}} p(\boldsymbol{x}) \log \frac{p(x_i)}{p(x_i \mid \mathrm{Pa}(x_i))} \\
&= -H(\boldsymbol{X}) + \sum_{i=1}^{N} H(X_i) \underbrace{- \sum_{i=1}^{N} \mathrm{MI}(X_i, \mathrm{Pa}(X_i))}_{\text{only term that depends on edges}}
\end{aligned}$$

Chow-Liu Algorithm

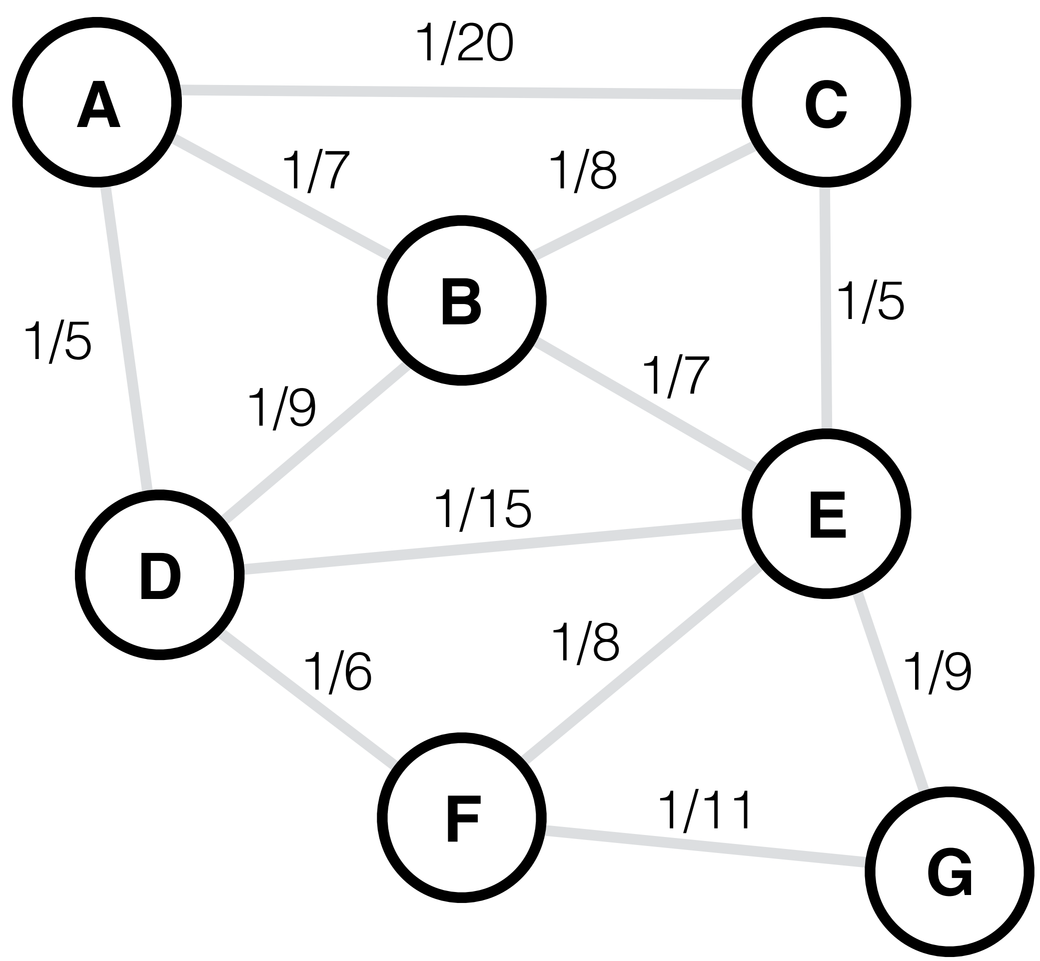
All we need to do is find the tree that maximizes the sum of the mutual information over each edge

$$\sum_{i=1}^{N} \mathrm{MI}(X_i, \mathrm{Pa}(X_i)) = \sum_{i=1}^{N} \sum_{\boldsymbol{x}} p(x_i, \mathrm{Pa}(x_i)) \log \frac{p(x_i, \mathrm{Pa}(x_i))}{p(x_i) p(\mathrm{Pa}(x_i))}$$

