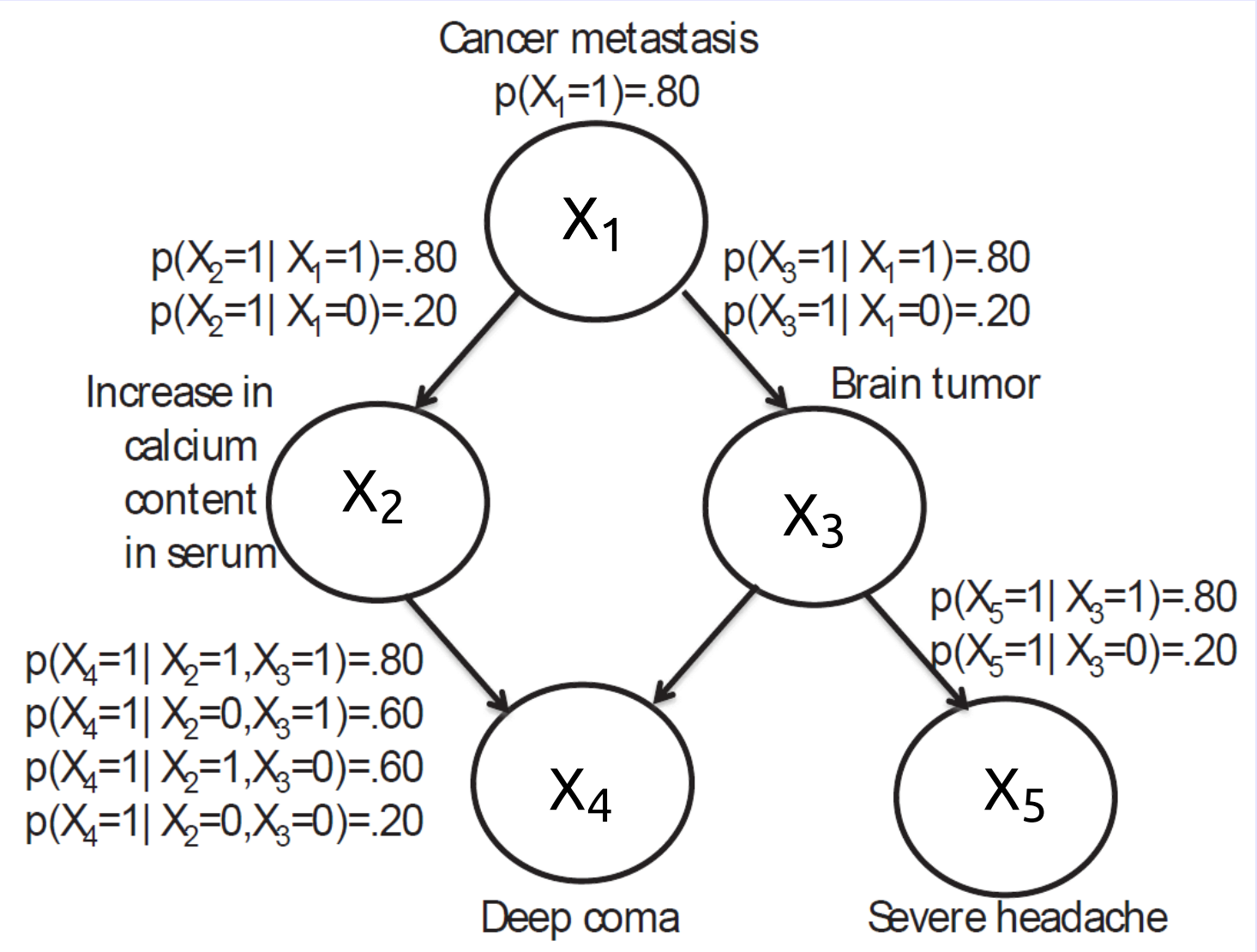
Many BN for different problems!

Bayesian Networks have been applied in many different scenarios!



One example: Topic Models

- Widely used Bayesian Networks firstly introduced in the context of text processing
- Typically they are employed on top of the Bag of Words Representation

