Computational analysis of biological structures and networks

Models for structured data: probabilistic graphical models

Manuele Bicego

Computer Science Dept.
University of Verona

Summary

- Introduction
- Probabilistic Graphical Models
- Bayesian Networks
  - Definition
  - Generative Process
  - Inference/learning
- Other Probabilistic Graphical Models

Introduction

Bayes decision rule.

Given an object x to classify, assign x to the class which posterior is maximum

$$class(x) = arg\,max_j P(\omega_j | x)$$

Or, in the same way (the evidence is a constant)

$$class(x) = arg\,max_j P(x | \omega_j) P(\omega_j)$$

Introduction

- The Bayes rule **guarantees** to achieve the minimum probability of error

but...

- Probabilities are not known, and should be estimated from the training set

- Two general ways of implenting this rule: Generative approaches vs Discriminative approaches

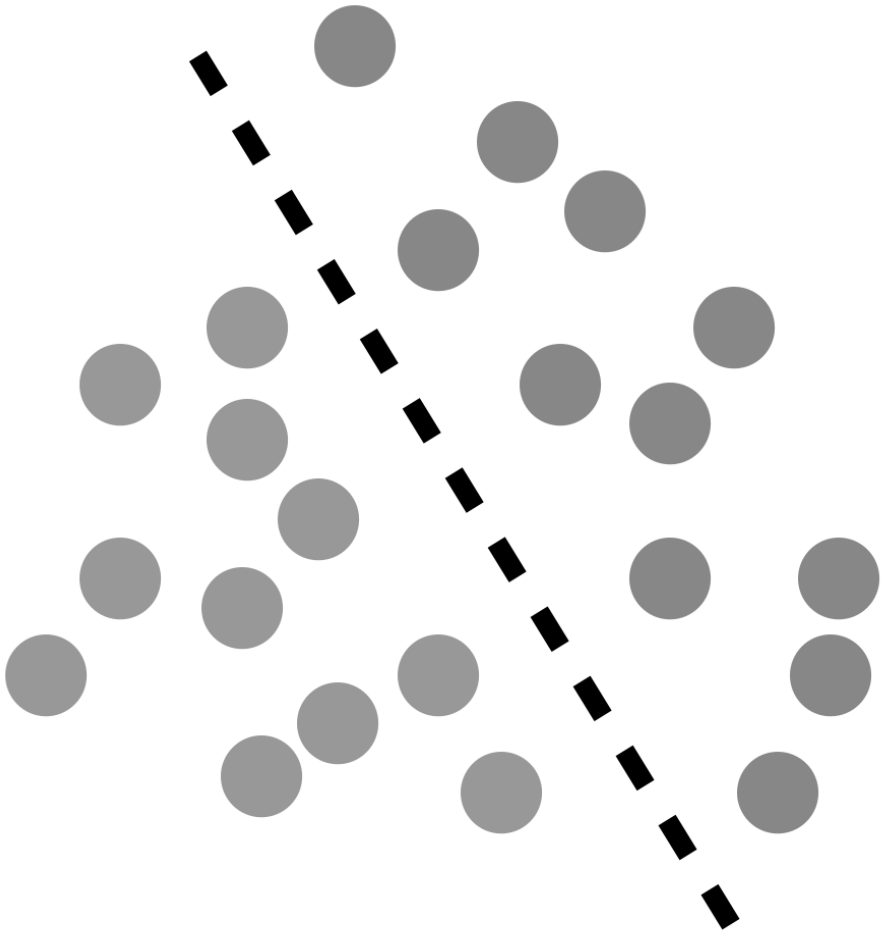
Introduction

- Generative approaches: estimation of the posterior probabilities of each class via the estimation of the **conditional** probabilities and the **prior** probabilities
  - More explicitly: estimation of a "**model**" **for every class**

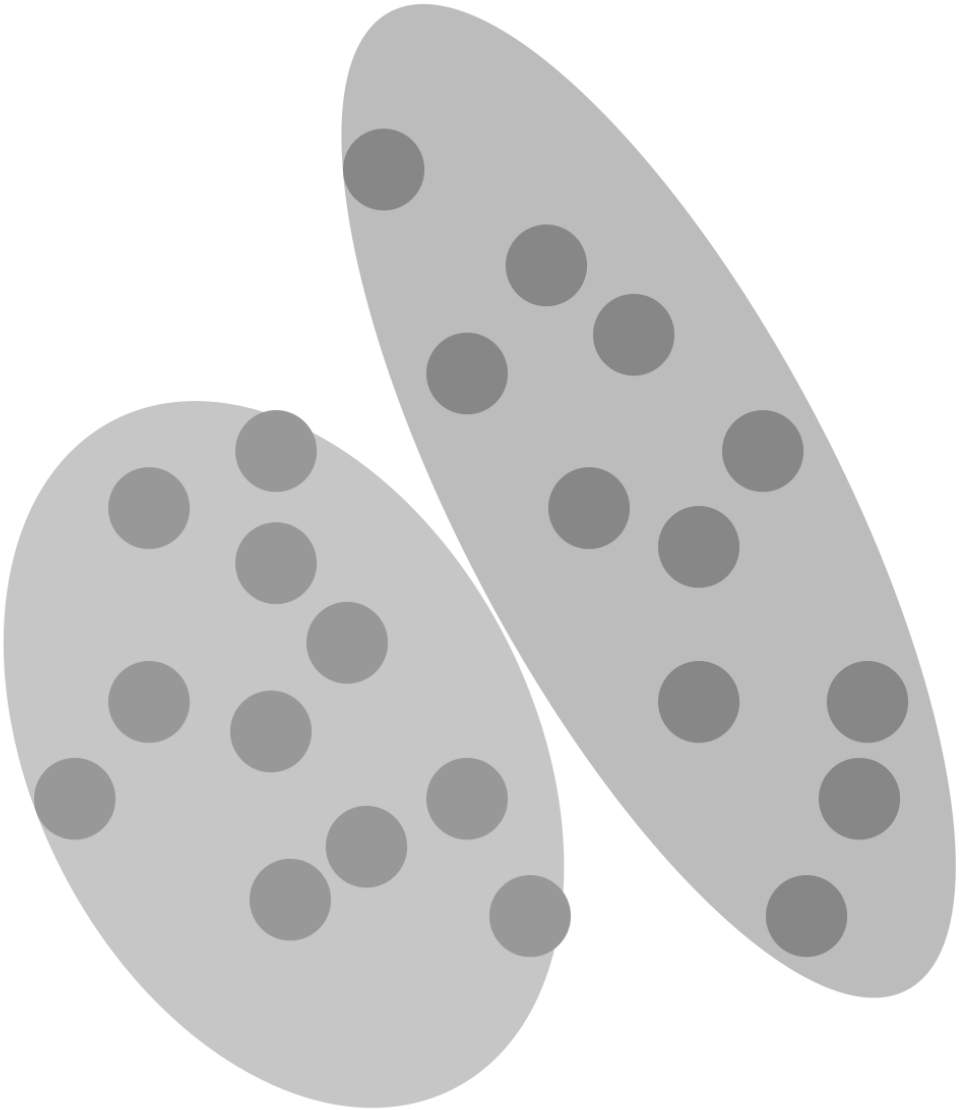
- Discriminative approaches: direct estimation of the posterior probabilities
  - More explicitly: direct estimation of the **boundary** of the classifier

From slides of chapter 1 (Basics)

Discriminative



Generative



Generative modeling

- We have a training set **T**, which can be used to infer the models (one for each class)
- We can divide the training set **T** in different subsets $\mathbf{T}_1, \mathbf{T}_2, \ldots, \mathbf{T}_c$ (one for every class)
- We use each $\mathbf{T}_i$ to estimate the model of class i
- In other words, for every class we want to solve the following problem

> Given a set of observations $x_1 \ldots x_N$, drawn from an unknown pdf p(x), the goal is to estimate p(x)

Note: p(x) represents the **generative model** we are looking for

Generative modelling

- IMPORTANT NOTE: a generative model can also model “non vectorial” objects
- In other words: the generative model formalizes a density p(**x**), and **x** is a generic object
- Example: the **Hidden Markov Model** represents a generative model
  - It describes a p(**x**) where **x** is a **sequence**!

Generative modeling

- One way to derive a generative model for modelling the p(x) is to use a **probabilistic graphical model**
  - A possible way of defining / modelling a pdf

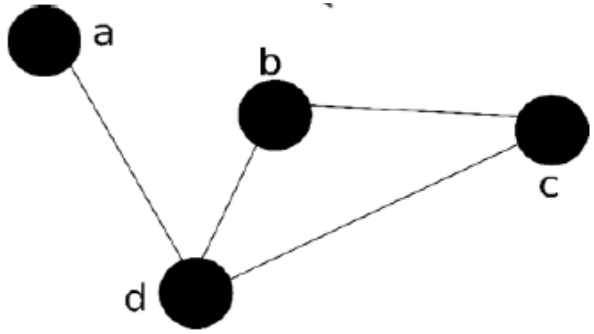
- Let's start the discussion from the concept of **graphical models**

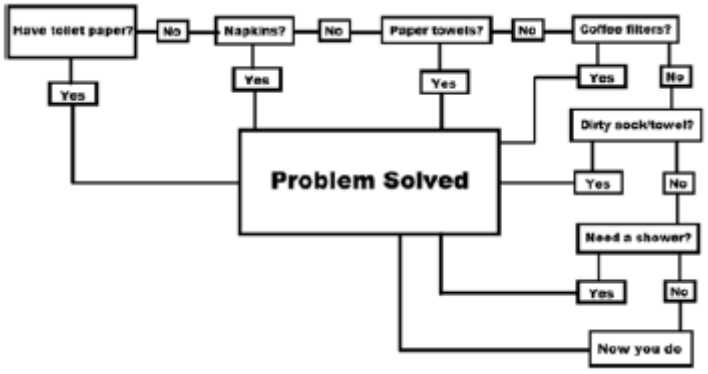
Graphical Models

Two definitions:

Graphical model**: models which can be described by a **graphical representation

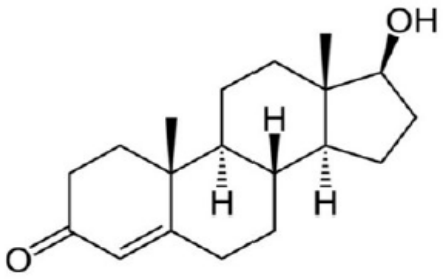
[Lauritzen, S.L.: Graphical models. Oxford University Press (1996)]





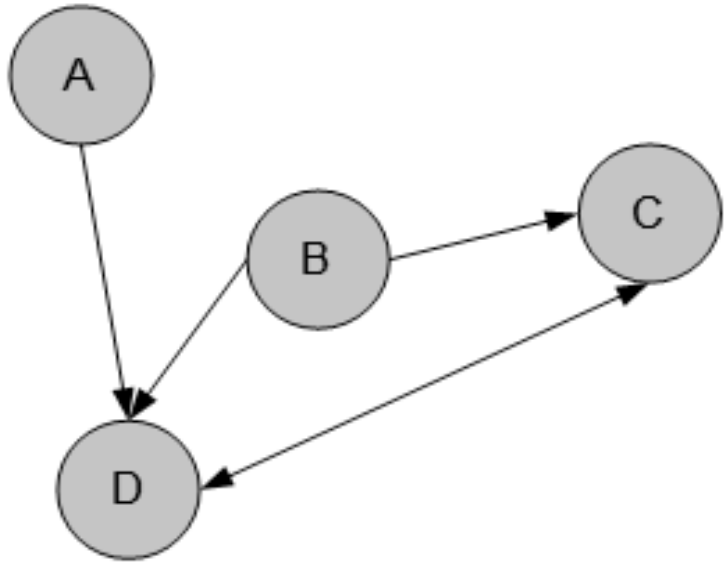
***Graphical models** are communication languages. They consist of a qualitative part, where features from graph theory are used, and a quantitative part consisting of potentials, which are real-valued functions over sets of nodes from the graph*

[Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer-Verlag (2001)]



Probabilistic Graphical Models

- **Probabilistic graphical models**: graphs where
  - each node represents a random variable
  - the links express probabilistic relationships between these variables.



A,B,C,D are random variables

There is a probabilistic relation between A and D

Probabilistic Graphical Models

Following the taxonomy introduced in Frey, B.J.: Graphical models for machine learning and digital communication. MIT press (1998):

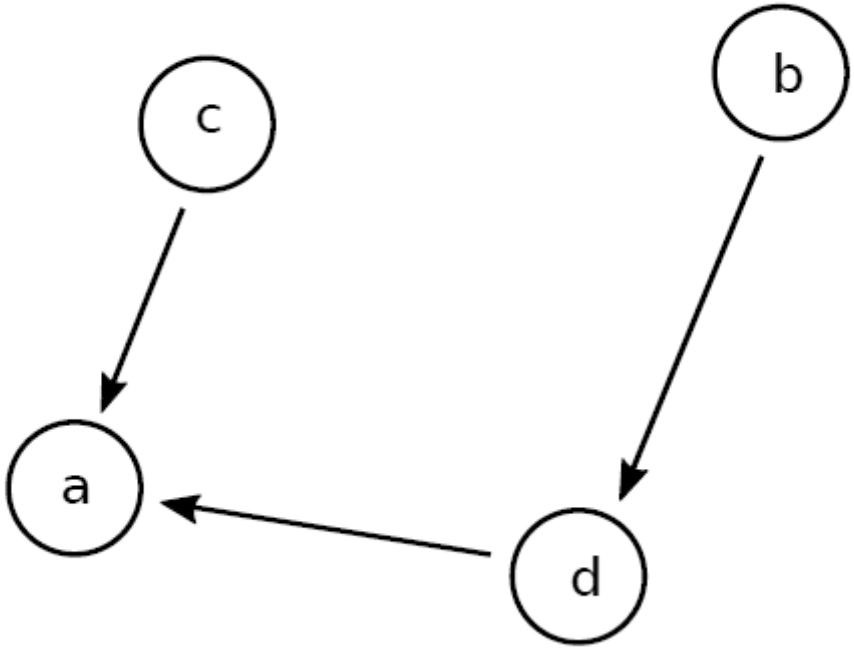
Probabilistic Graphical Models can be divided in three classes:

- Bayesian Networks
- Markov Random Field
- Factor Graphs

The differences stem from the typology of edges (directed / undirected) and on the kind of nodes

Bayesian Networks

- A Bayesian Network is a **directed acyclic** graph where nodes are **random variables** and edges describe relationship of **conditional probabilities**