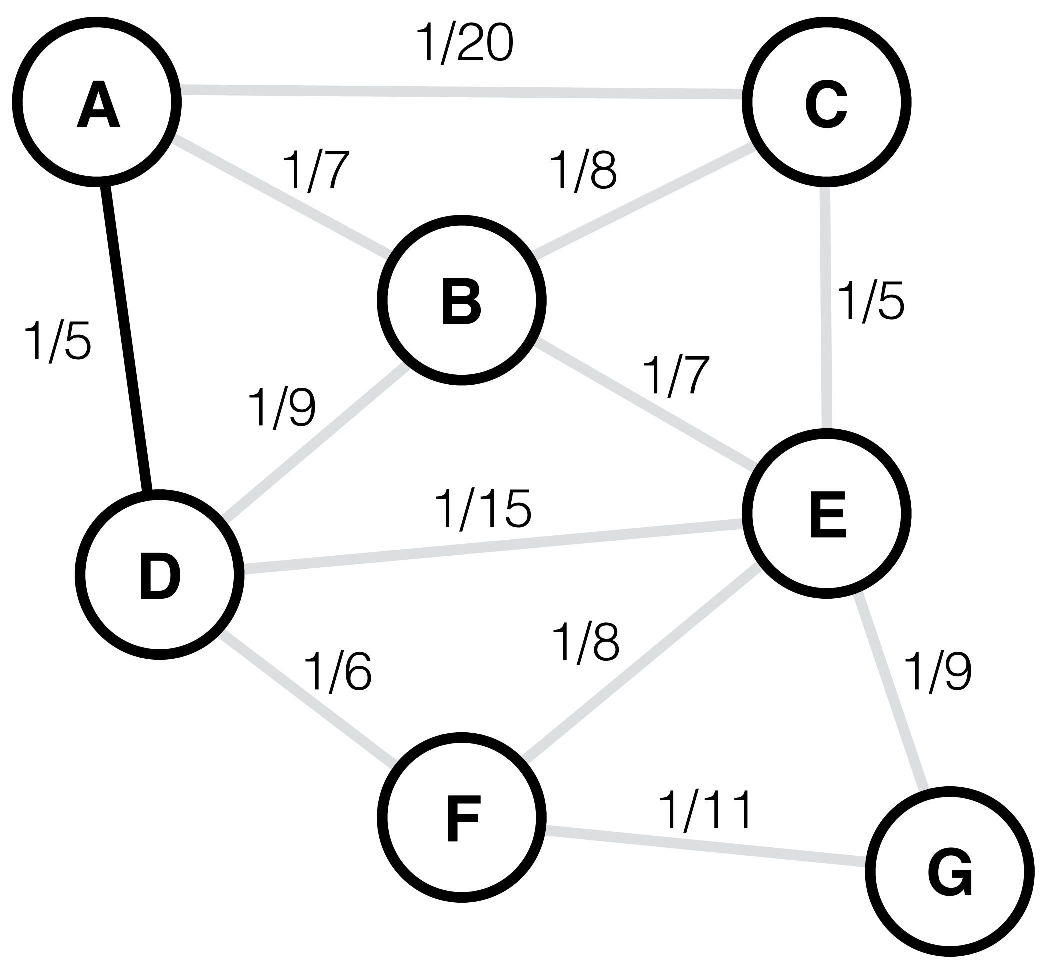
Algorithm

1. For each pair of variables A, B use observations to estimate $p(A)$, $p(B)$, and $p(A, B)$, and calculate the mutual information
2. Compute the maximum spanning tree over all variables using the mutual information of each pair as the corresponding edge weight
3. Add arrows to the edges to form a directed acyclic graph (DAG)
4. Learn the conditional probability tables (CPT) for this graph