… let's introduce them from the text processing perspective

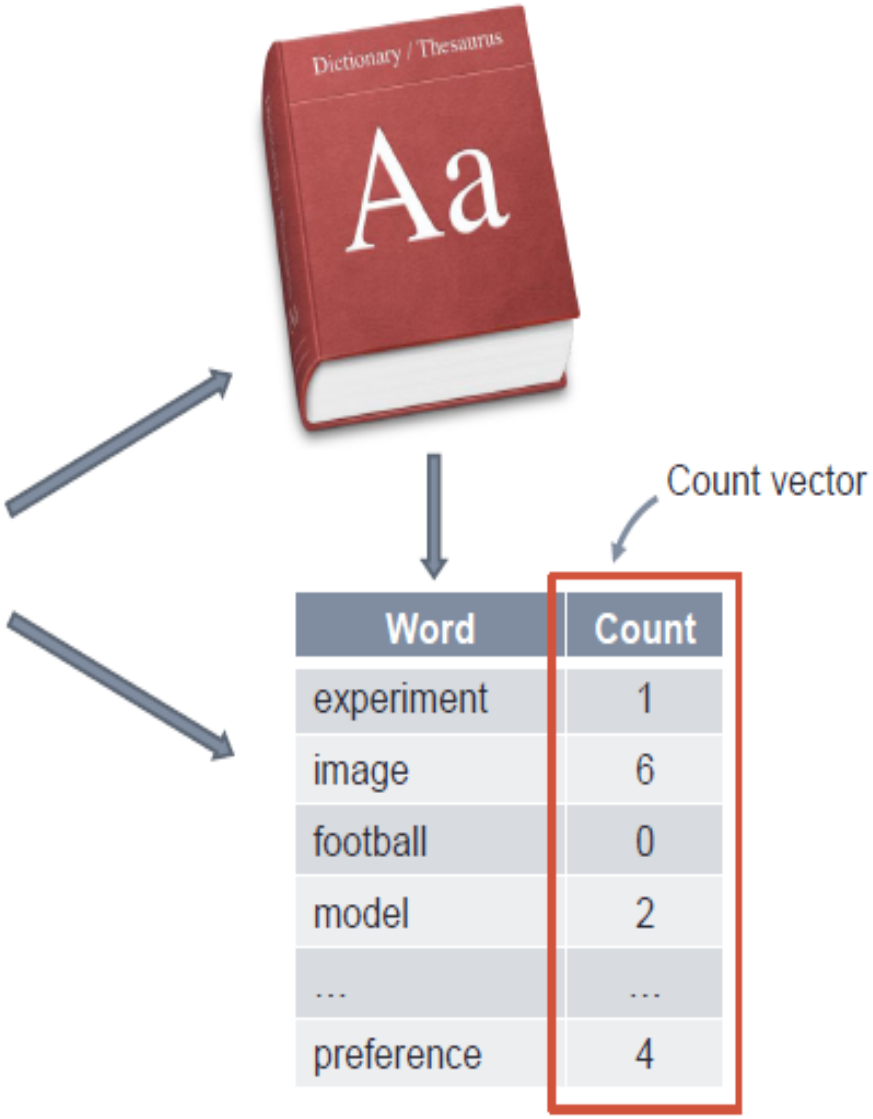
Reminder: Bag of Words

ABSTRACT

Modeling preferences in photographic images is often reduced to analyzing intermediate explicit representations (e.g. textual tags) as means of capturing the objective and subjective properties of image perception, trying to distill the essence of what gives pleasure. We propose an alternative approach that bypasses the necessity to build an explicit conceptual coding of image preferences, operating directly on the raw properties of the images, extracted with heterogeneous feature descriptors. This is achieved through the *counting grid* model, which fuses together content-based and aesthetics themes into a 2D map in an unsupervised way. We show that certain locations in this map correspond to perceptually intuitive image classes, even without relying on tags or other user-defined information. Moreover, we show that users' individual preferences can be represented as distributions over the map, allowing us to evaluate the affinity between different users' appreciations. We experiment on a large Flickr dataset, clustering users by affinity, and validating these clusters by checking users that belong to the same Flickr photo groups.

1. INTRODUCTION

Nowadays, people commonly enjoy watching and sharing images or photos when browsing popular social networks, often expressing preferences for pictures they like.



Problem with BOW

- Problem: a single word may have a different meaning depending on the context

Sun?

Sun microsystems

Windows?

"Home"	"sports"	"space"	"computers"	"weather"
Kitchen	Team	Space	Drive	Rain
Door	Game	Sun	Windows	Snow
Garden	Play	Research	Card	Sun
Windows	Year	Center	DOS	Season
Bedroom	Games	Earth	SCSI	Weekend
Space	Season	NASA	Sun	Cloudy

The solution: topic models

- Solution: we can disambiguate these situations by looking at the **context**
  - The context can be inferred by looking at **co-occurrence** of words
- Example: if the word “windows” appears together with “beds” then we are speaking about “homes”, if it appears with “browser” we are speaking about “computers”
- Topic Models model the context by using the concept of “Topics”

The solution: topic models

- A topic represents “what we are talking about”
- Topic models introduce an intermediate level between words and documents, referred to as **topics**
  - Every document is characterized by the presence of different topics (Document “A” speaks about “sport” (60%) and “finance” (40%))
  - Every topic induces the use of a particular set of words (when we are speaking about “sport” we are using quite often the word “soccer”)
- Topics are extracted in an automatic way by looking at co-occurences of words

The solution: topic models

- Different “topic models” have been introduced in the past: probabilistic Latent Semantic Analysis, Latent Dirichlet Allocation, Counting Grids, Discriminative Latent Dirichlet Allocation, Latent Process Decomposition...
  - They differ in the way the topics are found

- Let's see here the probabilistic Latent Semantic Analysis – pLSA - model (the first and most famous one)

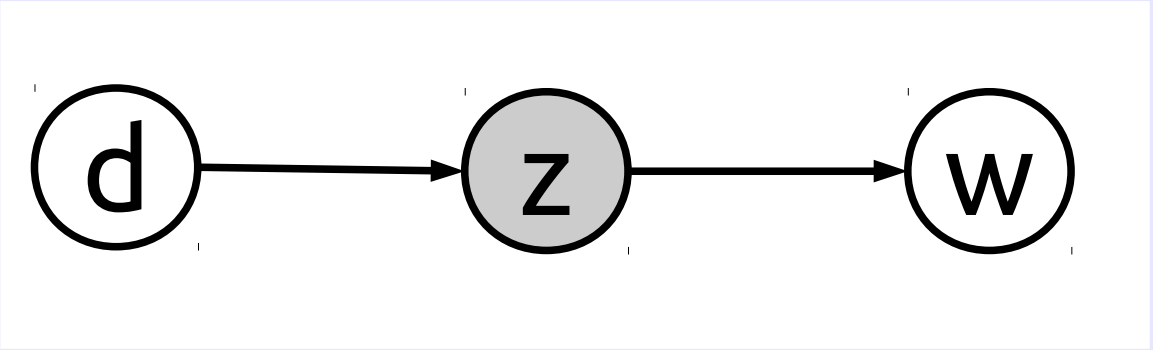
probabilistic Latent Semantic Analysis (pLSA)

The ingredients: variables and edges.