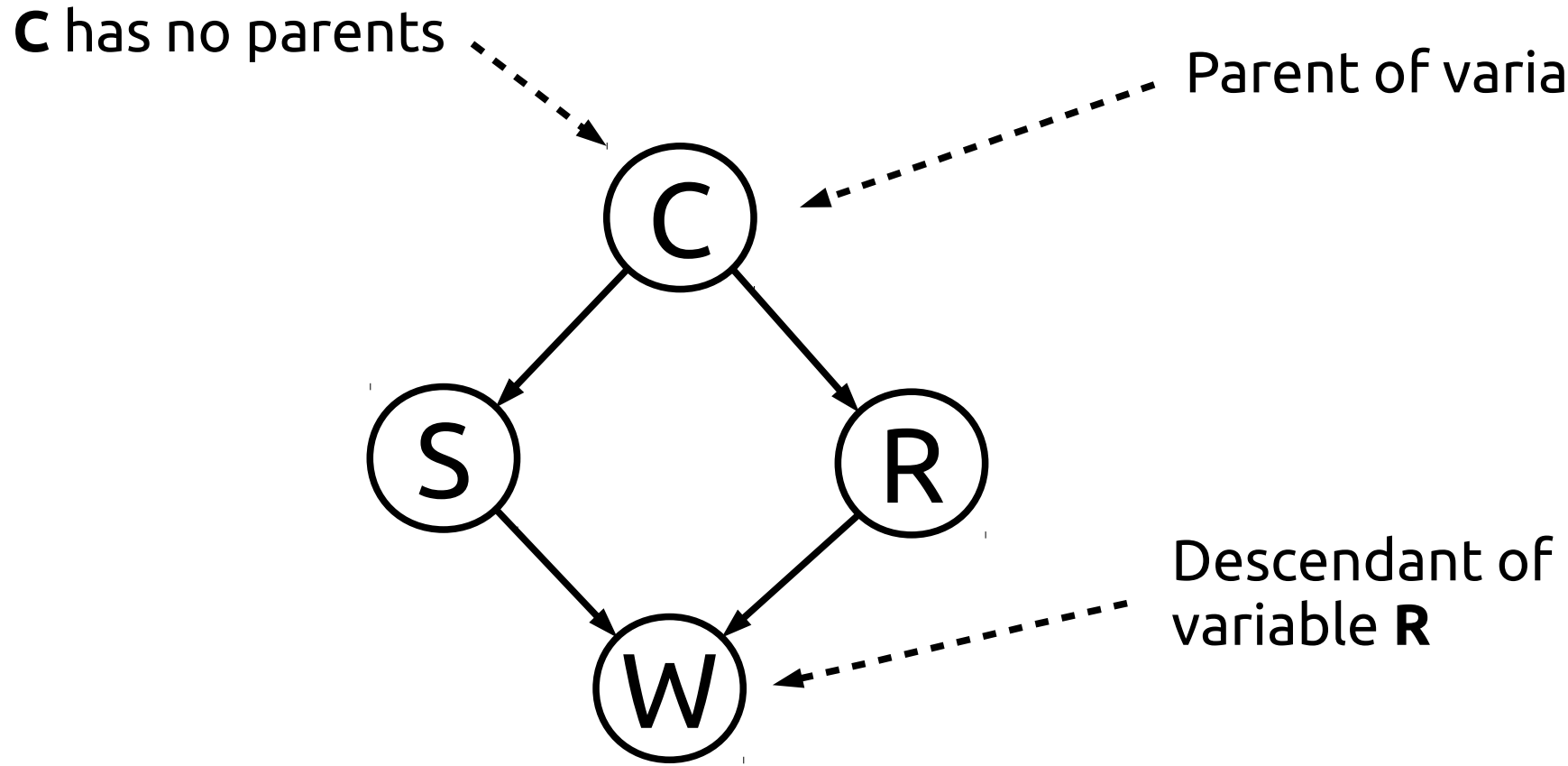


Bayesian Networks

Formally, a Bayesian Network is composed by

- A set of random variables $x_1, ... x_N$ (the nodes of the graph)
- A set of edges between the nodes
  - The resulting directed graph should be **acyclic**
- A set of conditional probability distributions, one for each variable x_i , indicating the probability of x_i given its parents $pa(x_i)$

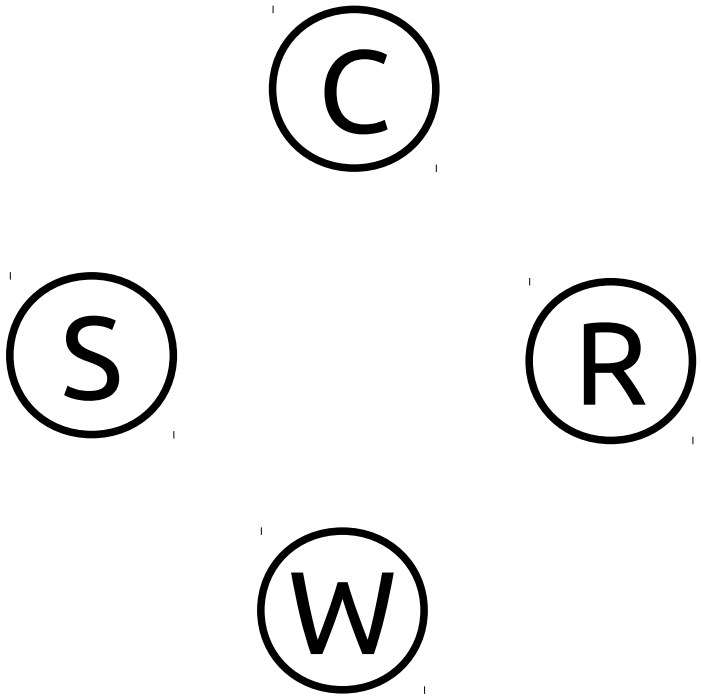
Bayesian Networks



Notation: `pa(x)` indicates the set of parents of the variable **`x`**

Example

Let's consider a Bayes Network for the problem of monitoring if the grass of the garden is wet or not



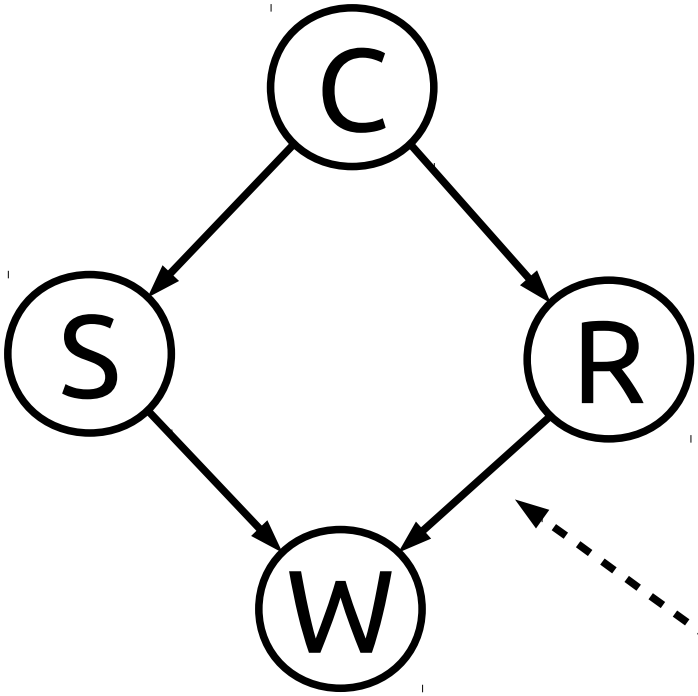
First ingredient: the set of random variables describing the entities involved the problem

Variables

C = Cloudy (True or False)
S = Sprinkler (True or False)
R = Rain (True or False)
W = Wet Grass (True or False)

Example: R = True → it is raining

Example



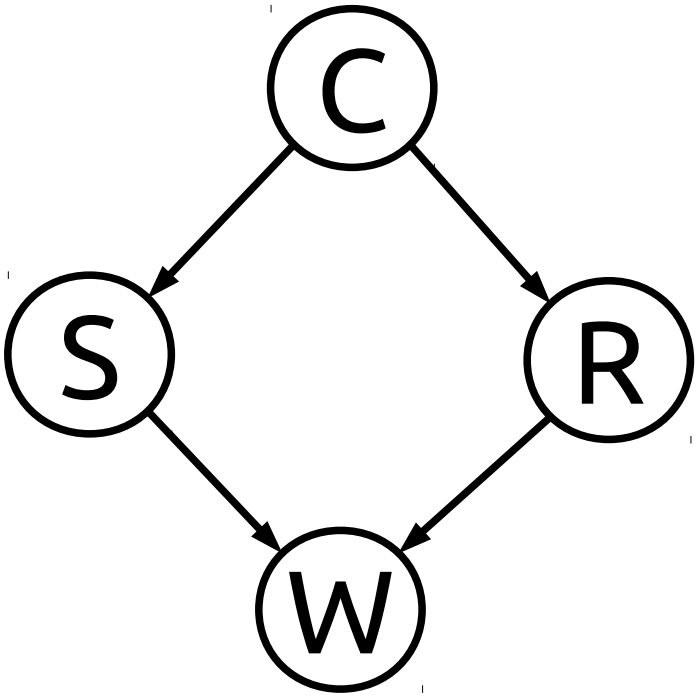
Second ingredient: the set of edges

Approximate meaning of the edges (K. Murphy):

"One can regard an arc from R to W as indicating that R ``causes'' W. This can be used as a guide to construct the graph structure."

This arc indicates that the variable R (it is raining or not) has an **influence** on the variable W (the grass is wet or not): **R "causes" W**

Example



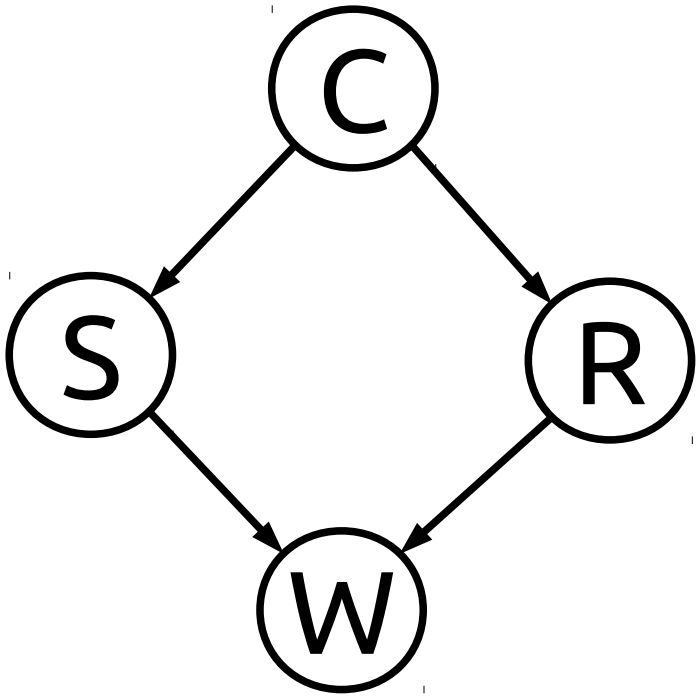
Third ingredient: the set of conditional probabilities: for each variable x we should define the probability `P(x|pa(x))`

(K. Murphy): Every conditional probability measures the strength of the relation between a given variable and its parents

Example

Example: define `p(R|pa(R))`
`pa(R) = {C}`, therefore we have to define `P(R|C)`

This probability describes the relation between R and C (or, better, in which way R is influenced by C)



P(R|C)

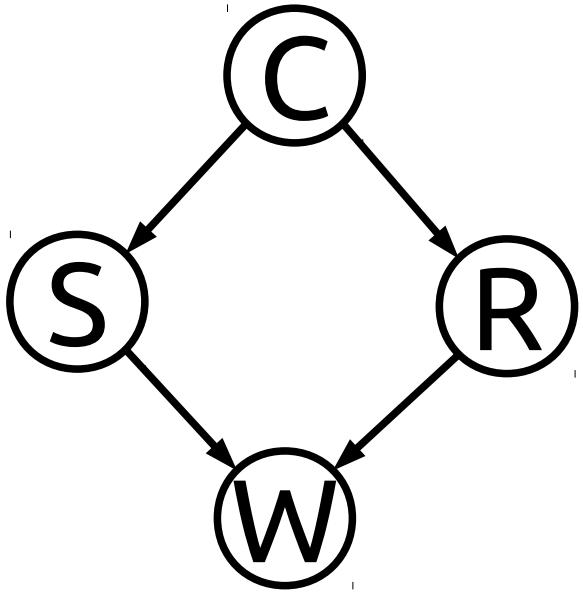
```
P(R=T | C=T) = 0.8
P(R=F | C=T) = 0.2
P(R=T | C=F) = 0.2
P(R=F | C=F) = 0.8
```

The event "it is raining" (R=T) is more probable if the sky is cloudy (C=T)

Example

The complete Bayesian Network

P(C=T) = 0.5
P(C=F) = 0.5



P(S=T | C=T) = 0.1
P(S=F | C=T) = 0.9
P(S=T | C=F) = 0.5
P(S=F | C=F) = 0.5

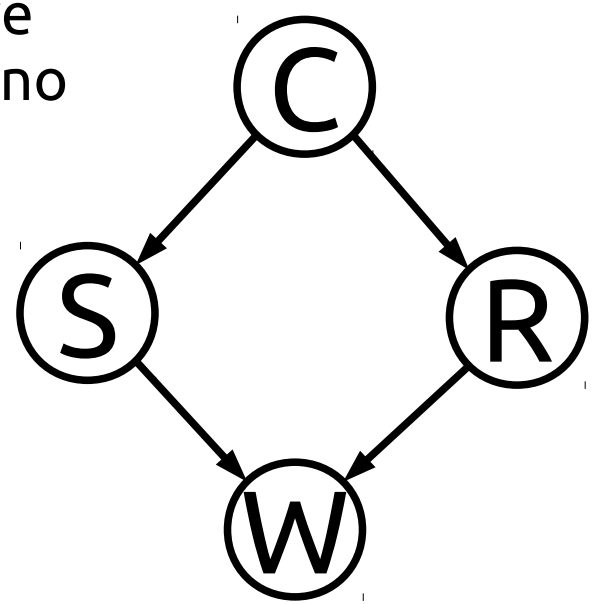
P(R=T | C=T) = 0.8
P(R=F | C=T) = 0.2
P(R=T | C=F) = 0.2
P(R=F | C=F) = 0.8

P(W=T | S=T,R=T) = 0.99	P(W=T | S=T,R=F) = 0.9
P(W=F | S=T,R=T) = 0.01	P(W=F | S=T,R=F) = 0.1
P(W=T | S=F,R=T) = 0.9	P(W=T | S=F,R=F) = 0.0
P(W=F | S=F,R=T) = 0.1	P(W=F | S=F,R=F) = 1.0

BN: a lot of information

The event "the sprinkler is opened " (S=T) is more probable if there are no clouds (C=F)

```
P(C=T) = 0.5
P(C=F) = 0.5
```



The event "it is raining" (R=T) is more probable if the sky is cloudy (C=T)

```
P(S=T | C=T) = 0.1
P(S=F | C=T) = 0.9
P(S=T | C=F) = 0.5
P(S=F | C=F) = 0.5
```

```
P(R=T | C=T) = 0.8
P(R=F | C=T) = 0.2
P(R=T | C=F) = 0.2
P(R=F | C=F) = 0.8
```

```
P(W=T | S=T,R=T) = 0.99
P(W=F | S=T,R=T) = 0.01
P(W=T | S=F,R=T) = 0.9
P(W=F | S=F,R=T) = 0.1
```

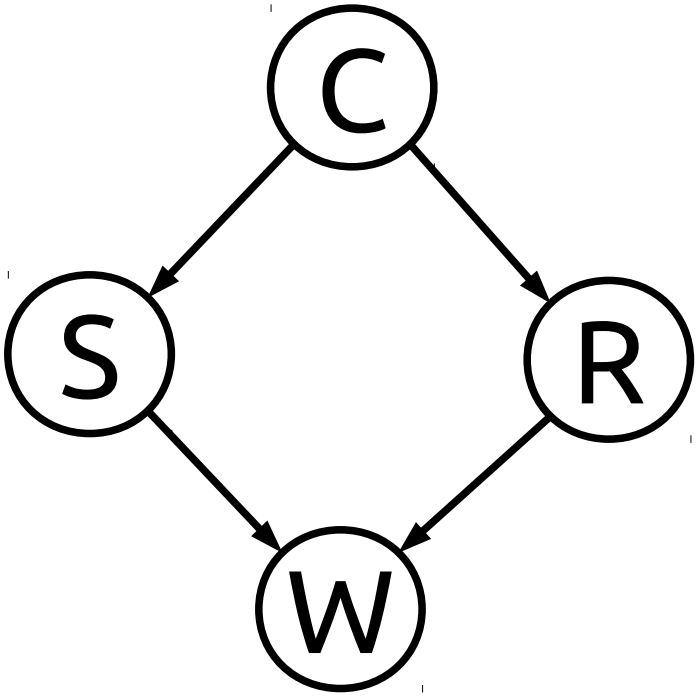
```
P(W=T | S=T,R=F) = 0.9
P(W=F | S=T,R=F) = 0.1
P(W=T | S=F,R=F) = 0.0
P(W=F | S=F,R=F) = 1.0
```

The event "grass is wet" (W=T) has two possible causes: either the water sprinker is on (S=T) or it is raining (R=T)

Bayesian Networks

Main property of BN: the joint probability (i.e. the probability of all variables) can be factorized as

$$P(x_1, x_2, ..., x_N) = \prod_{i=1}^{N} P(x_i | pa(x_i))$$



P(**C**,**S**,**R**,**W**) =
P(**W**|**S**,**R**) P(**S**|**C**) P(**C**) P(**R**|**C**)

C has no parents!