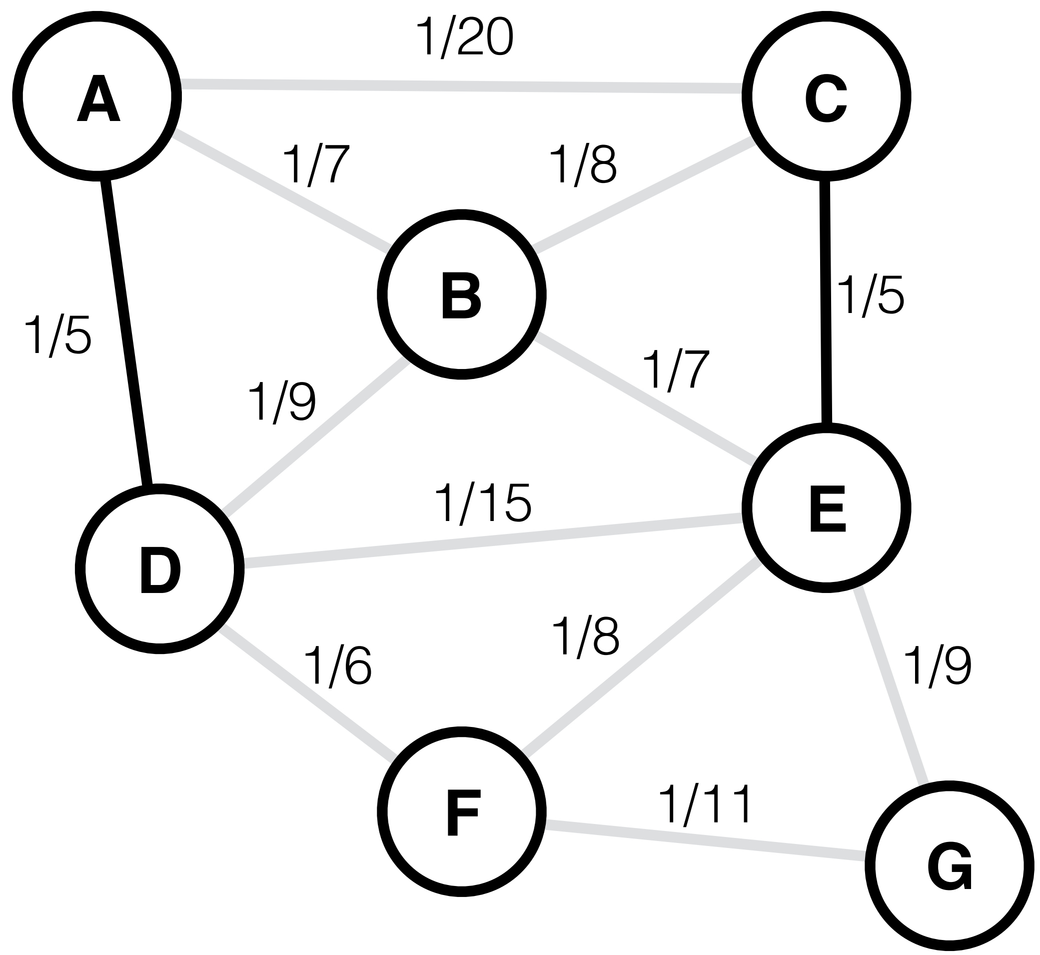
Chow-Liu Example



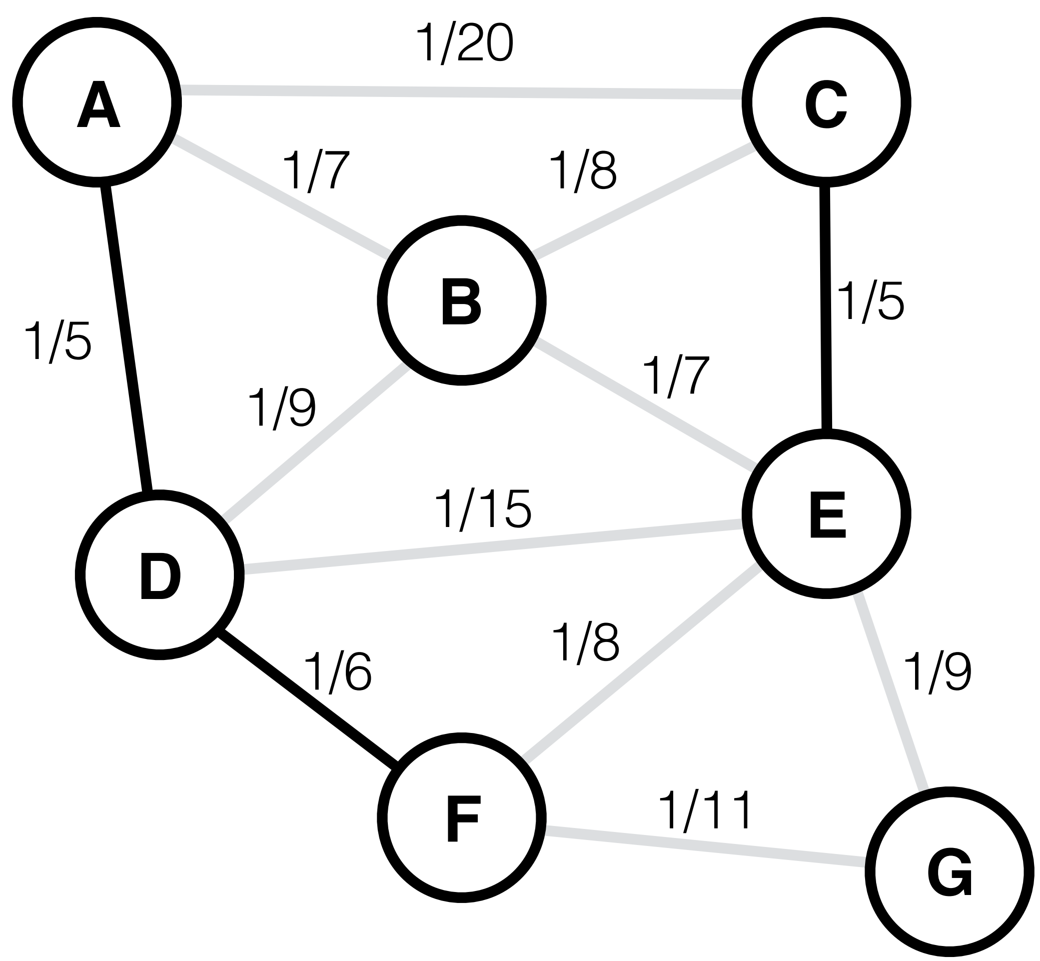
Chow-Liu Example



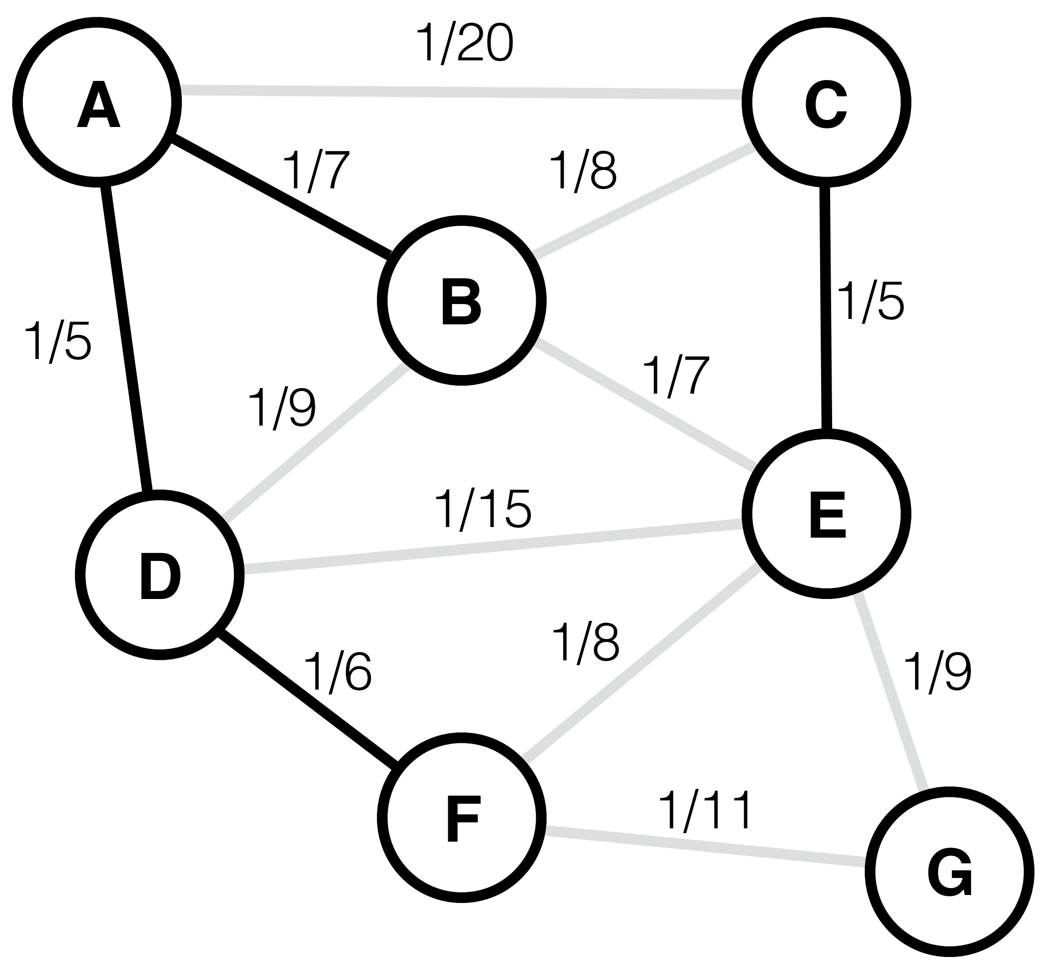
Chow-Liu Example



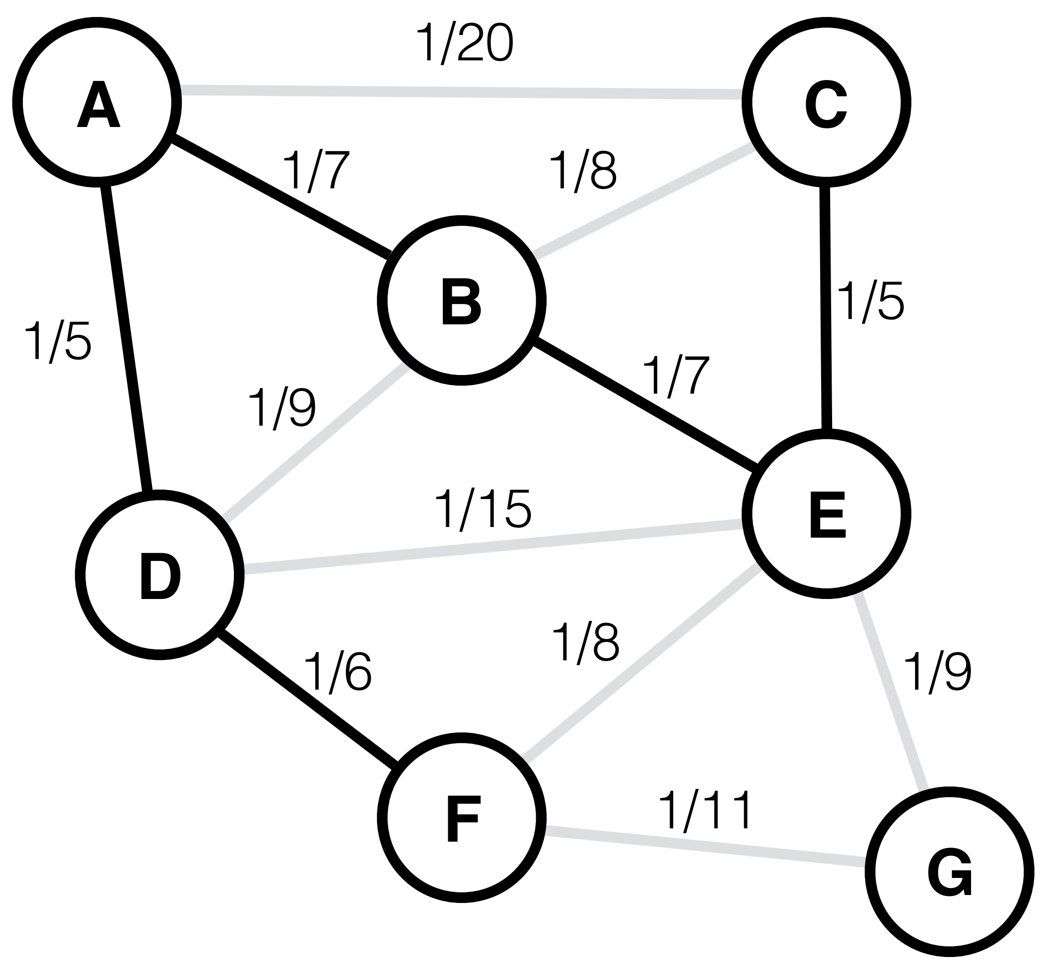
Chow-Liu Example