- d: variable used to describe the document (**visible**)
- w: variable used to describe the word (**visible**)
- z: variable used to describe the topic (**hidden**)

pLSA

- Clearly the words depend on the topics which depend on the document



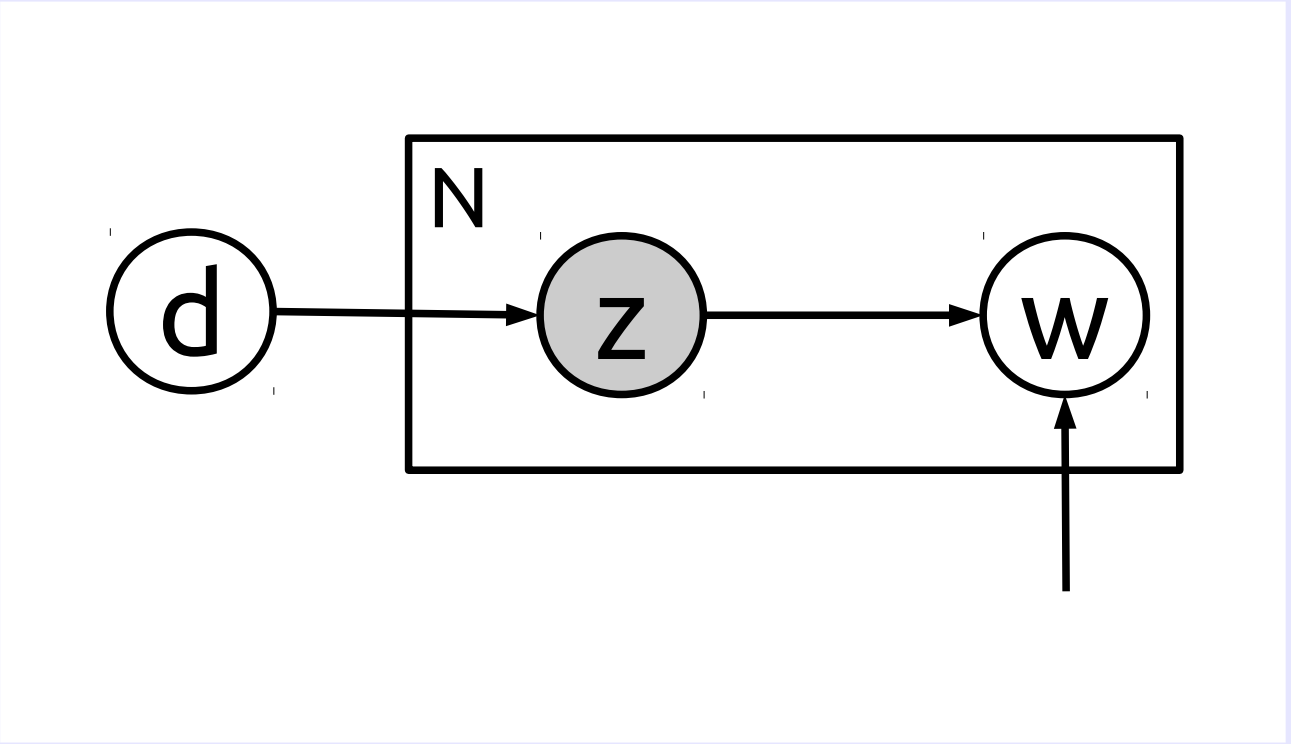
pLSA

- A document is formed by N words
- Every word depends on the topic
- The probability of observing the words in the different topics is parametrised with β (the same for all words)

$$\beta = \begin{bmatrix} P(w = \text{'windows'} \mid z = 1) \\ \dots \\ P(w = \text{'sky'} \mid z = 1) \\ \dots \\ P(w = \text{'windows'} \mid z = 2) \\ \dots \\ P(w = \text{'sky'} \mid z = 2) \end{bmatrix}$$

pLSA

- The full Bayesian Network for one document



pLSA: an interesting probability

- What is the probability of observing a particular word w_i in a document d? I.e. we are interested in $p(w_i|d)$

$$
\begin{aligned}
p(w_i|d) &= \frac{p(w_i, d)}{p(d)} \\
&= \frac{\sum_z p(w_i, z, d)}{p(d)} \\
&= \frac{\sum_z p(d)p(z|d)p(w_i|z)}{p(d)} \\
&= \sum_z p(z|d)p(w_i|z)
\end{aligned}
$$