S and **R** are the parents of **W**

Bayesian Networks

- **Important note:** many **conditional independence** properties among variables can be read directly from the graph without any analytical manipulation!

$$p(x|y) = p(y|x) = p(x)$$

Marginal independence

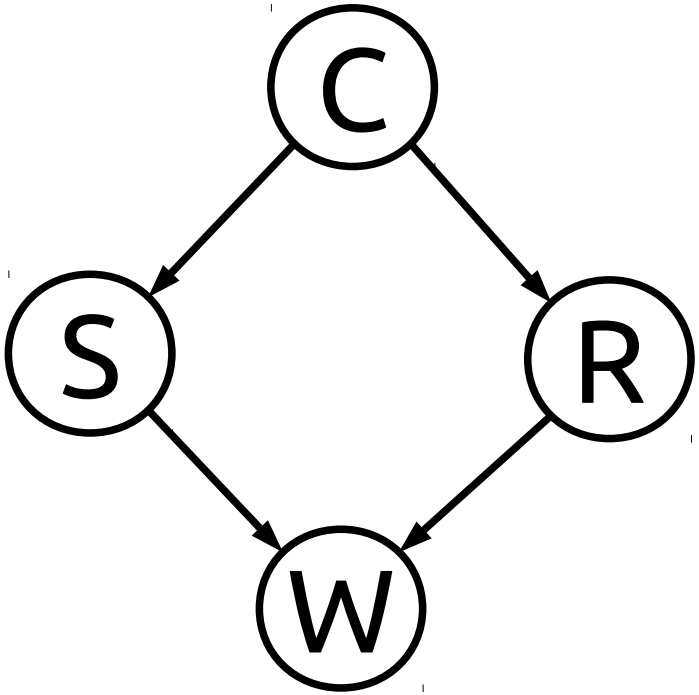
$$p(x|y, z) = p(x|z)$$

Conditional independence

- Examples of properties: D-separation, Markov Blanket, Directed Markov Property, ...

Bayesian Networks

<u>Directed Markov Property:</u> Each variable is **conditionally independent** of all its **non-descendants** in the graph given the value of all its **parents**



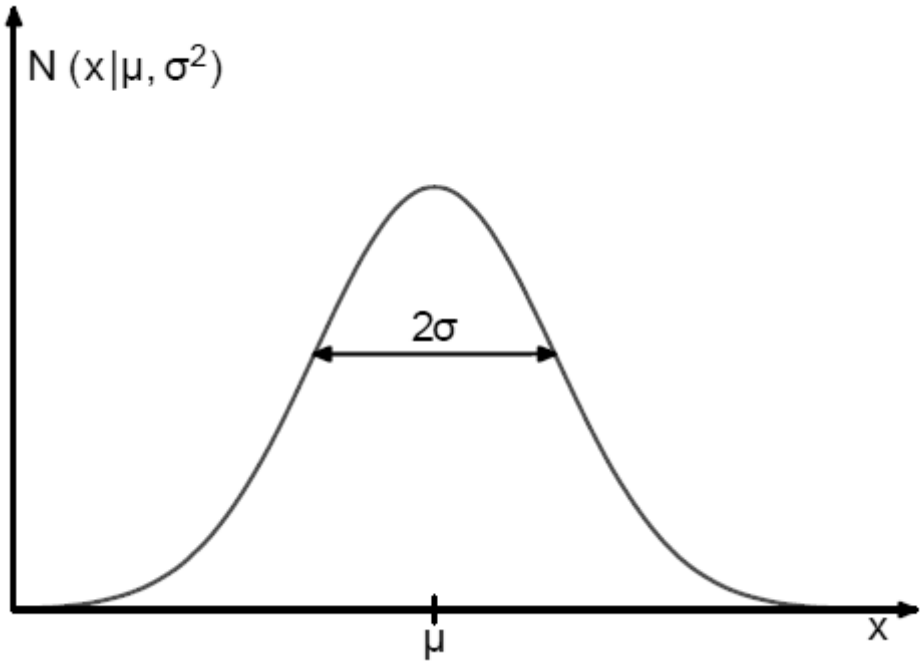
$$P(\mathbf{W}|\mathbf{C},\mathbf{S},\mathbf{R}) = P(\mathbf{W}|\mathbf{S},\mathbf{R})$$

Further notes (1)

- Note 1. We have to specify the **form** of each conditional probability p(**x**|pa(**x**))
  - If **x** is discrete, then we can use **discrete pdf** (i.e. tables)
  - If x is continuous, then we should use **continuous pdf**
  - In all cases a reasonable choice is to employ **parametrized** function (i.e. functions which are completely defined by a **set** of **parameters**)

Further notes (1)

- Example. The Gaussian distribution



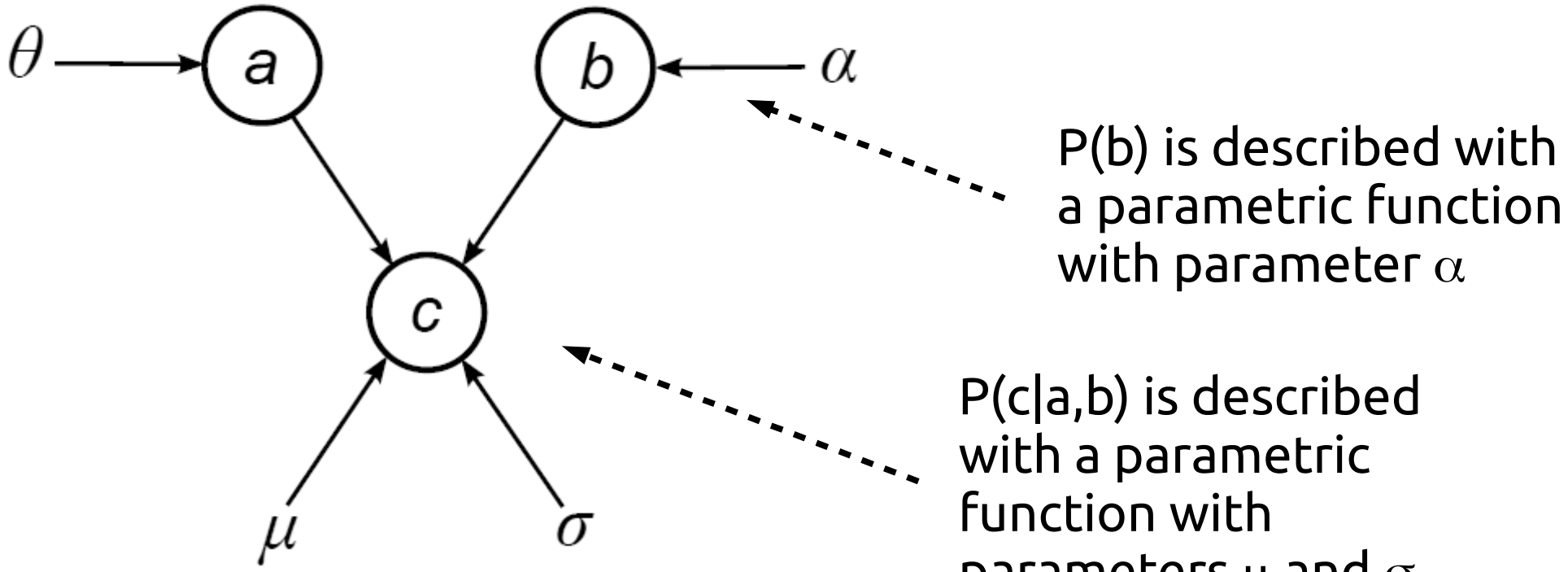
Parameters of the Gaussian: μ and σ

Once the parameters are known, the function is completely defined, i.e. we can compute the value of the function for all possible x

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

Further notes (1)

- The information about parameters can be inserted in the model through shapeless nodes containing the parameter name



Further notes (2)

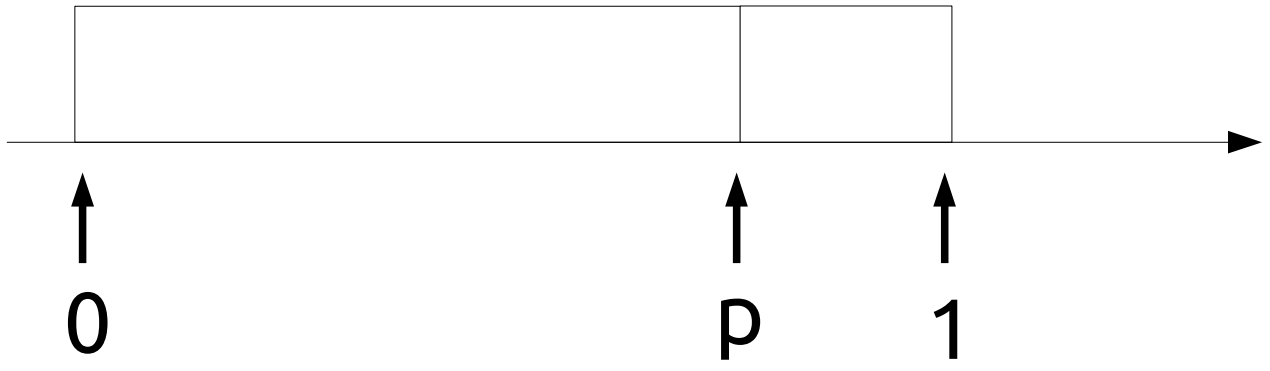
- Note 2. Bayesian Networks are **generative models**
  - We can use these models to describe a set of points; but we can also use these models to generate **new** points via the so-called **generative process** (which explains how to generate new points from a given pdf)
  - Generative process: “sampling” from the pdf described by the model

Further notes (2)

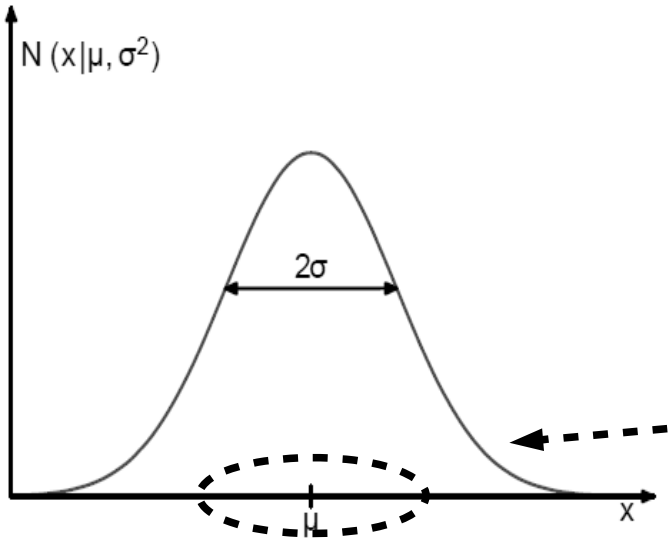
- Example: “sampling” from a **Bernoulli** distribution
- **Bernoulli**: the **binary** variable takes value 1 with probability **p** and 0 with probability **1-p** (“p” represents the **parameter** of the Bernoulli distribution)
- Sampling: extract a random number from 0 to 1 (uniform sampling, all points have the same probability)
- If the random number is less than p, then assign to x the value 1, zero otherwise

Further notes (2)

If the random point falls in the **red** part, then the variable assumes the value 1, otherwise it assumes the value 0



Since p is large (approaching 1), if we repeat the process most of the points will be 1



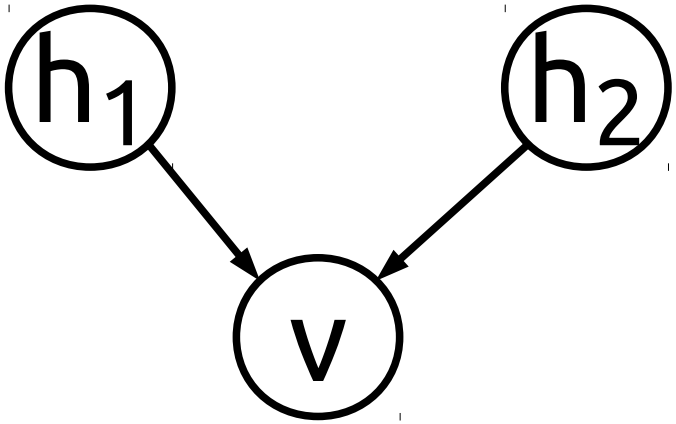
Sampling from a Gaussian: most of the points will be in this zone

Further notes (3)

- Note 3: Typically, Bayesian Networks are exploited in very complex scenarios, where the relations are not all explicit
- In such situations it is useful to introduce **hidden/latent** variables: they are variables which **can not be observed**
  - We can not “sample” their value from the problem
- These variables are meant to represent latent causes that influence the visible variables (the available data)

Further notes (3)

- Graphically, to distinguish between visible and hidden variables we adopt **empty circles** to represent visible variables and **shadowed circles** to represent hidden variables



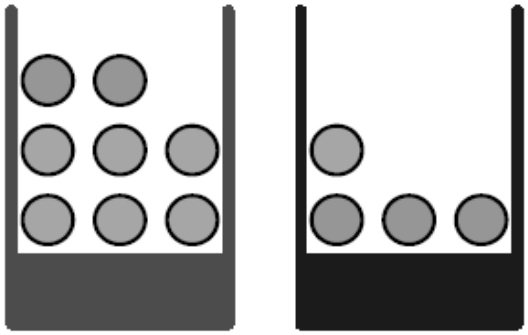
The observable variable **v** depends on two hidden variables $\mathbf{h_1}$ and $\mathbf{h_2}$

If we observe the problem (e.g. to get the training set) we can measure only **v**

Another example: the “Two-boxes” problem

We should design a BN to model the following problem:

- There are two boxes, one red and one blue
- The boxes are covered with a blanket, so that we cannot observe the color
- In the red box there are 2 apples and 6 oranges, in the blue box there are 3 apples and 1 orange
- We want to model the procedure of extracting fruits from the boxes (after extraction, a fruit is re-integrated in the box)



Bishop, C.: Pattern Recognition and Machine Learning. Springer-Verlag (2006)

Exercise

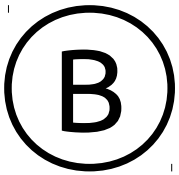
- EXERCISE: Try to design a Bayesian Network modeling the “Two-boxes” problem

  Suggestion: you should identify variables (visible and hidden), edges and conditional probabilities

The “Two-boxes” problem

Variables of the BN.

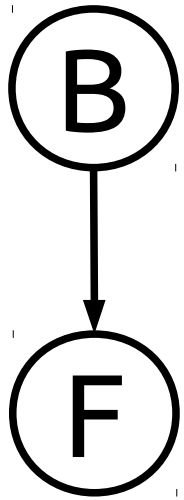
- **B**: it represents the box. It can take the value 'r' (red box) or 'b' (the blu box). This represents a **hidden variable** (we cannot see the box from which the fruit comes)
- **F:** it represents the fruit. It can take the value 'o' (orange) or 'a' (apple). This represents a **visible variable** (we can see the fruit once extracted)

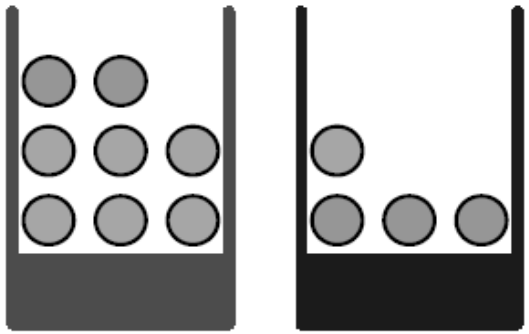


The “Two-boxes” problem

Edges of the BN.