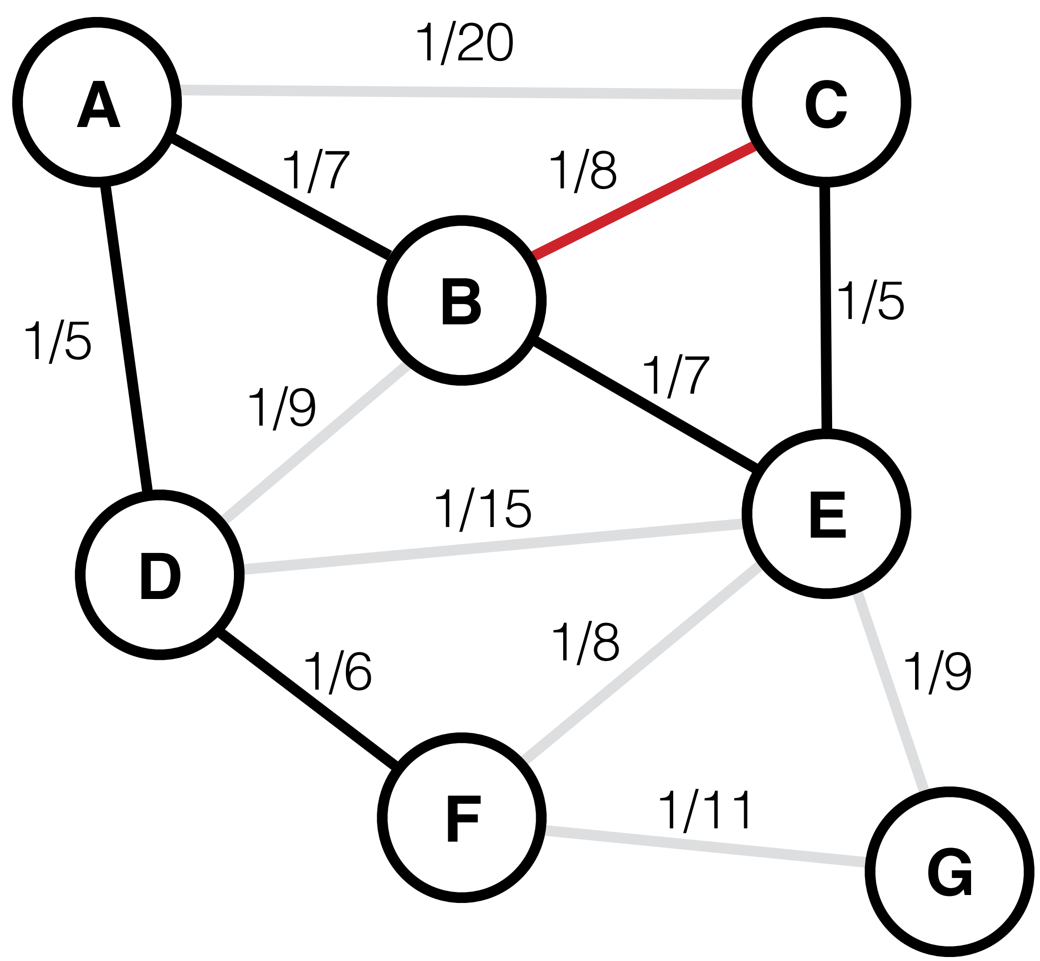
Chow-Liu Example



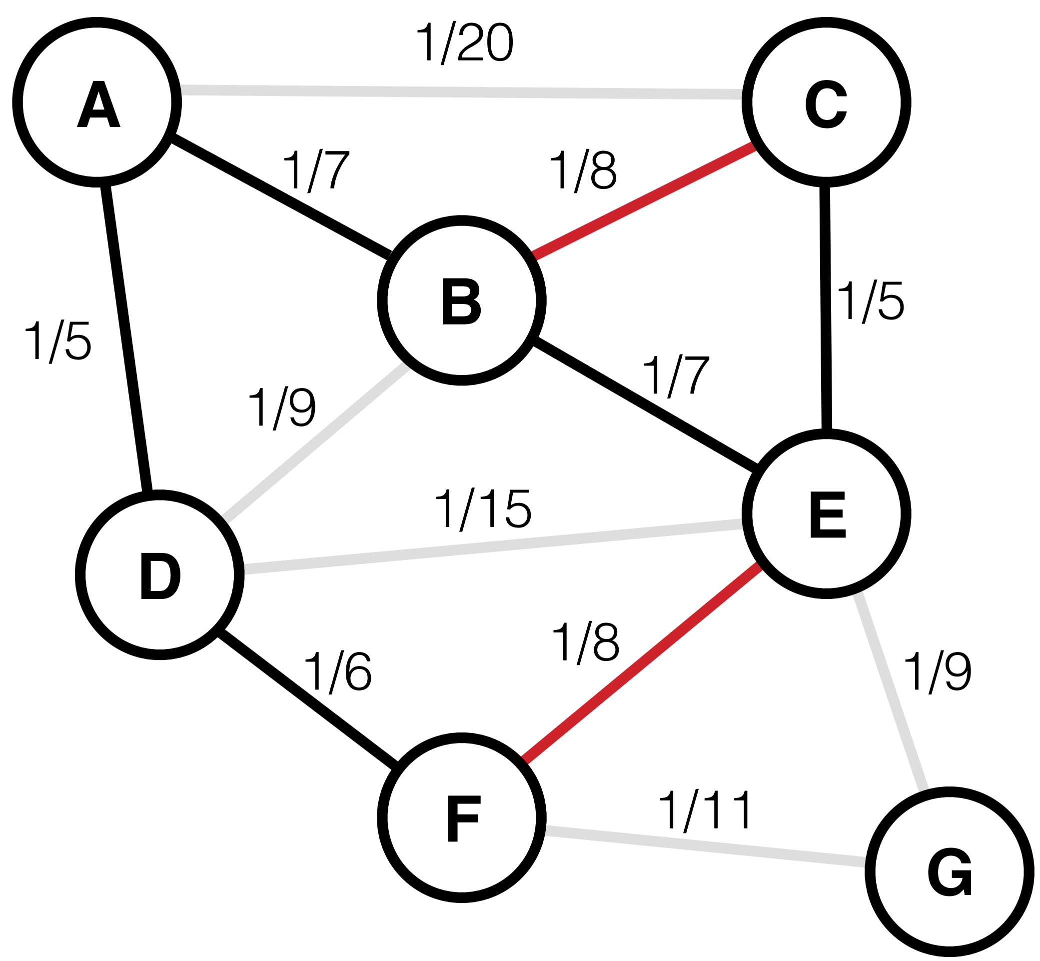
Chow-Liu Example



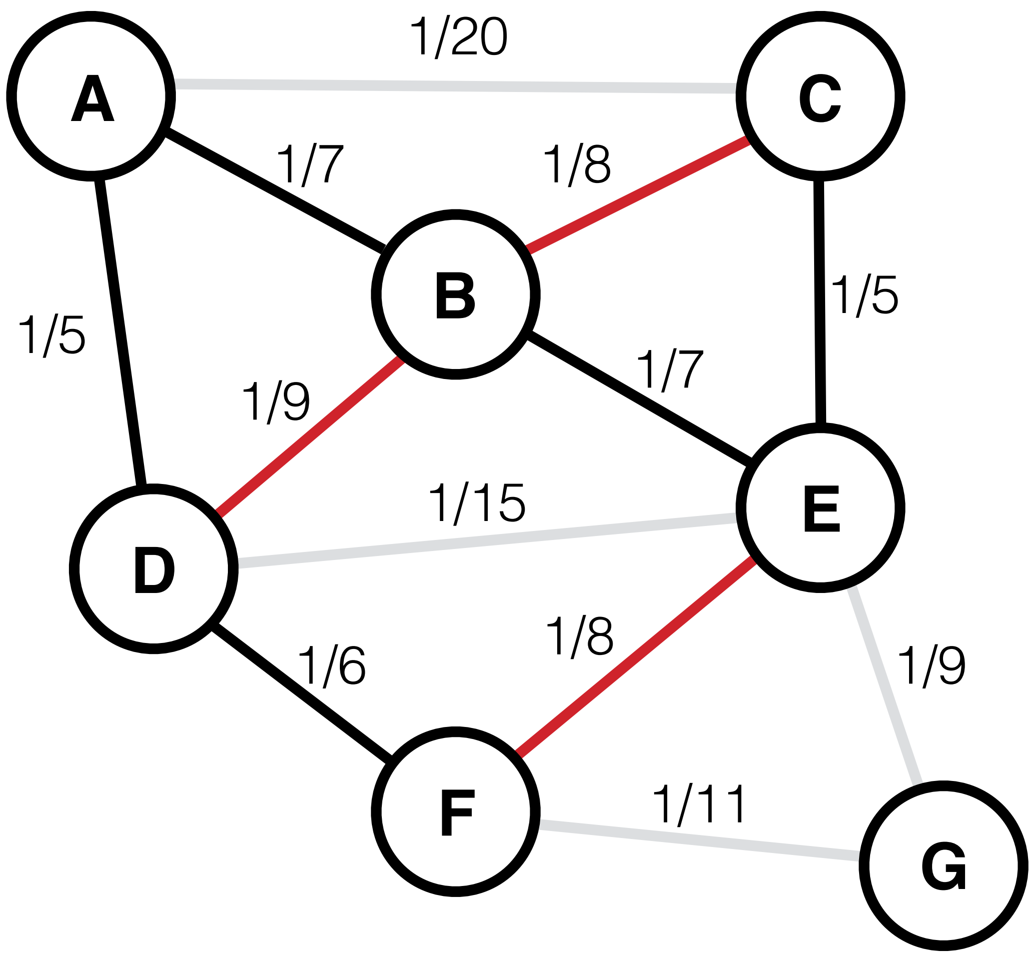
Chow-Liu Example



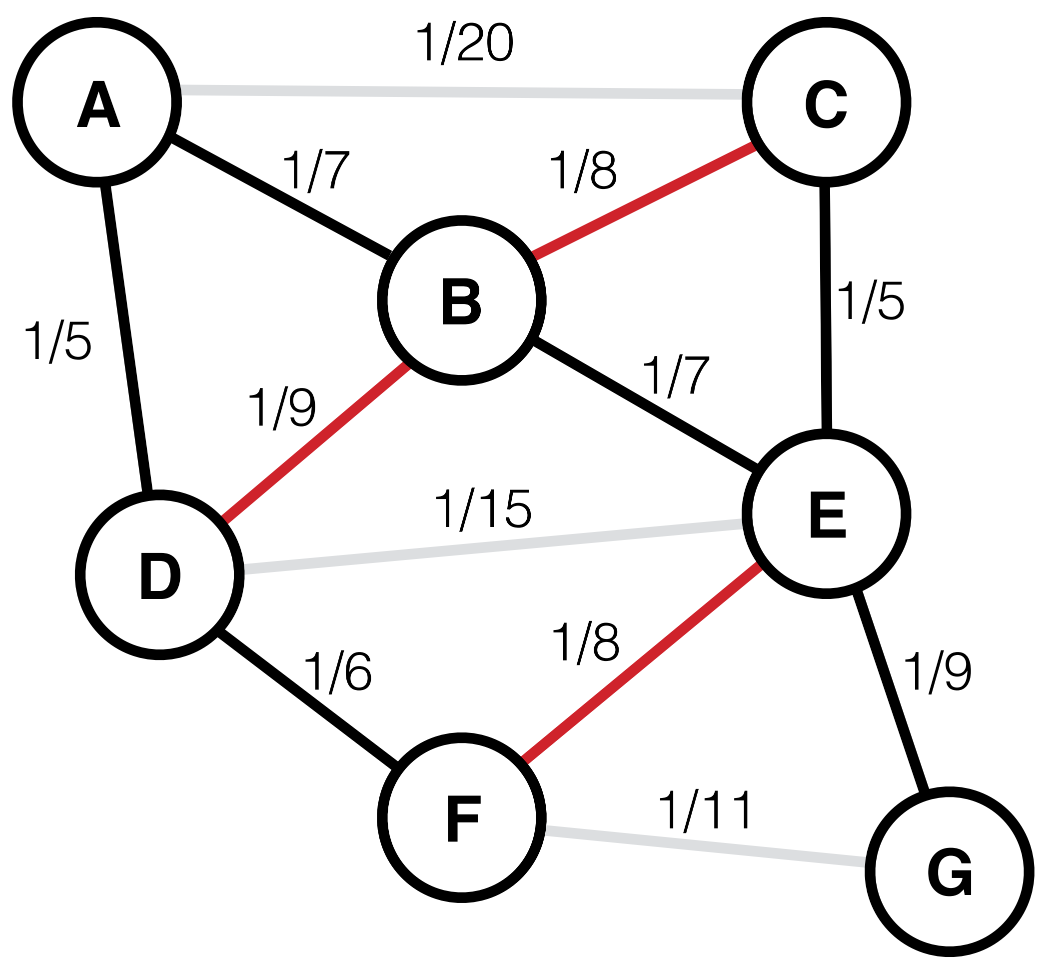
Chow-Liu Example



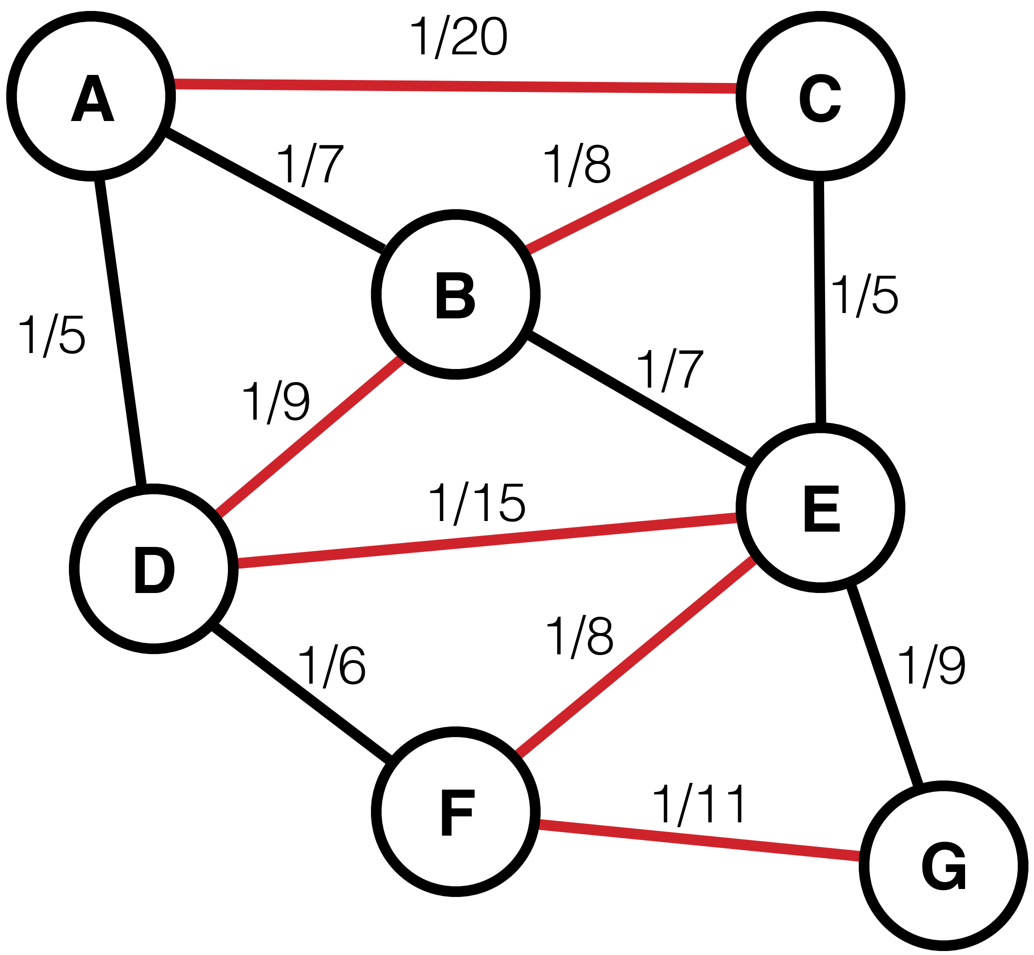