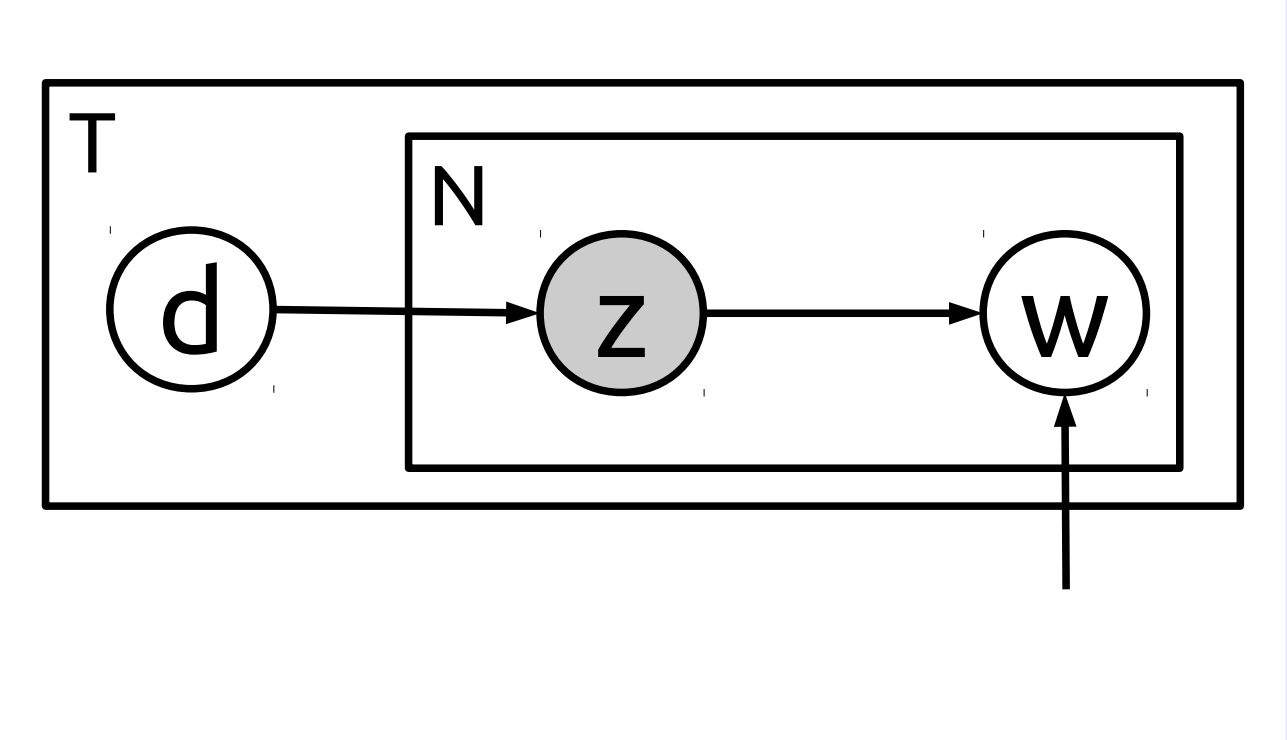
Definition of conditional probability

Marginalization

Factorization of the joint probability as given by the Bayes network

pLSA

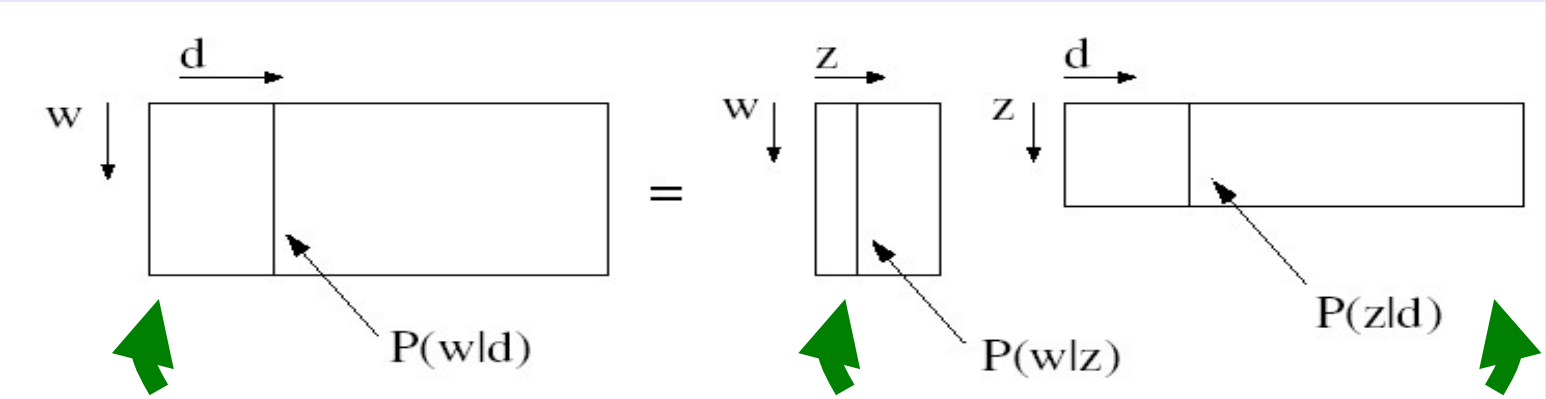
- The Bayesian Network for T documents



pLSA

- The probability of observing a particular word w_i in a document d_j is

$$p(w_i|d_j) = \sum_{k=1}^{K} p(z_k|d_j)p(w_i|z_k)$$



Observed word distributions

word distributions per topic

Topic distributions per document

pLSA

- How to build the pLSA (training): estimation of the probabilities p(w|z) e p(z|d)
  - This is done by starting from the Bag of Words representation of the documents
- How to use a trained pLSA (inference):
  - Given a document, we can understand which are the topic spoken (using p(z|d))
  - Given a topic, we can understand which are the most used words inside that topic (using the p(w|z))
- Key feature: interpretability!!!

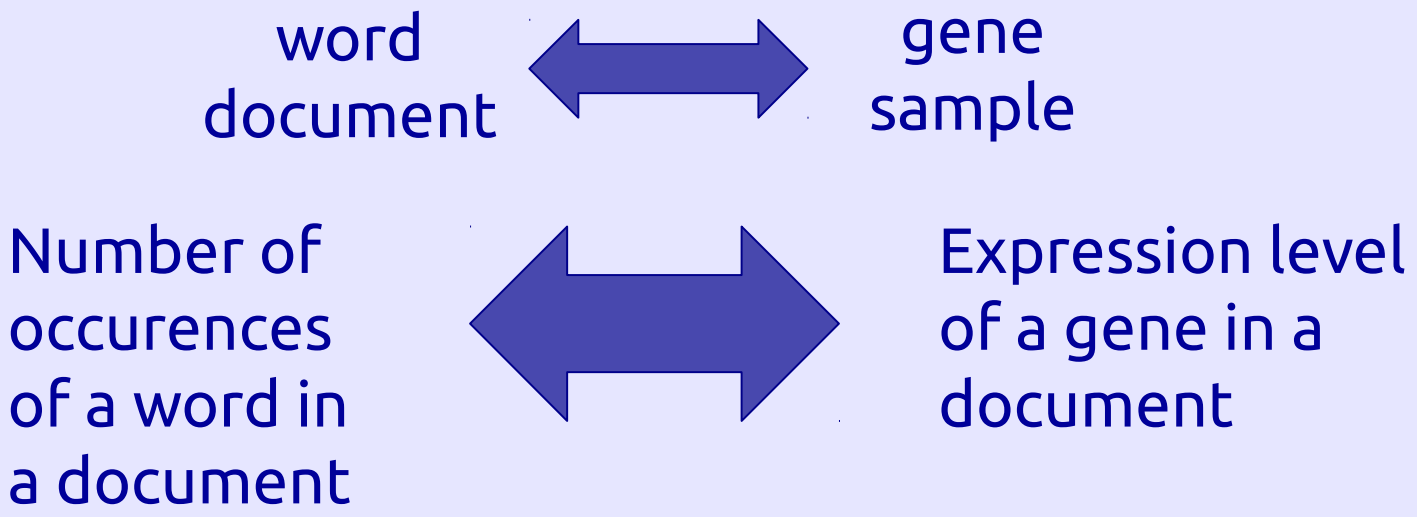
pLSA for gene expression

- Example: using pLSA (or more in general Topic Models) to model gene expression
- Starting point: we can postulate an analogy between words/documents and genes/samples