- From the problem we know that the extracted fruit depends on the box from which we extract it
- A link from **B** to **F**: the choice of the box influences the extracted fruit





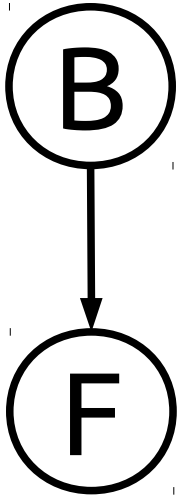
The “Two-boxes” problem

Conditional probabilities: we have to define p(B|pa(B)) and p(F|pa(F))

- p(B|pa(B)) = p(B). It can be modelled via a Bernoulli distribution of parameter α

P(B)

P(B = 'b') = α
P(B = 'r') = 1-α

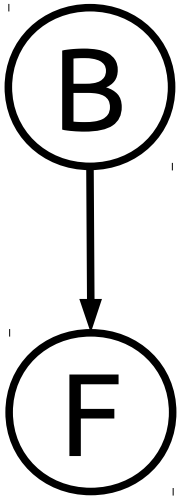


The “Two-boxes” problem

- p(F|pa(F)) = p(F|B). This can be modelled via two Bernoulli distributions:
  - One if the fruit is taken from the red box (i.e. if B='r') – let's call this Bernoulli's parameter as β
  - One if the fruit is taken from the blu box (i.e. if B='b') -- let's call this Bernoulli's parameter as γ

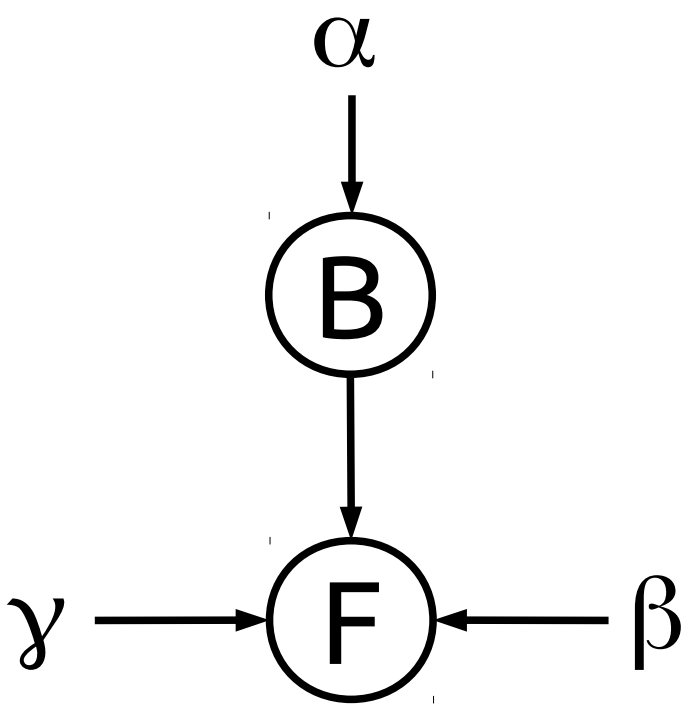
P(F|B)

P(F = 'o' | 'B = 'r') = β
P(F = 'a' | 'B = 'r') = 1-β
P(F = 'o' | 'B = 'b') = γ
P(F = 'a' | 'B = 'b') = 1-γ



The “Two-boxes” problem

The full Bayesian Network

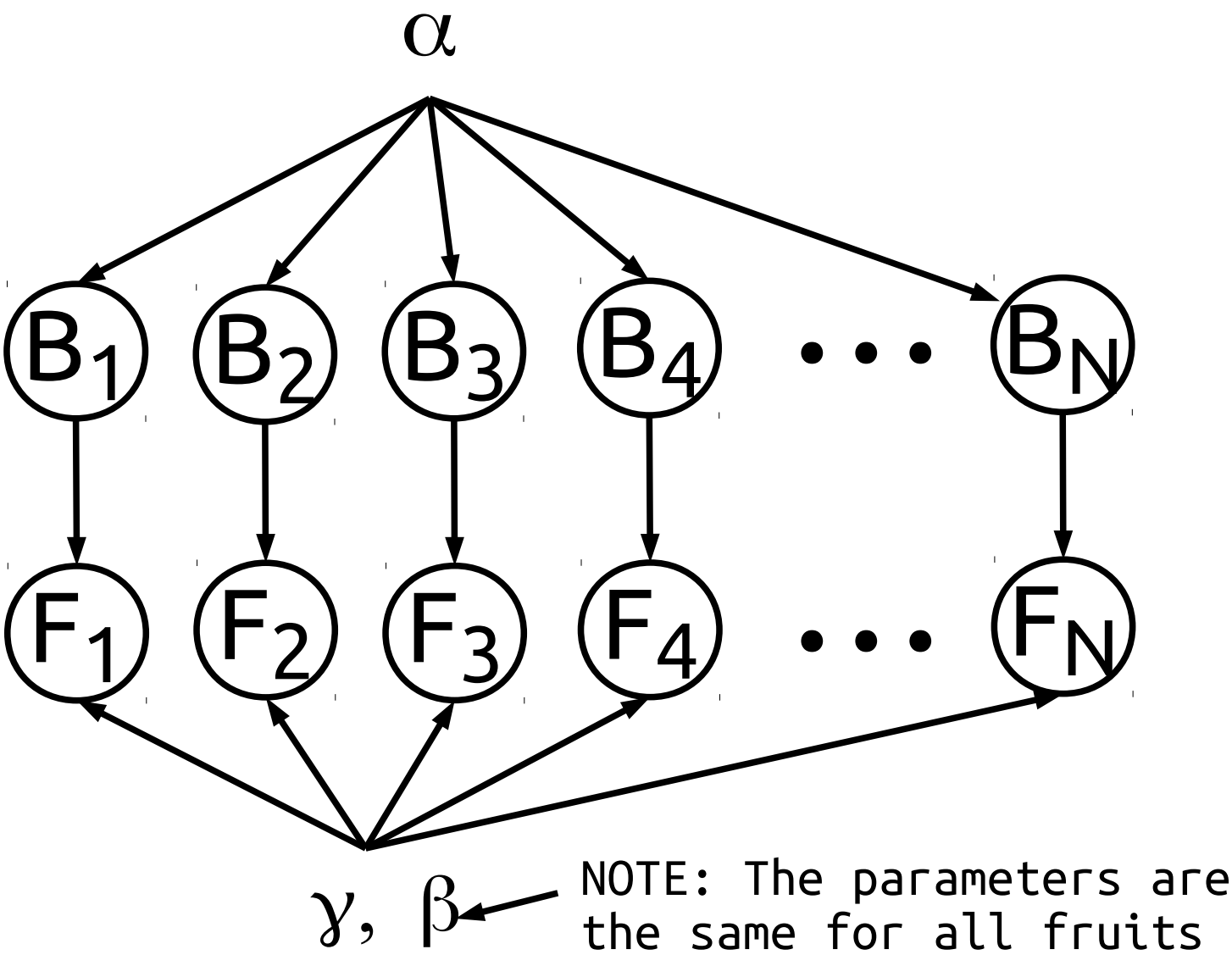


The “Two-boxes” problem

- This represents a Bayesian Networks for **one** single extracted fruit.

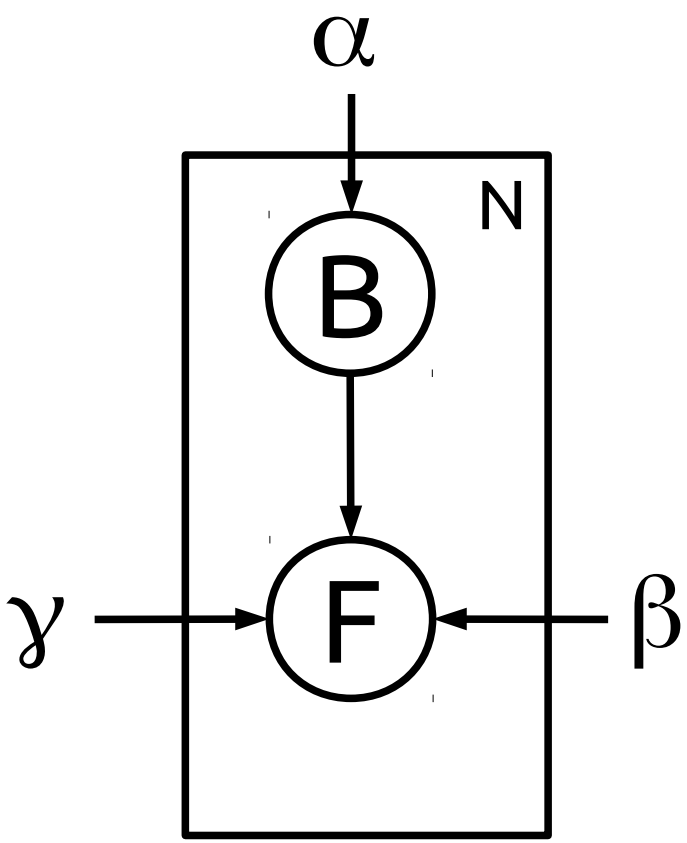
- If we want to extract a set of fruits f_1, … f_N, then we should create a BN which contain a pair of variables (F,B) for every fruit of the dataset

The “Two-boxes” problem



The “Two-boxes” problem

This can be written in a more compact way with the so called **plate notation**: a single set of representative nodes is surrounded with a plate labeled with N, indicating that there are N nodes – which are independent and identically drawn (i.i.d) – of this kind.

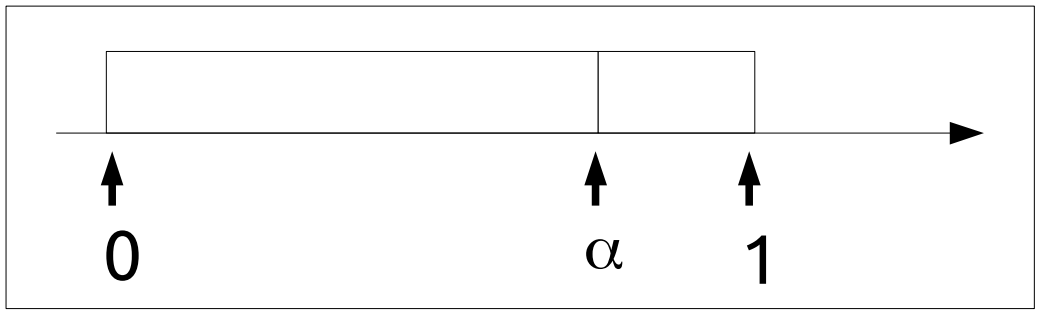


The “Two-boxes” problem: generative process

- **Generative process**: how to generate new observations from this Bayesian Networks
- First step: sample a box B from the Bernoulli parametrized with α

P(B)

```
P(B = 'b') = α
P(B = 'r') = 1-α
```



The “Two-boxes” problem: generative process

- Second step: sample a fruit F from the Bernoulli corresponding to the box B (i.e. with parameter β if B = 'r', with parameter γ if B = 'b')

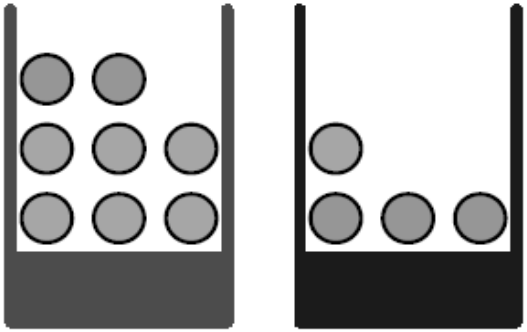
P(F|B)

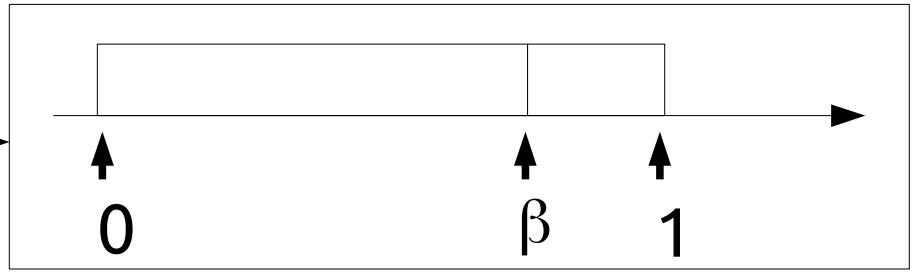
P(F = 'o' | 'B = 'r') = β

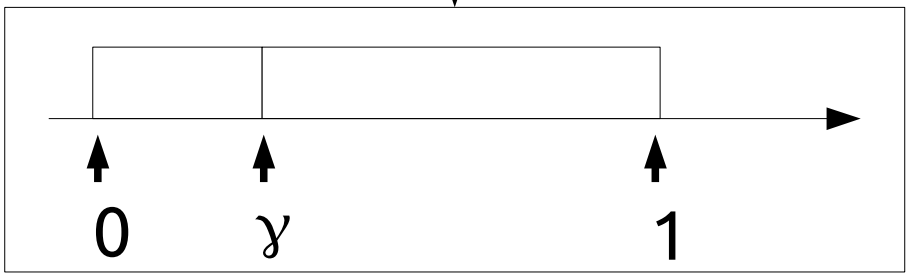
P(F = 'a' | 'B = 'r') = 1-β

P(F = 'o' | 'B = 'b') = γ

P(F = 'a' | 'B = 'b') = 1-γ







The probability of getting an orange from the blu box is lower

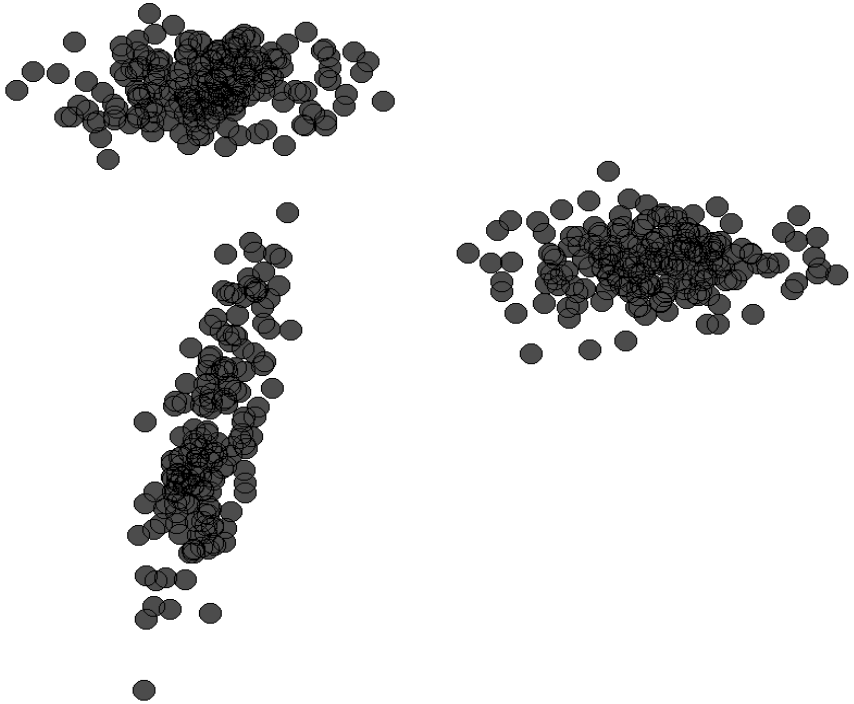
Known probabilistic models seen as Bayesian Networks

- Different well known probabilistic models can be seen from the perspective of Bayesian Networks
- In particular here:
  - Gaussian Mixture Models
  - Hidden Markov Models
- The goal is to identify hidden variables, observable variables, edges, parameters
- Let's also sketch the generative process