Chow-Liu Example



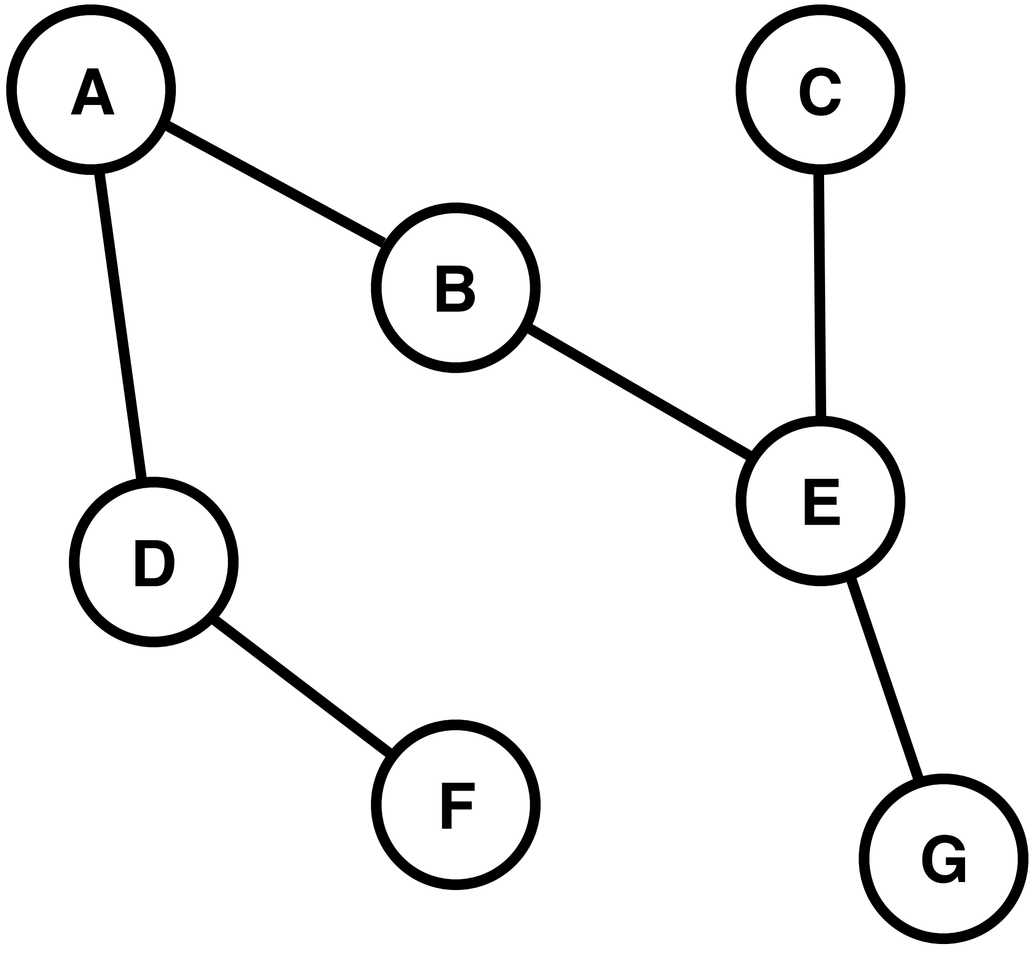
Chow-Liu Example



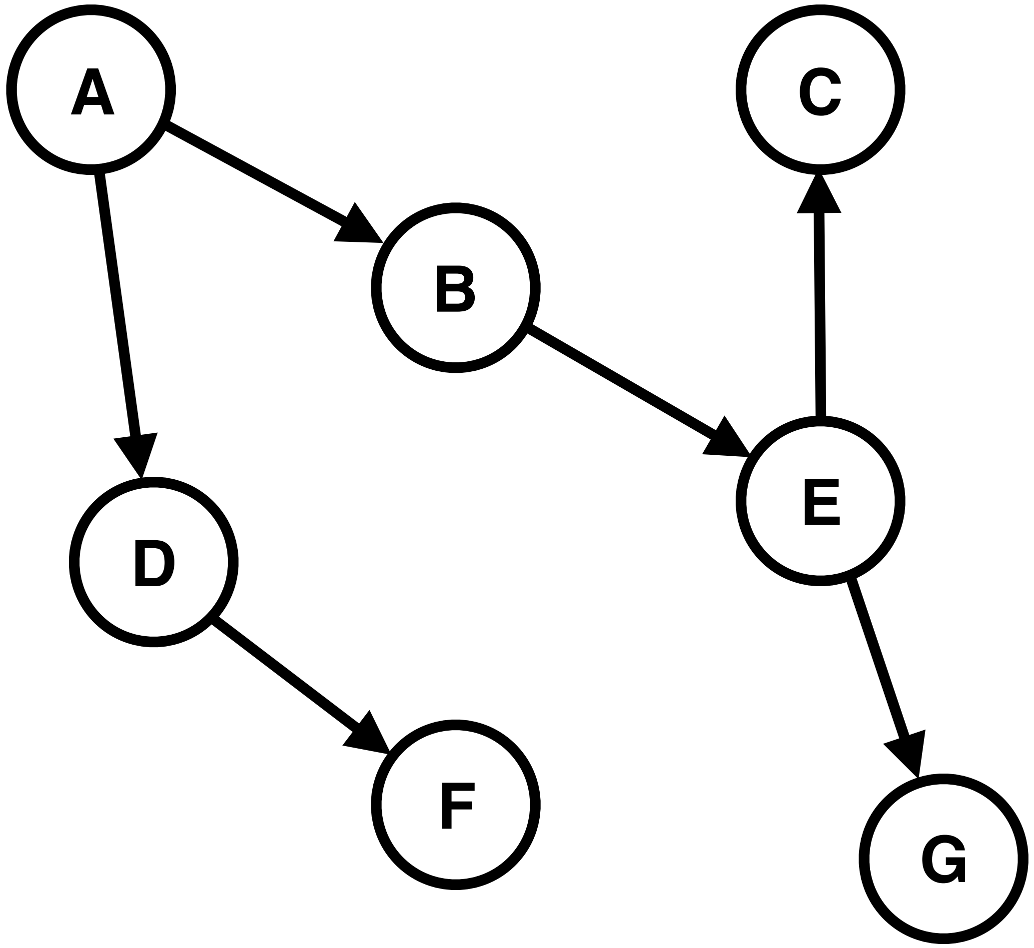
Chow-Liu Example



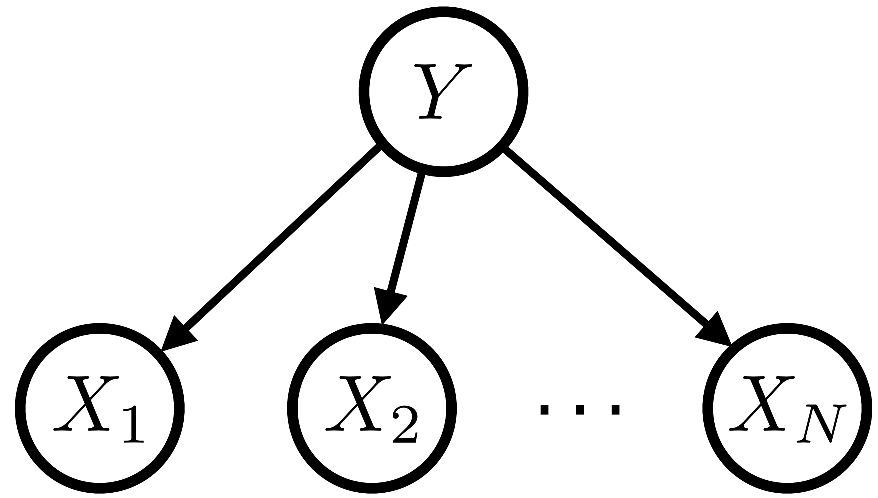
Chow-Liu Example



Chow-Liu Example

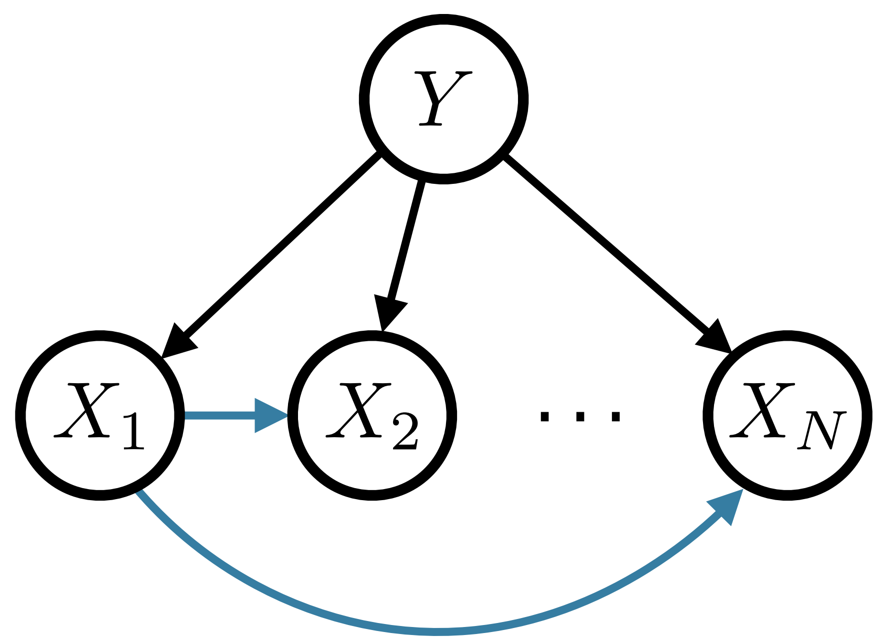


Naive Bayes



Tree Augmented Naive Bayes

Using Chow-Liu to learn the tree structure



Bayesian Networks Recap

Bayesian Networks

- Model **conditional independence assumptions**
- Model the **joint probability distribution** of variables
- Combine **prior knowledge** over:
  - Dependencies
  - Parameter values

Inference

- **NP-hard** in general
- Has closed-form solution for some graphs
- **Approximate methods** exist (e.g., Monte Carlo methods)

Learning

- Easy for fully observed data with known graph structure
- Using **EM** for partially observed data with known graph structure
- Structure learning is generally hard (possible with **Chow-Liu** for tree-structured networks)
- Structure learning very hard with partially observed data

Questions?