- A document is characterized by the different occurrences of the dictionary words
- A sample is characterized by the different expression of the different genes

pLSA for gene expression

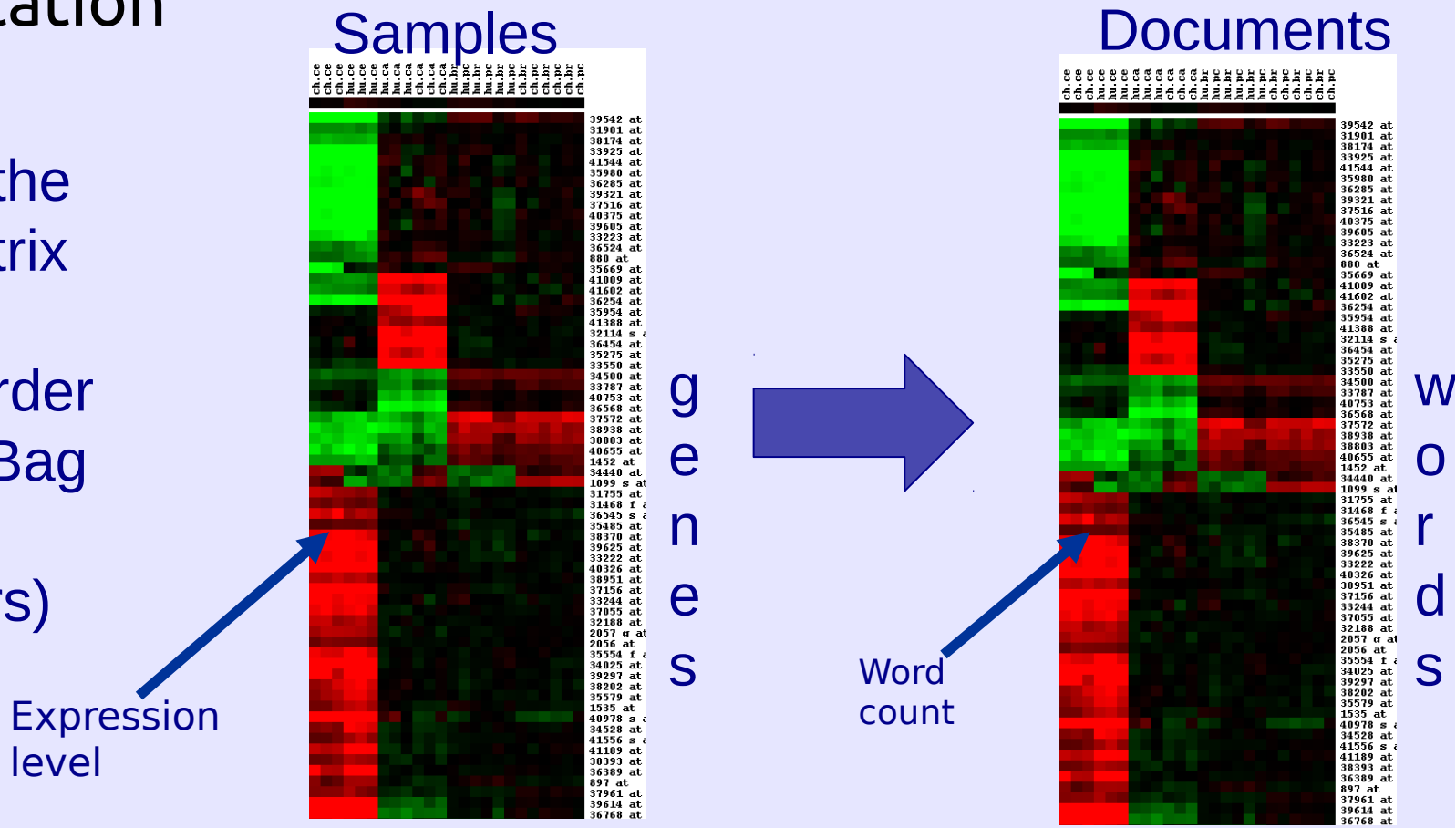
- We can set the analogy “words = genes” and “documents = samples”



pLSA for gene expression

- We can use pLSA for gene expression by considering the expression matrix as a Bag of Words representation

NOTE: clearly the expression matrix has to be pre-processed in order to have a true Bag of Words (i.e. positive integers)



pLSA for gene expression

- Key feature: interpretability!
- We can interpret the topics as “biological processes” occurring in the samples
  - A biological process is active only in few samples (namely the documents which speak about that topic)
  - A biological process involves only some genes (namely the words which are used when such topic is spoken)

pLSA for gene expression

- We can derive interesting information from the probabilities
  - **P(z|d)** can be used to infer which are the active processes in a given sample (and to which extent they are active)
  - **P(w|z)** can be used to measure the impact of the different genes on a given biological process

Learning and Inference in Bayesian Networks

Learning and Inference

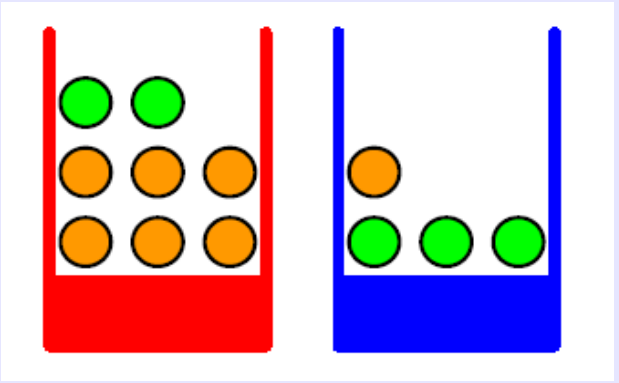
Two main tasks when dealing with Bayesian Networks