Gaussian Mixture Models

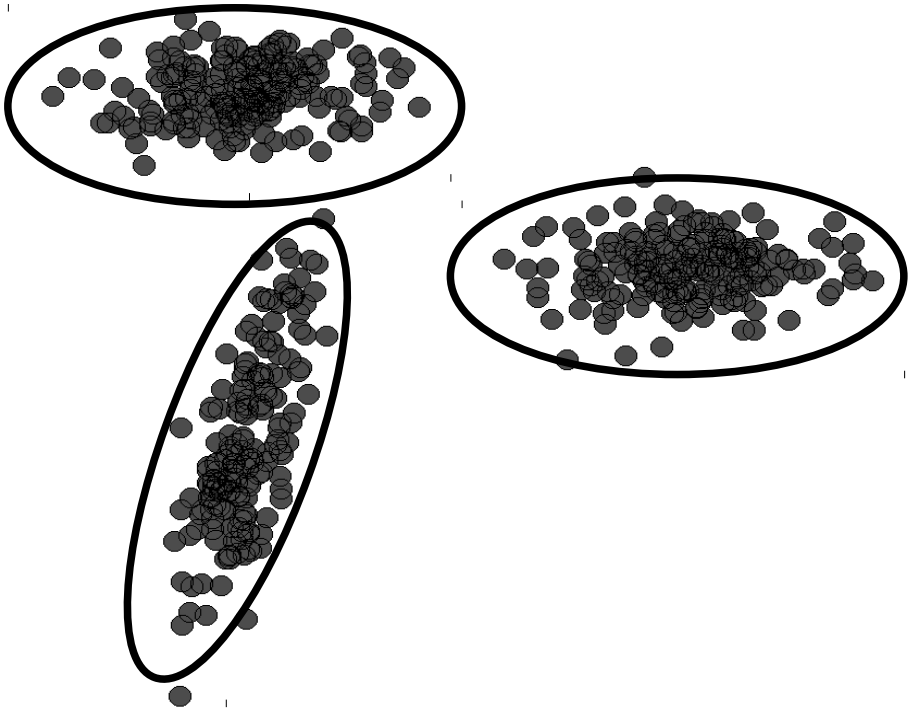
- Widely known probabilistic model for clustering
  - It belongs to the “model-based” class of clustering approaches
- Main idea: describe a set of points using different Gaussian distributions
- In particular GMM assumes that the points **come** from a **Mixture** of **K Gaussians**

Gaussian Mixture Models



Original data

Gaussian Mixture Models



GMM

Gaussian Mixture Models

- In particular, within a GMM we assume that the distribution of the points follows the following formula

$$p(x) = \sum_{j=1}^{K} \pi_j f_j(x|\Theta_j)$$

Where every component f_j is a Gaussian

$$f_j(x_i|\theta_j) = f_j(x_i|\mu_j, \Sigma_j) = \frac{1}{2\pi^{d/2}|\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T \Sigma_j^{-1}(x_i-\mu_j)}$$

Gaussian Mixture Models

- Example: GMM in 1D with 2 components

$$p(x) = \pi_1 \mathcal{N}(x|\mu_1, \sigma_1) + \pi_2 \mathcal{N}(x|\mu_2, \sigma_2)$$

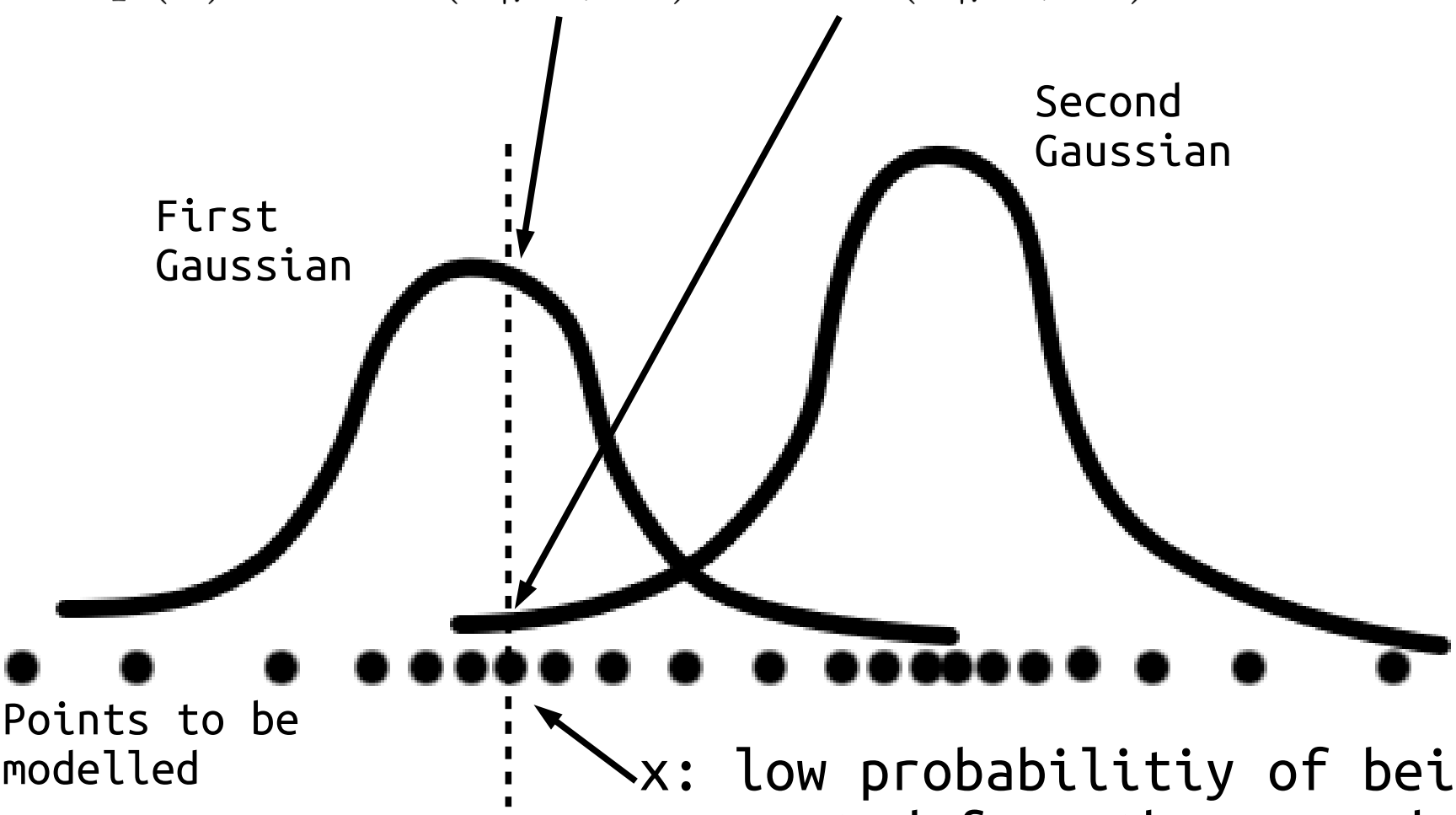
$\pi_1 =$ Prior probability for the first Gaussian

$\pi_2 =$ Prior probability for the second Gaussian

$$N(x|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{||x-\mu||^2}{2\sigma^2}}$$

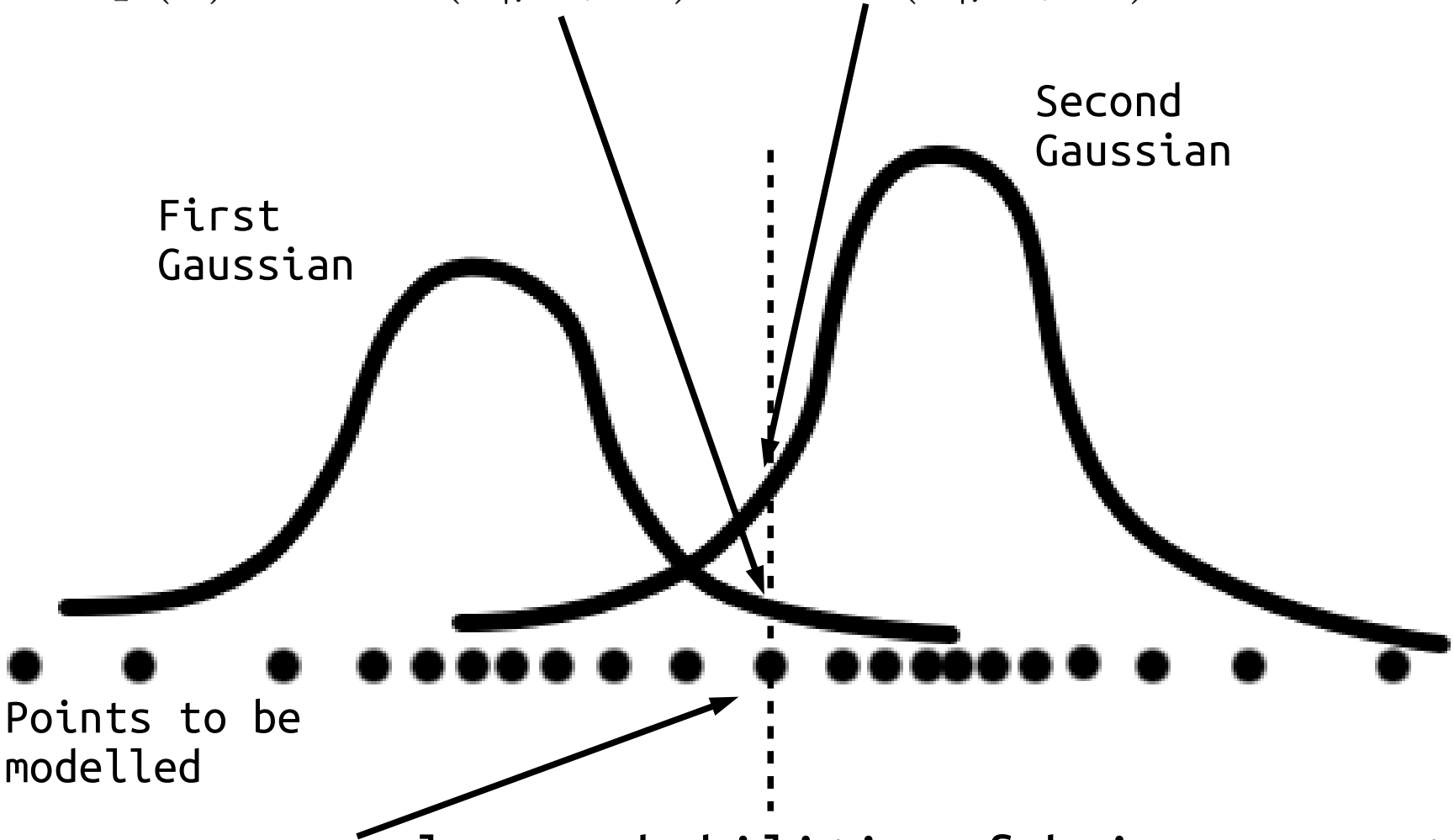
Gaussian Mixture Models

$$p(x) = \pi_1 \mathcal{N}(x|\mu_1, \sigma_1) + \pi_2 \mathcal{N}(x|\mu_2, \sigma_2)$$



Gaussian Mixture Models

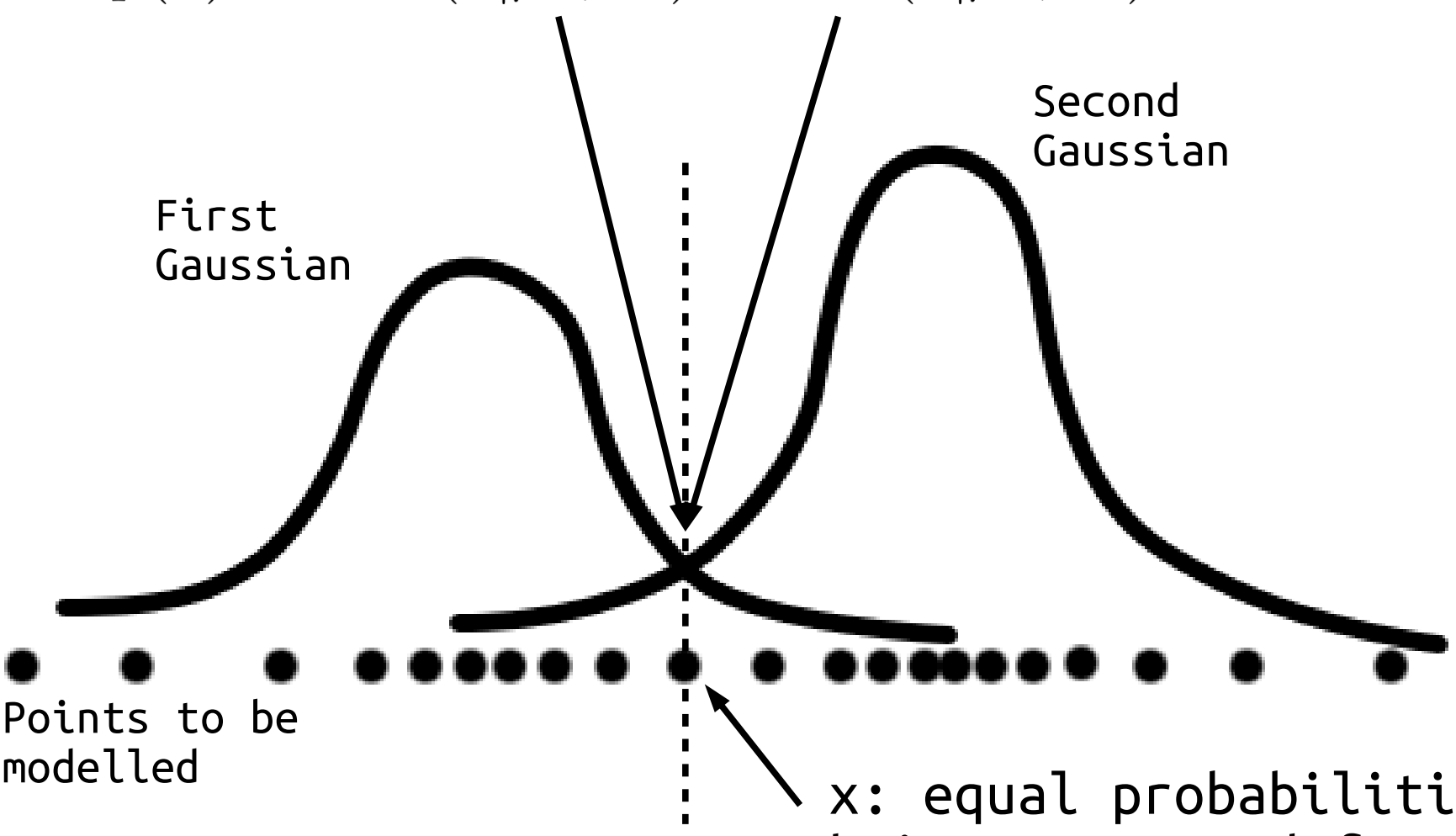
$$p(x) = \pi_1 \mathcal{N}(x|\mu_1, \sigma_1) + \pi_2 \mathcal{N}(x|\mu_2, \sigma_2)$$



x: low probabilitiy of being generated from the first Gaussian, high probability of being generated from the second