- **Learning:** the problem of “building” the model
  - how to set the parameters? Typical solution: to exploit a set of objects sampled from the problem to estimate them (learning from examples)
- **Inference:** the problem of “querying” the model
  - Compute interesting probabilistic relations or values of variables in the model. Typically the inference is performed once the model is learnt

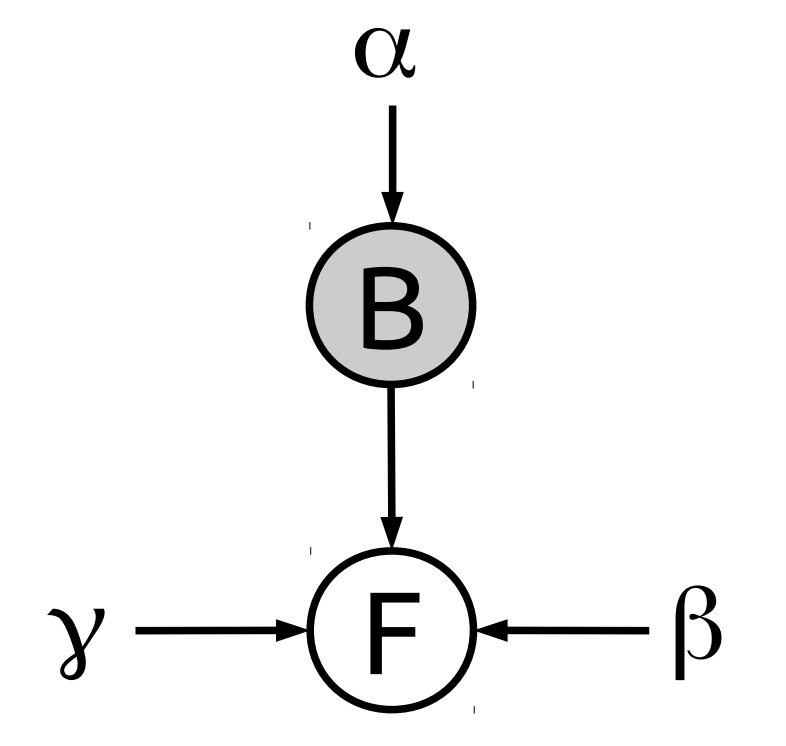
Learning

Example 1: the "**Two Boxes 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 (with re-integration of the fruit in the box)*

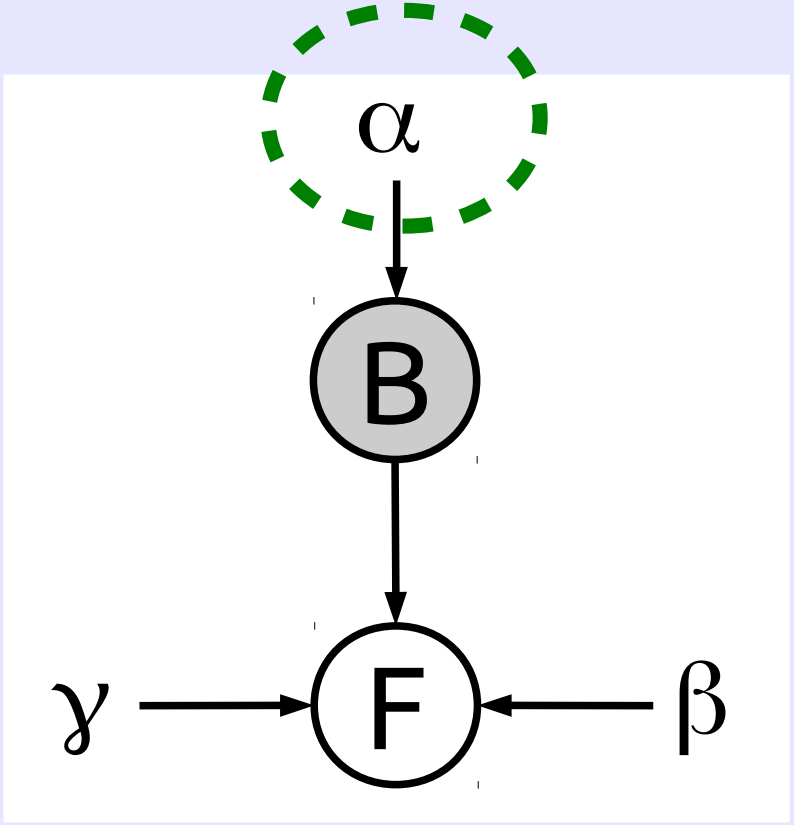


Learning



The problem is to estimate the parameters α, β, and γ

Learning



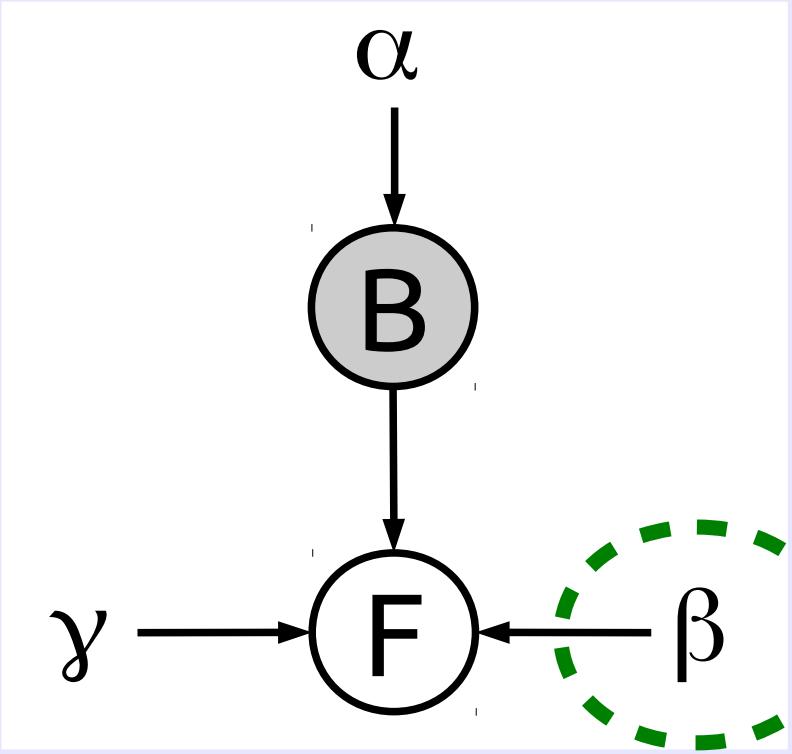
P(B)