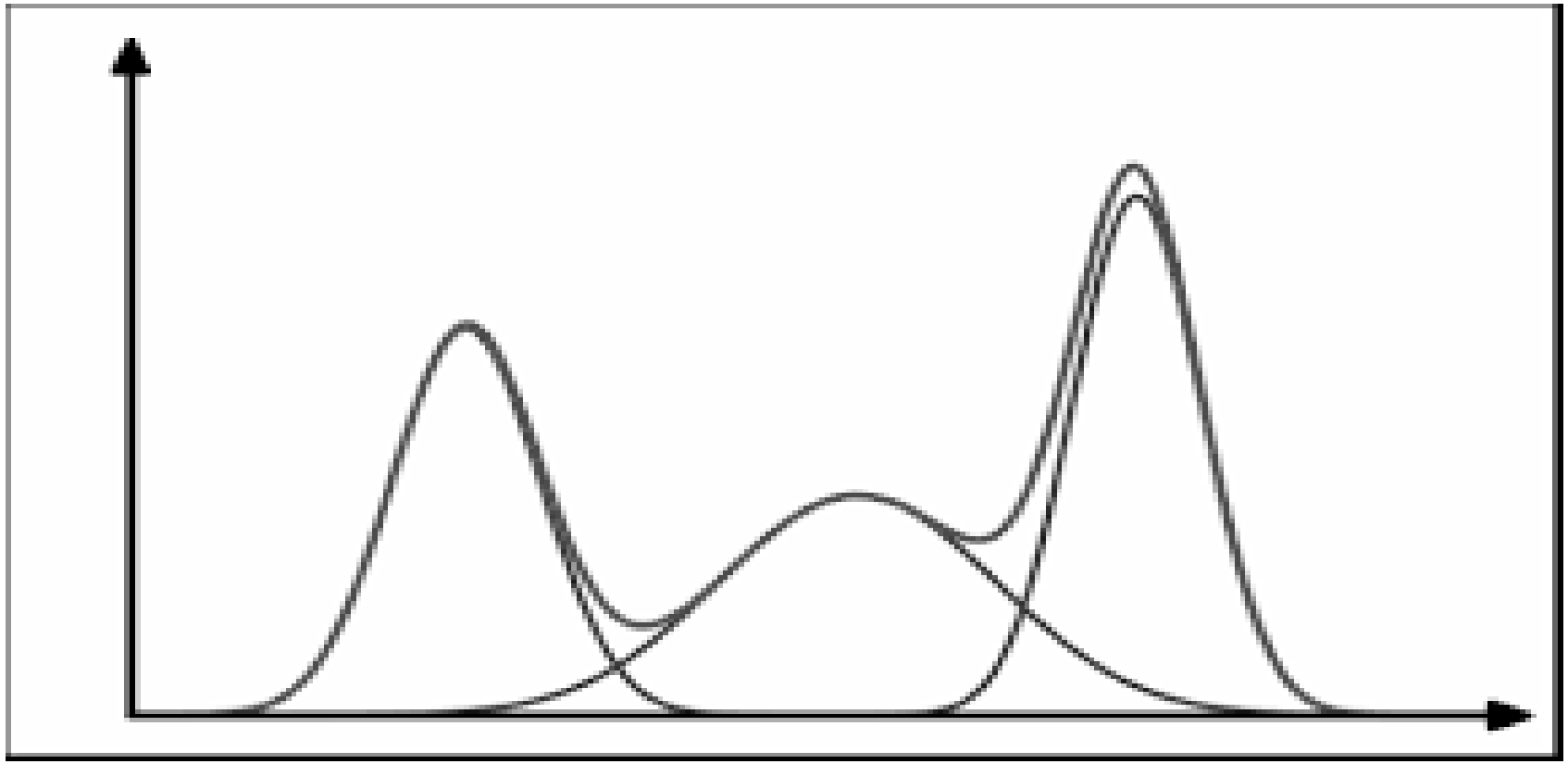
Gaussian Mixture Models

$$p(x) = \pi_1 \mathcal{N}(x|\mu_1, \sigma_1) + \pi_2 \mathcal{N}(x|\mu_2, \sigma_2)$$



Gaussian Mixture Models

- NOTE: The probability in one point is **the sum** of the probabilities of all the Gaussians (weighted by priors)
  - This permits to estimate different shapes of pdf



Gaussian Mixture Models

Generative process: how to sample from a GMM?

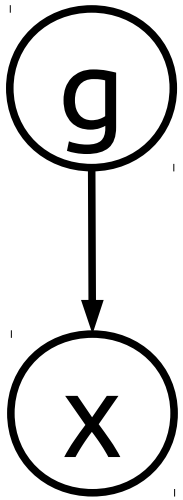
GMM assumes that every point derives from one of the two Gaussians, even if we don't know which one

- Step 1. Select one of the two Gaussians using the prior probability π
- Step 2. Sample the point from the Gaussian chosen (i.e. from the Gaussian with parameter μ_1,σ_1 if in the first step we select the first Gaussian, or μ_2,σ_2 otherwise)

GMM as Bayesian Networks

Variables and edges:

- **x** = variable usable to describe the **value** of the point (**visible** variable, continuous)
- **g** = variable usable to describe which is the **component**, i.e. from which Gaussian we have sampled the point (hidden variable, discrete)
- Clearly the value of **x** depends on the chosen Gaussian (i.e. on the value of **g**)



GMM as Bayesian Networks

Conditional probabilities:

- p(g): the prior probabilities (g has no parents)

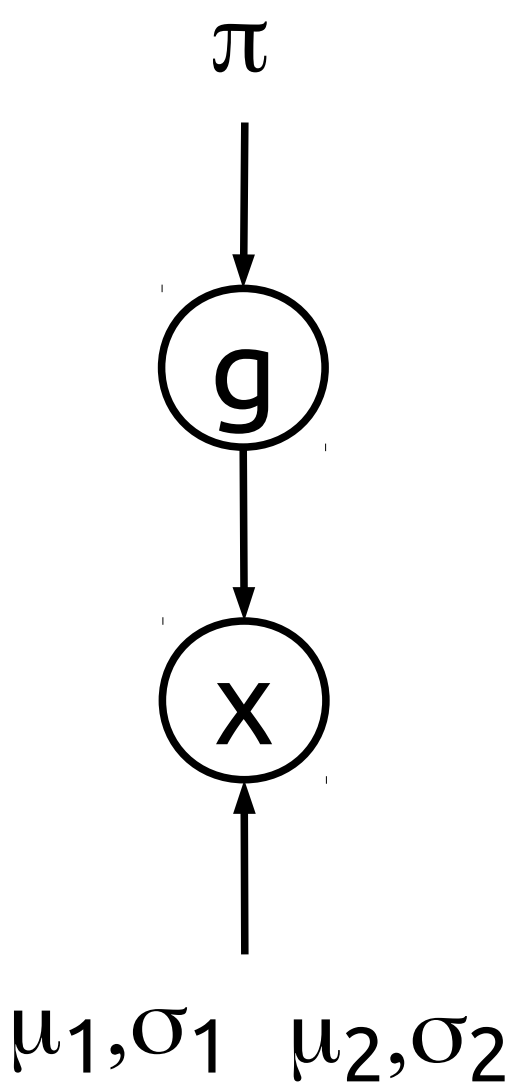
$$p(g=1)=\pi_1 \qquad p(g=2)=\pi_2$$

- p(x|g) = the two Gaussian probabilities

$$p(x|g=1)=\mathcal{N}(x|\mu_1,\sigma_1)$$
$$p(x|g=2)=\mathcal{N}(x|\mu_2,\sigma_2)$$

GMM as Bayesian Networks

The full Bayesian Network

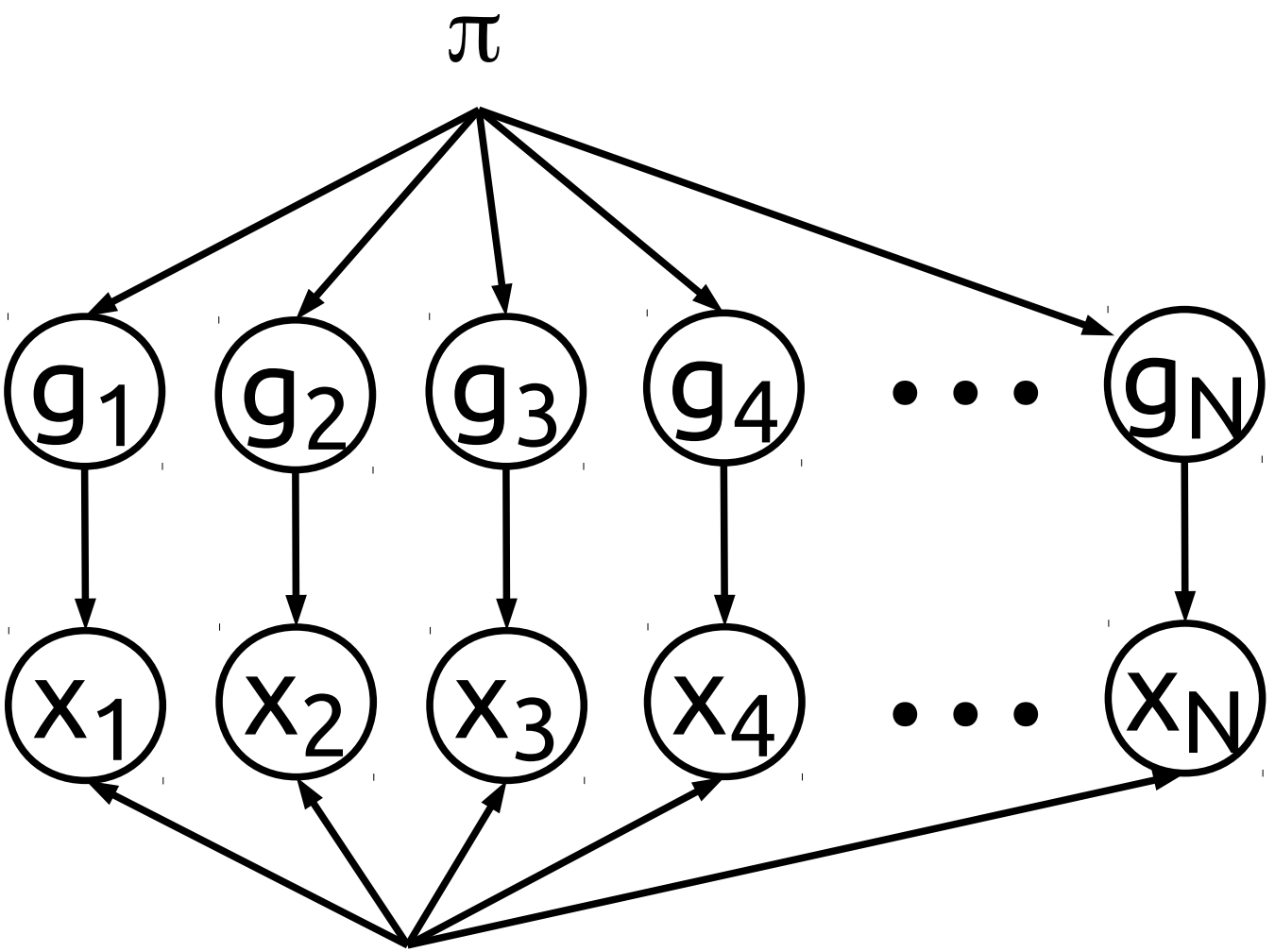


GMM as Bayesian Networks

- NOTE: This represents a Bayesian Networks for **one** point

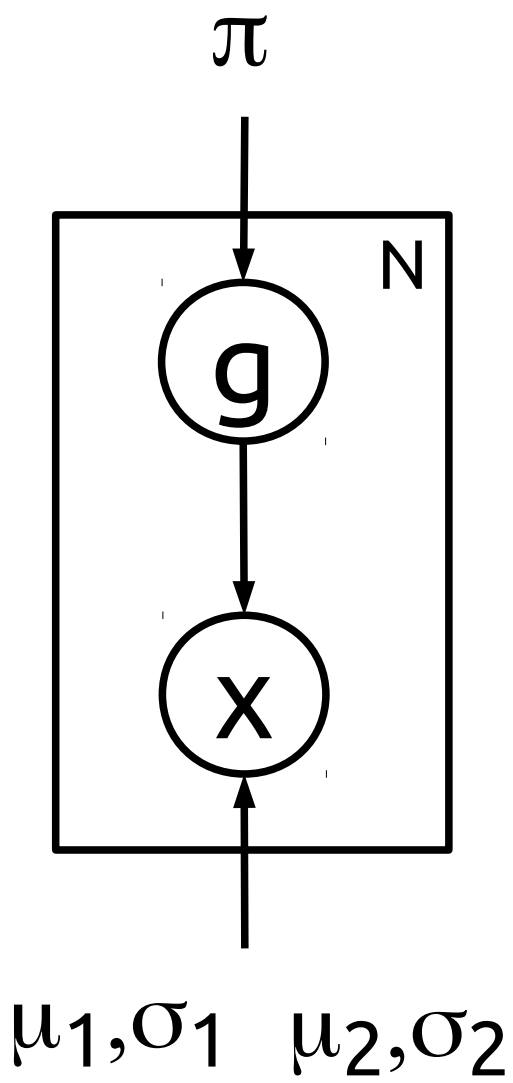
- As in the other example (the two boxes case), here we want to model **a set** of points $x_1, \ldots x_N$, all generated with the scheme described before
  - This means that there is a pair of variables (x,g) for every point of the dataset

GMM as Bayesian Networks



GMM as Bayesian Networks

Again, with the **plate notation**



Hidden Markov Models

- Widespread approach for sequence modelling
- Main features:
  - the intrinsic capability of dealing with sequential evolution
  - the effective and fast training algorithm (based on Expectation-Maximization – EM)
  - the clear Bayesian semantic interpretation
  - if properly used, they work very well in several practical applications

Hidden Markov Models

- Introduced by Baum *et al.* at the end of 60's
  - Baum & Petrie (1966), Baum & Egon (1967), Baum & Sell (1968), Baum *et al.* (1970), Baum (1970)
- Initially almost employed in speech recognition applications
  - Levinson et al. (1983), Rabiner (1989), Rabined & Juang (1993)
- Starting from 90's, largely (and successfully!) applied in several fields

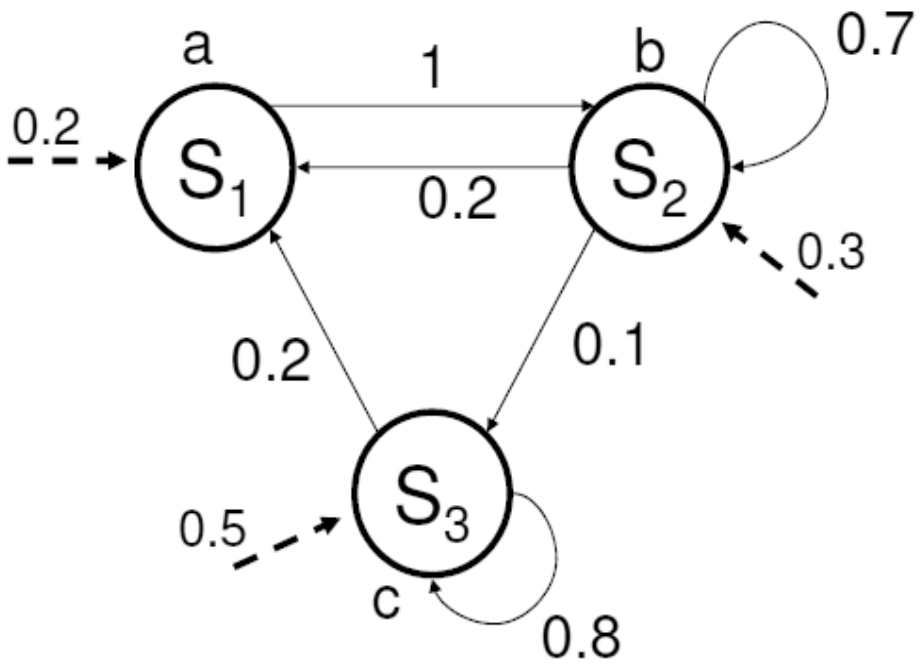
Hidden Markov Models

Applications

- Handwritten character recognition: on-line and off-line
- Computer Vision: Image classification, Gesture recognition, Activity classification, Face recognition, 2D shape classification, Texture analysis, 3D object recognition from range images
- Signal processing, Robotics, Bioinformatics, Finance, Meteorology, Geomagnetism, Neural signal analysis, Acoustics, EEG signal modelling

Hidden Markov Models

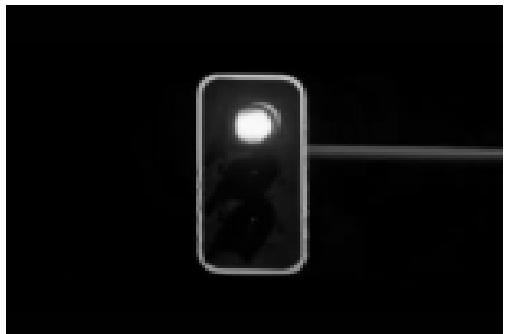
It represents an extension of a Markov Model



Markov Model: models systems which can be at every instant in one out of a possible set of states

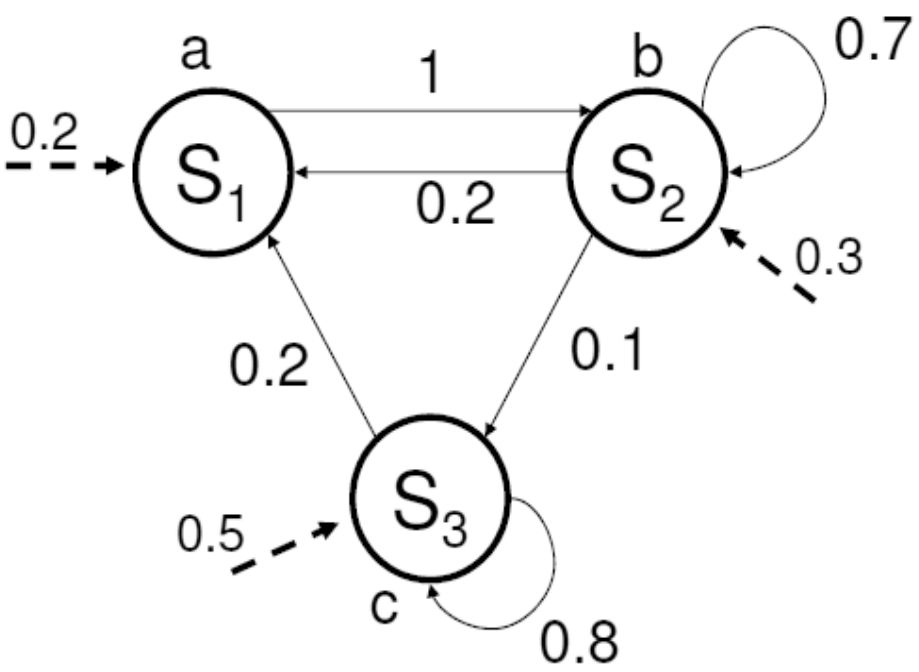
At every time step the system changes state

Classical example: the trafic light



Hidden Markov Models

It represents an extension of a Markov Model

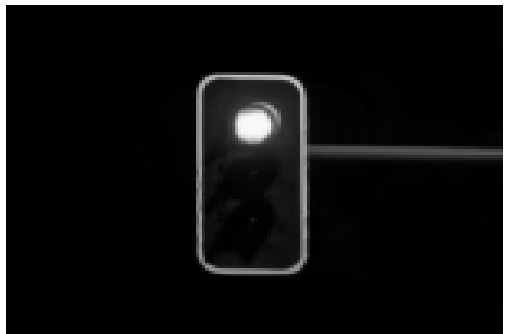


When the system enters in a state, a symbol is emitted (this represents the observation)

The state transition is governed by a probability

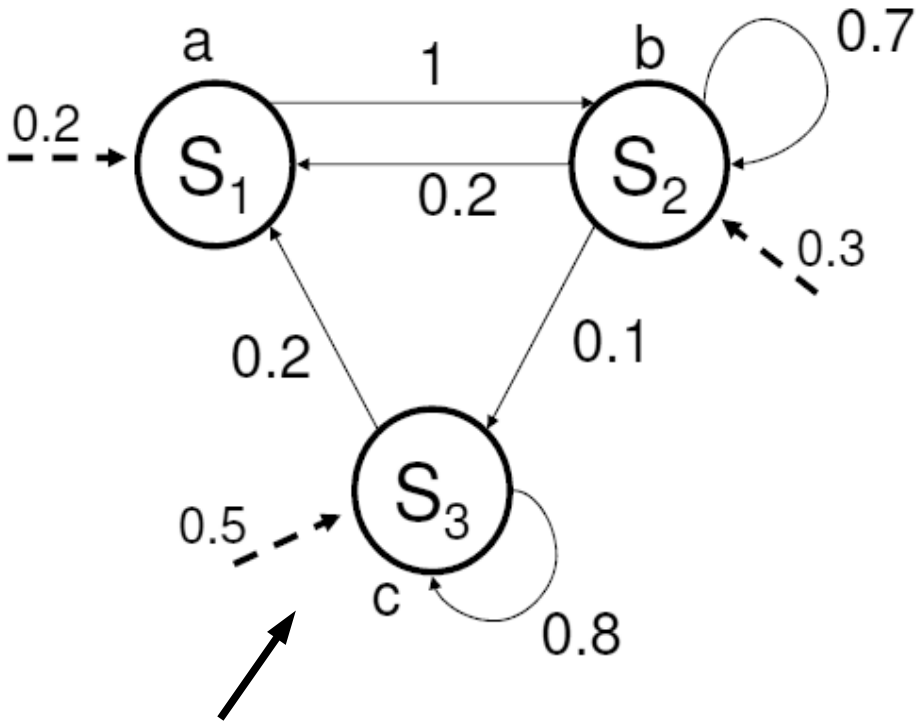
The next state depends only on the curent state (Markovianity of first order)

Classical example: the trafic light



Hidden Markov Models

- Important note: in Markov Models we have a different symbol emitted in every different state



It the systems enters in state S2, then the symbol “b” is emitted

NOTE: this is NOT a representation following the Bayesian Network formalism!

Hidden Markov Models

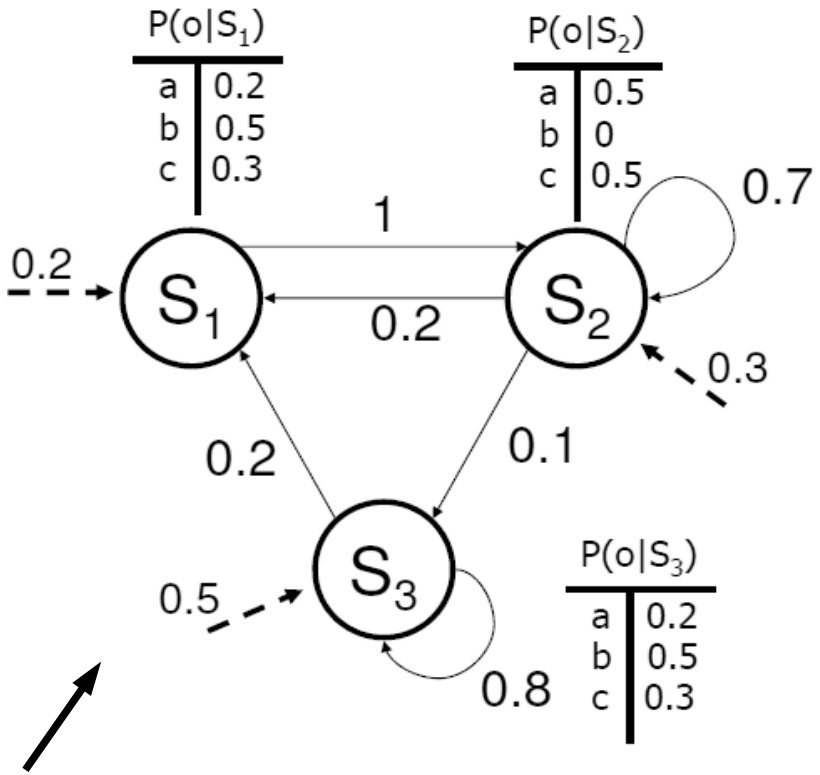
- Consequence: by observing a set of symbols emitted from a Markov Model it is possible to derive the states

a	b	b	b	b	c	c	c
S1	S2	S2	S2	S2	S3	S3	S3

- The states of a Markov Model are **observable**!

Hidden Markov Models

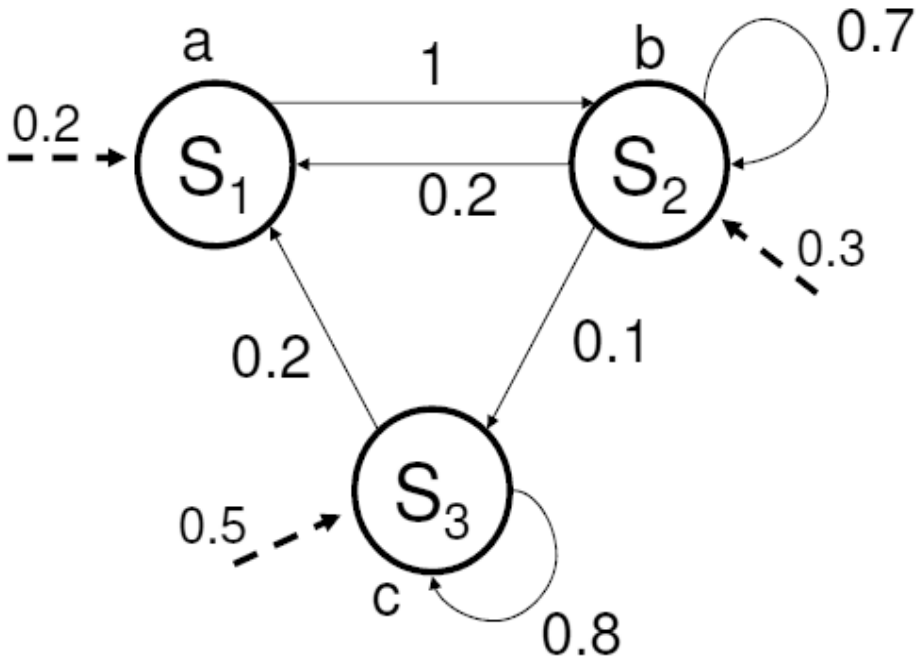
- A Hidden Markov Model (HMM) is a Markov Model where the states are **not observable** (i.e. they are **hidden**)



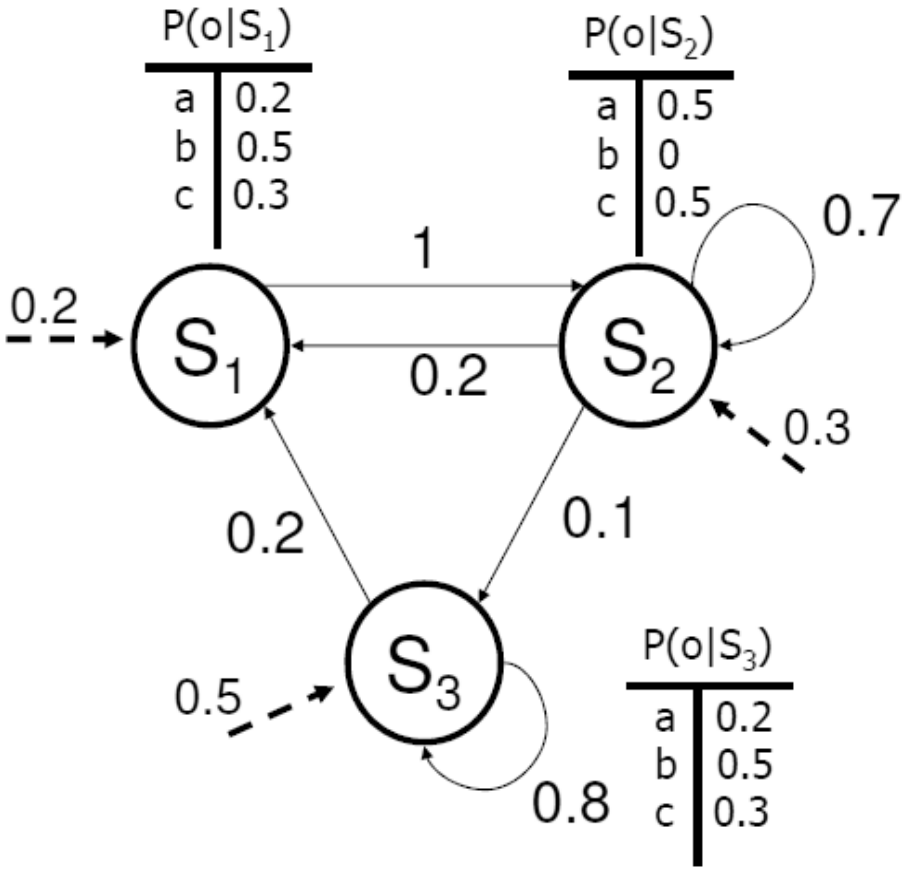
In other words, when the system enters in a given state, there is a distribution probability describing the probability of emitting the symbols

NOTE: this is NOT a representation following the Bayesian Network formalism!

Hidden Markov Models



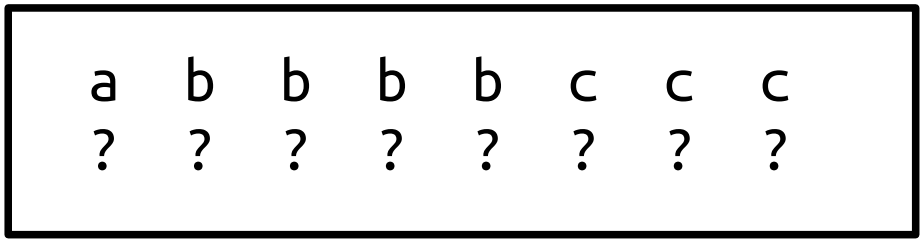
Markov Models



Hidden Markov Models

Hidden Markov Models

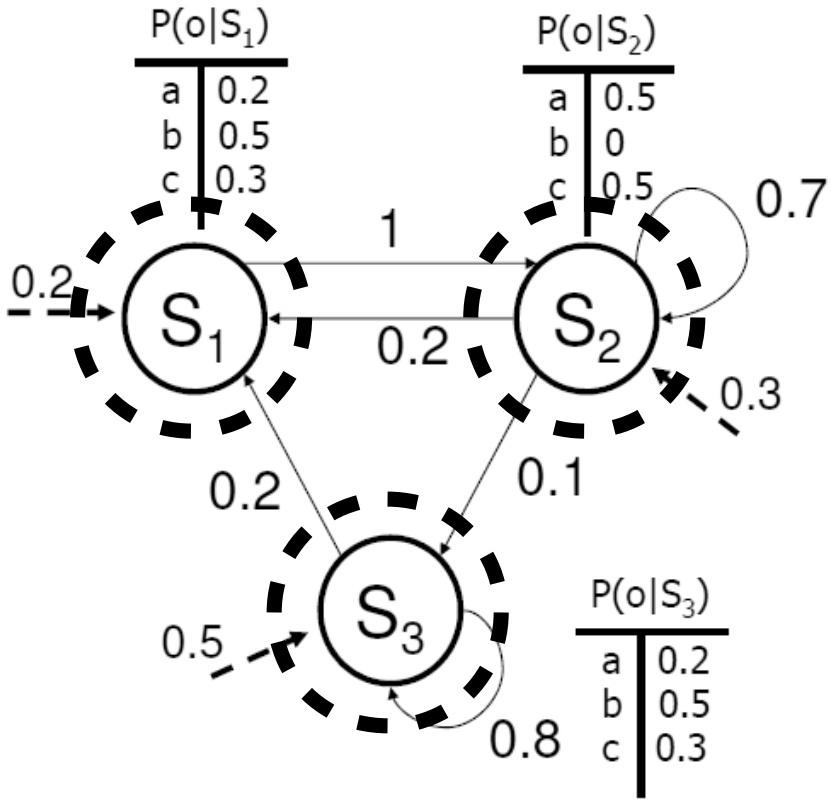
- Consequence: by observing a set of symbols emitted from a Hidden Markov Model it is **not possible** to derive the states



- The states of a Hidden Markov Model are **hidden**!