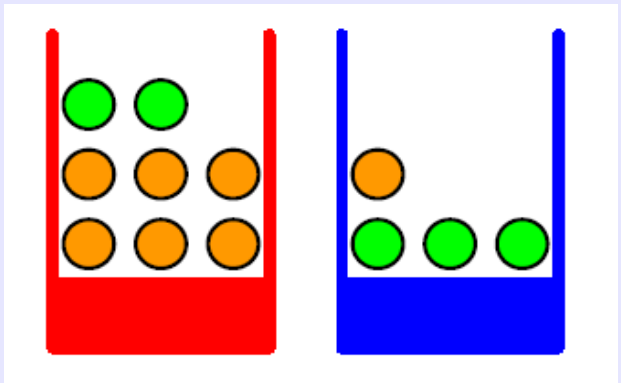
```
P(B = 'b') = α
P(B = 'r') = 1-α
```

α represents the probability of choosing the blu box

We can assume that the two boxes are equivalent (i.e. α = 0.5)

Learning





P(F|B)

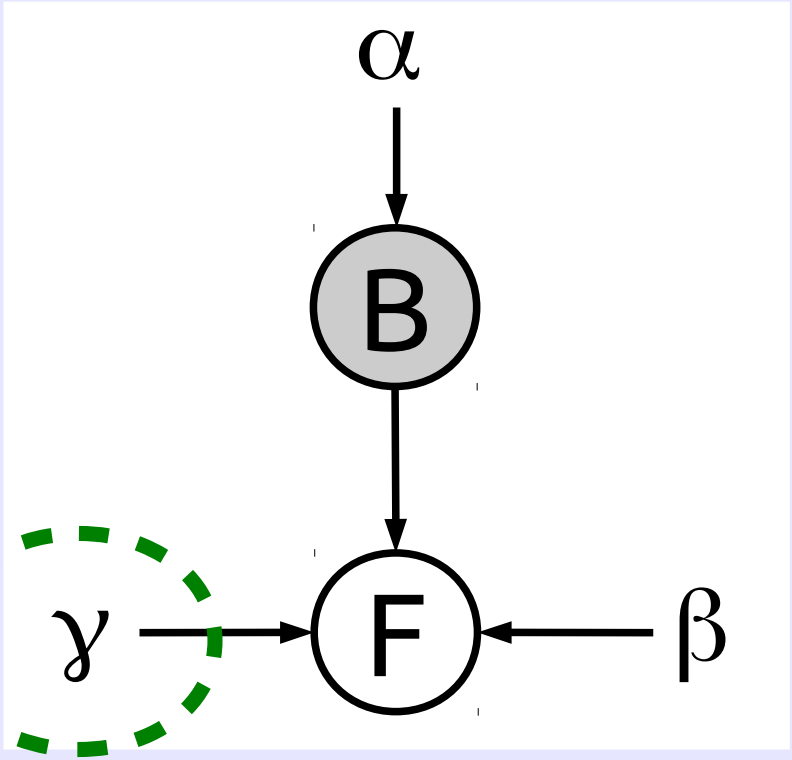
```
P(F = 'o' | 'B = 'r') = β
P(F = 'a' | 'B = 'r') = 1-β
P(F = 'o' | 'B = 'b') = γ
P(F = 'a' | 'B = 'b') = 1-γ
```

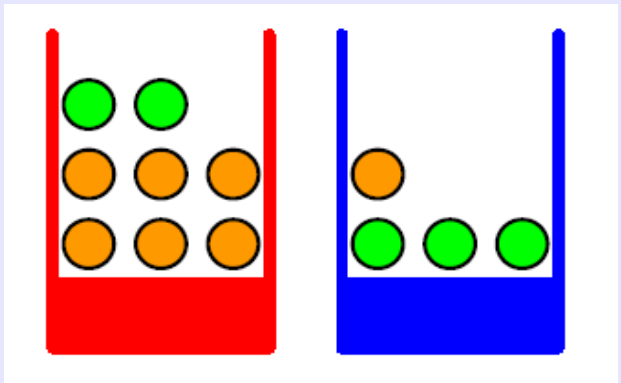
β represents the probability of extracting the orange from the red box