HMM components

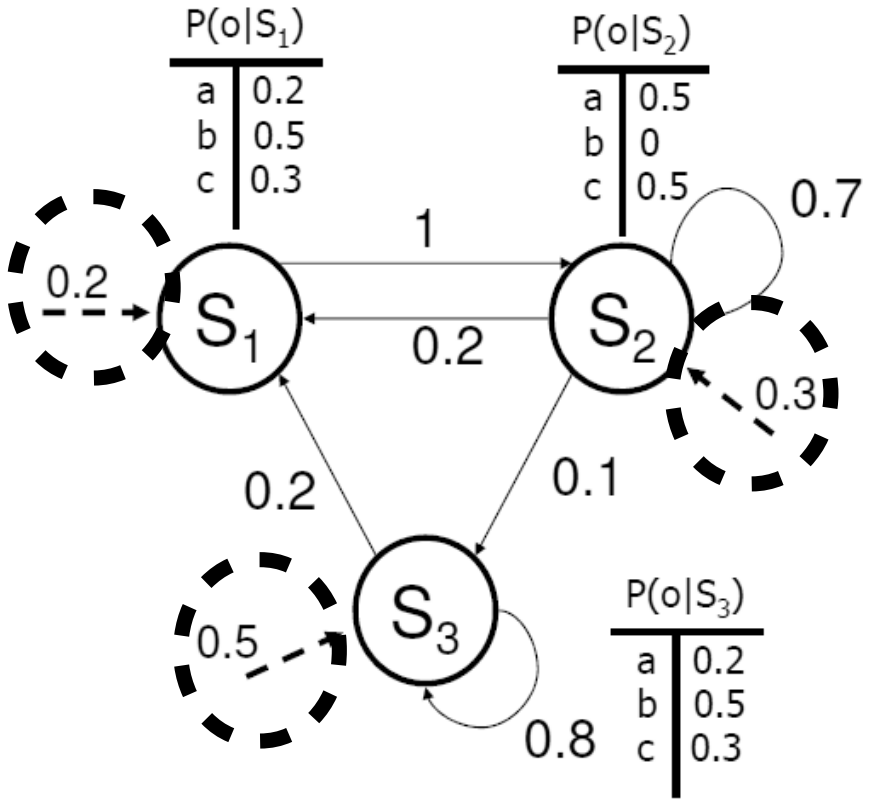
- A set $S=\{S_1,S_2,...,S_N\}$ of (hidden states)
  - even if hidden, for many practical applications a physical meaning could be inferred



HMM components

- An initial state probability distribution $\boldsymbol{\pi} = \{\pi_i\}$

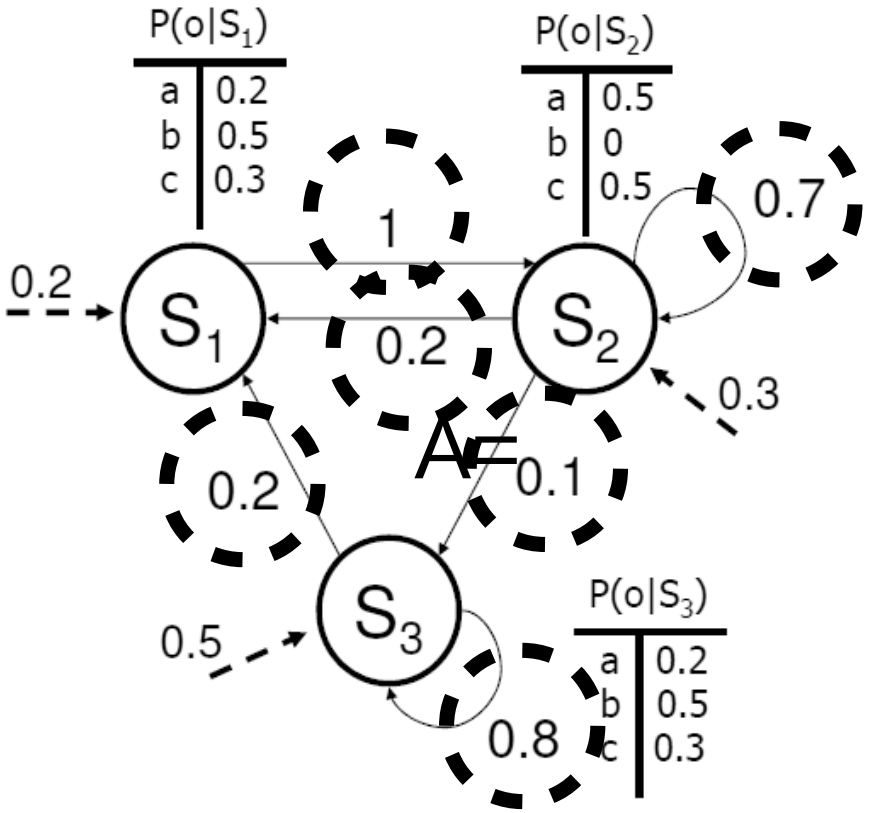
$$\pi_i = P(Q_1 = S_i) \qquad 1<i<N$$

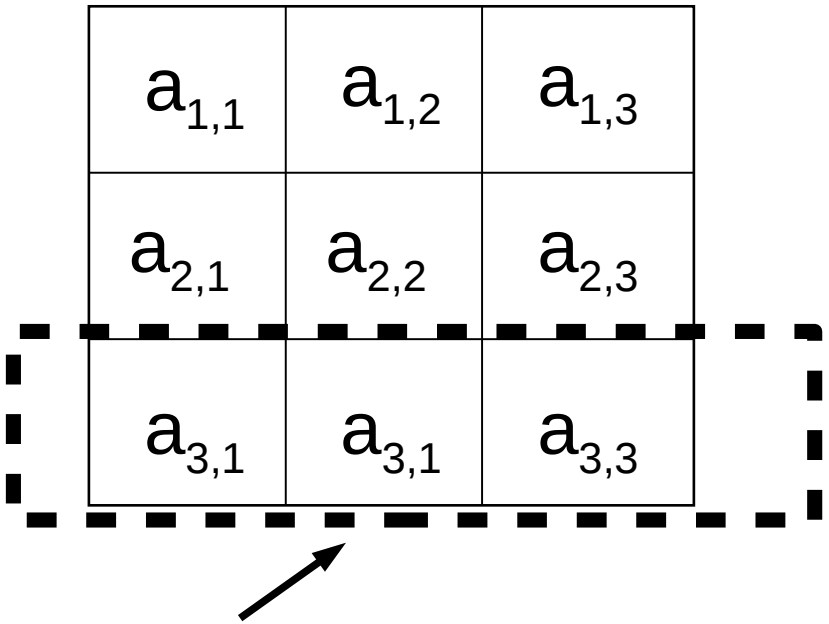


HMM components

- A state transition probability distribution $\mathbf{A} = \{a_{ij}\}$

$$a_{ij} = P(Q_t=S_j|Q_{t-1}=S_i) \qquad 1<i,j<N$$



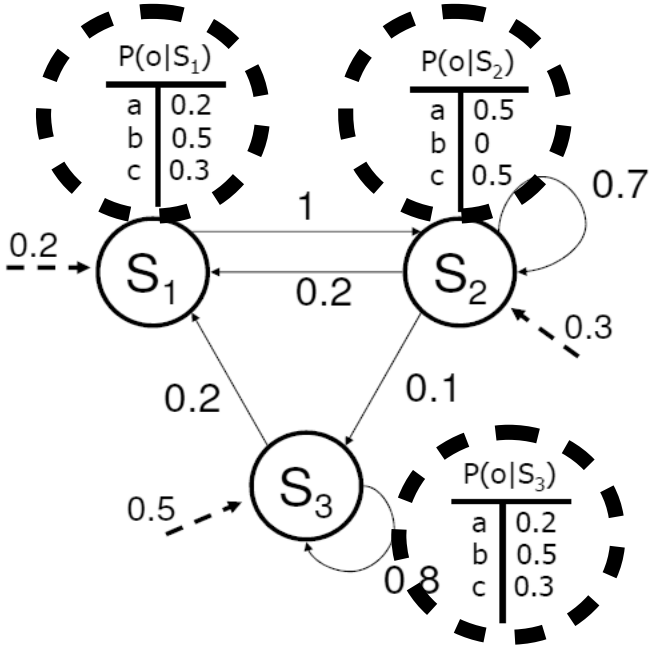


HMM components

- An observation symbol probability distribution **B**={$b_j(v)$}

$$b_j(v) = P(v \text{ is emitted at time } t \mid Q_t = S_j)$$

v belongs to the set of observation symbols

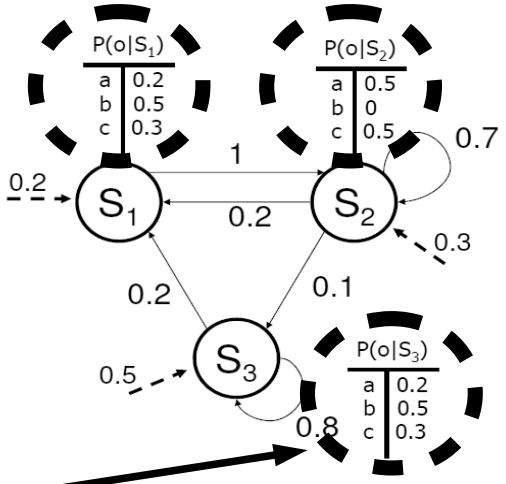


HMM generative process

Generative process: how to sample from a HMM?

HMM assumes that every symbol of the sequence of observations derives from one state, even if we don't know which one

- First symbol (t=1):
  - select the first state from the “initial state probability”
- Sample a point from the “emission probability” of the selected state



If we have selected the third state, then we should sample from this

HMM generative process

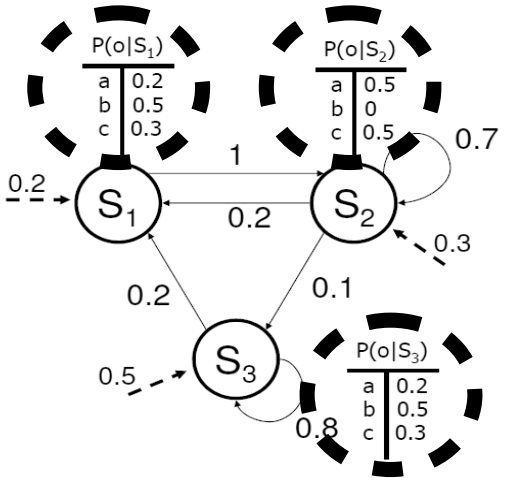
All other symbols (t>1):

- select the state from the “transition probability” corresponding to the previous state (t-1)

If the system is in the third state, then we should sample from this →

$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,1}$	$a_{3,1}$	$a_{3,3}$

- Sample a point from the “emission probability” of the selected state



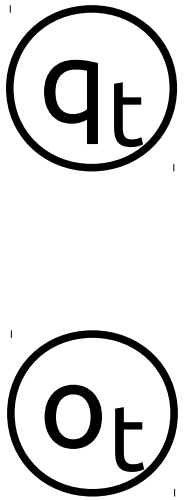
HMM as Bayesian Networks

- Goal: create a BN that models a sequence of T symbols emitted from a HMM
- We have one different variable for every symbol emitted
- We have one different variable for every state

HMM as Bayesian Networks

Variables:

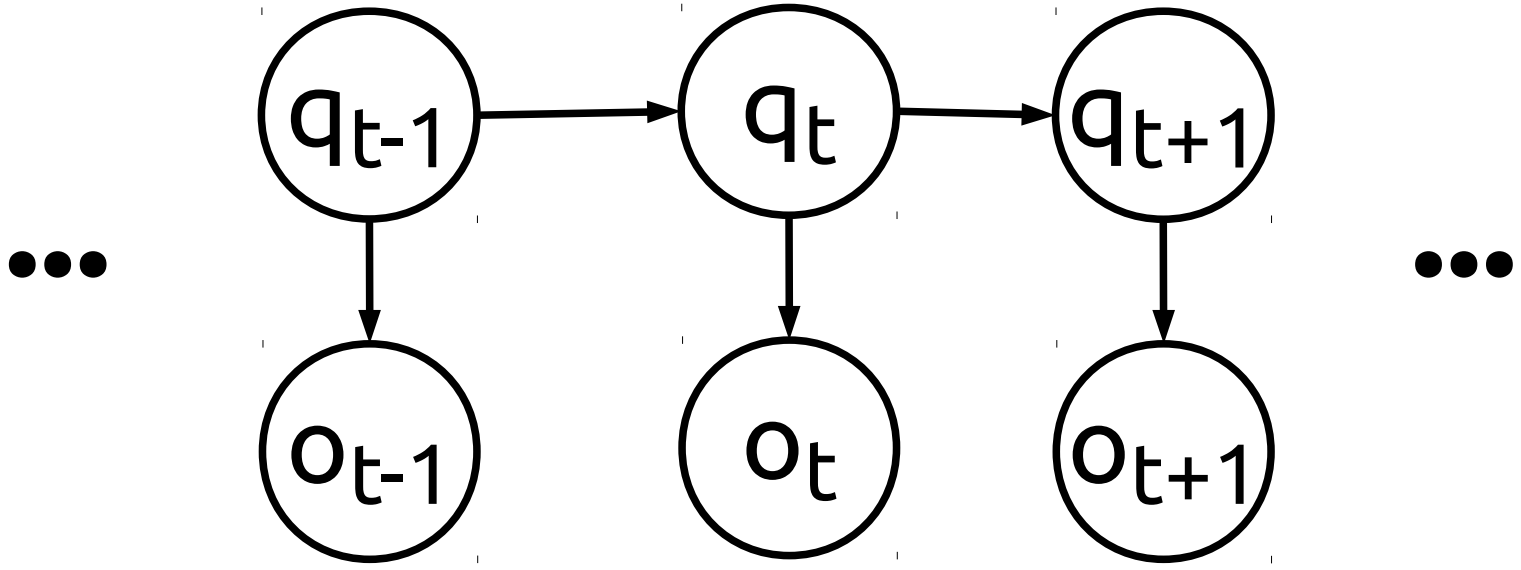
- o_t = variable usable to describe the **symbol** emitted at time t (**visible** variable)
- q_t = variable usable to describe the **state** at time t (**hidden** variable, discrete)



HMM as Bayesian Networks

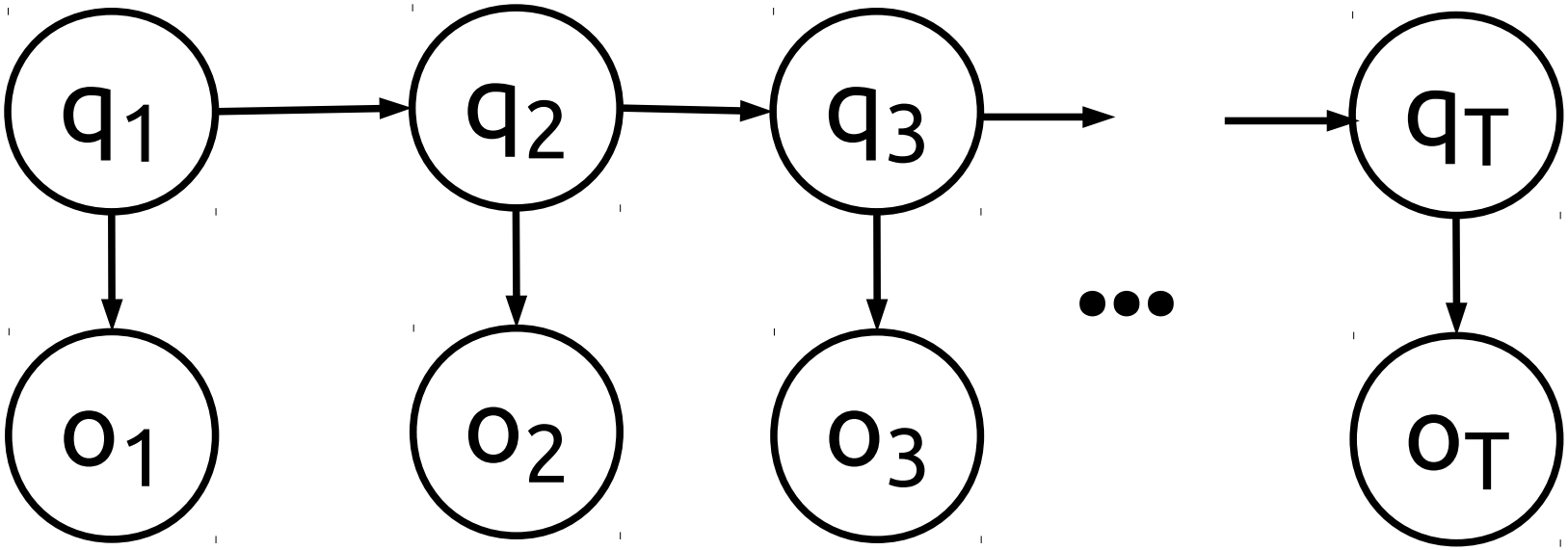
Edges:

- The symbol emitted at time t $\mathbf{o_t}$ depends on the state at time t $\mathbf{q_t}$ (edge from $\mathbf{q_t}$ to $\mathbf{o_t}$)
- The state at time t $\mathbf{q_t}$ depends on the state at time t-1 $\mathbf{q_{t-1}}$ (edge from $\mathbf{q_{t-1}}$ to



HMM as Bayesian Networks

- For a sequence of lenght T (T symbols), the BN of the HMM is defined as



HMM as Bayesian Networks

The conditional probabilities are defined by using the prior probability, the transition probability and the emission probability