In the red box there are 6 oranges over 8 fruits

β = 0.75

Learning





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
```

γ represents the probability of extracting the orange from the blue box

In the blue box there is 1 orange over 4 fruits

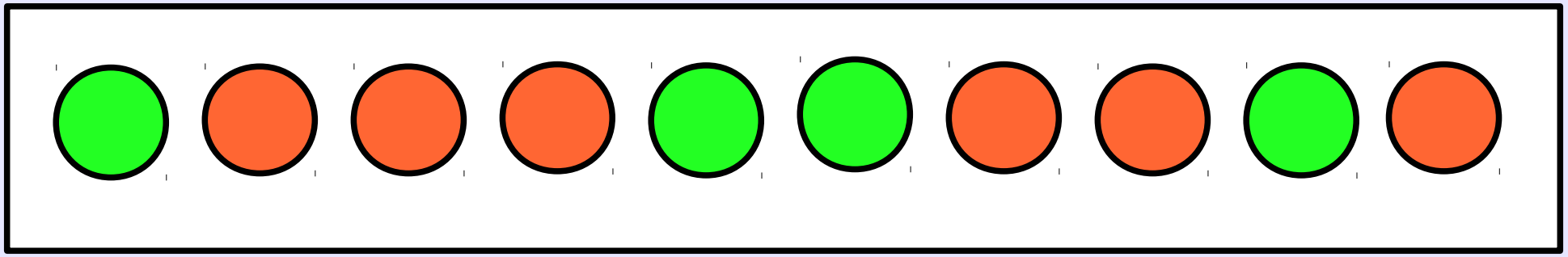
γ = 0.25

Learning

- In this case we learn the model using a priori knowledge (information we have)
  - This is not the case in many contexts!
- In alternative, the learning can be carried out **automatically** by using the “learning from examples” paradigm
  - We sample some objects from the problem (**Training set**), usable to estimate the parameters
- This represents a difficult task, especially if the BN contains hidden variables

Learning

Training set: some objects sampled from the problem.



In this case we don't have any a priori information on the content of the two boxes, we only observe the fruits (oranges/apples) – the **visible** variables!

Problem: we don't know from which box these fruits derive (the variable B is **hidden**)
The estimation of γ and β can be complicated!

Learning

- To learn Bayesian Networks with hidden variables we can resort to the Expectation – Maximization algorithm
- This represents a widely applied method to perform a **Maximum Likelihood** estimation of the parameters of a probabilistic model

.. let's briefly summarize the Maximum Likelihood estimation

Maximum Likelihood estimation

- The problem: given the training set **D**={x_1..x_N} (which contains objects sampled from the problem), the goal is to learn the model (i.e. the Bayesian Network), i.e. to estimate the parameters (of the BN)
- Let's call this set of parameters θ
  - In the “two boxes” problem, $\theta=\{\alpha,\beta,\gamma\}$
- Please note that, if we know θ the model is **completely specified** (we can compute p(x) for all objects)
- Goal: estimation of θ from **D**

Maximum Likelihood estimation

- Starting point: the likelihood function P(**D**|θ)
  - It represents the joint probability of all points in the training set
  - It clearly depends on the choice of the parameter θ (this is why it is denoted as P(**D**|θ))
- Assuming that all points in the training set are i.i.d (**independent** and **identically distributed**), the likelihood is defined as

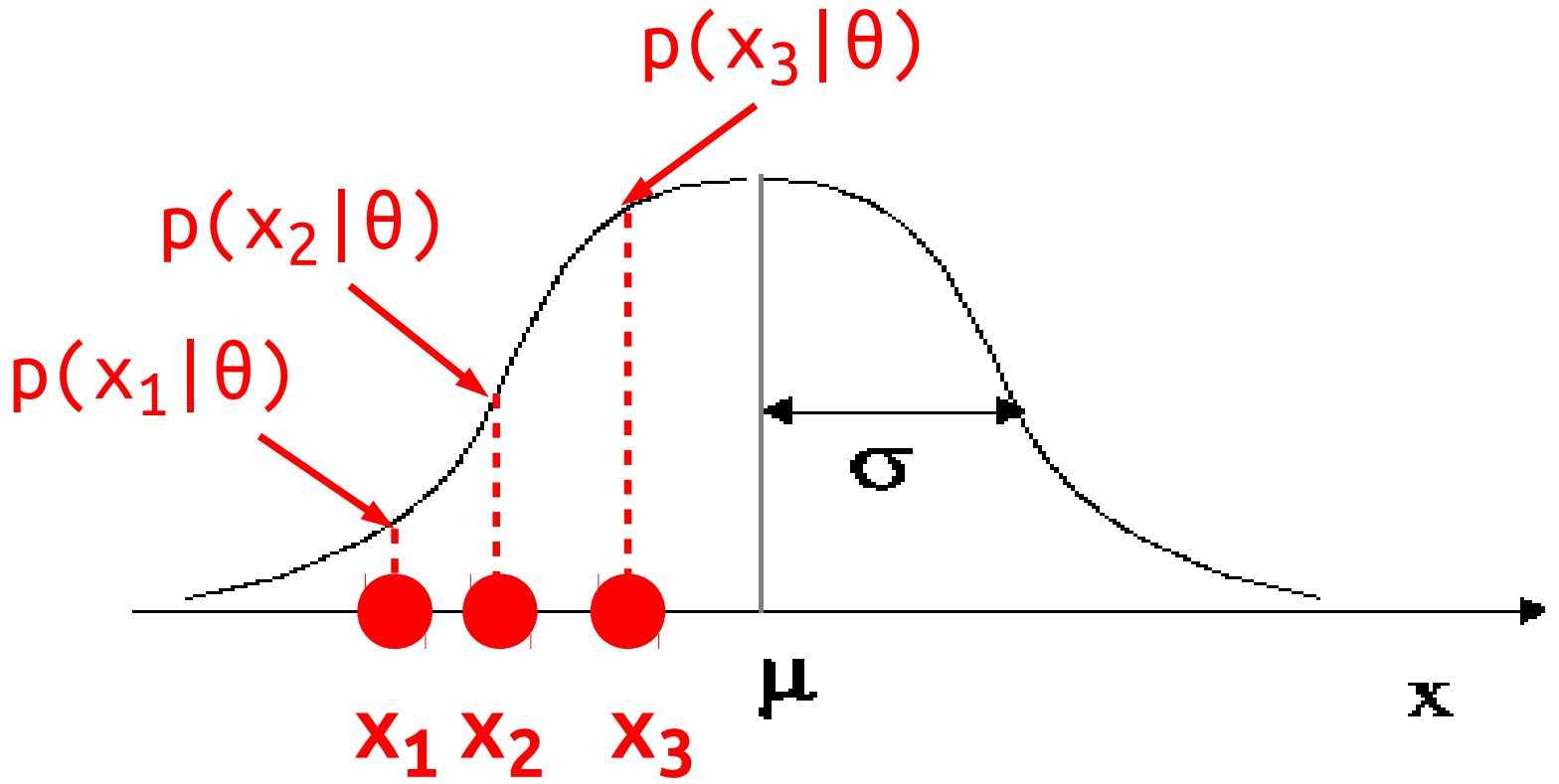
$$P(\mathbf{D}|\theta) = \prod_{i=1}^{N} p(x_i|\theta)$$

Example: 1D Gaussian, D contains 3 points

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

$$D = \{x_1, x_2, x_3\}$$

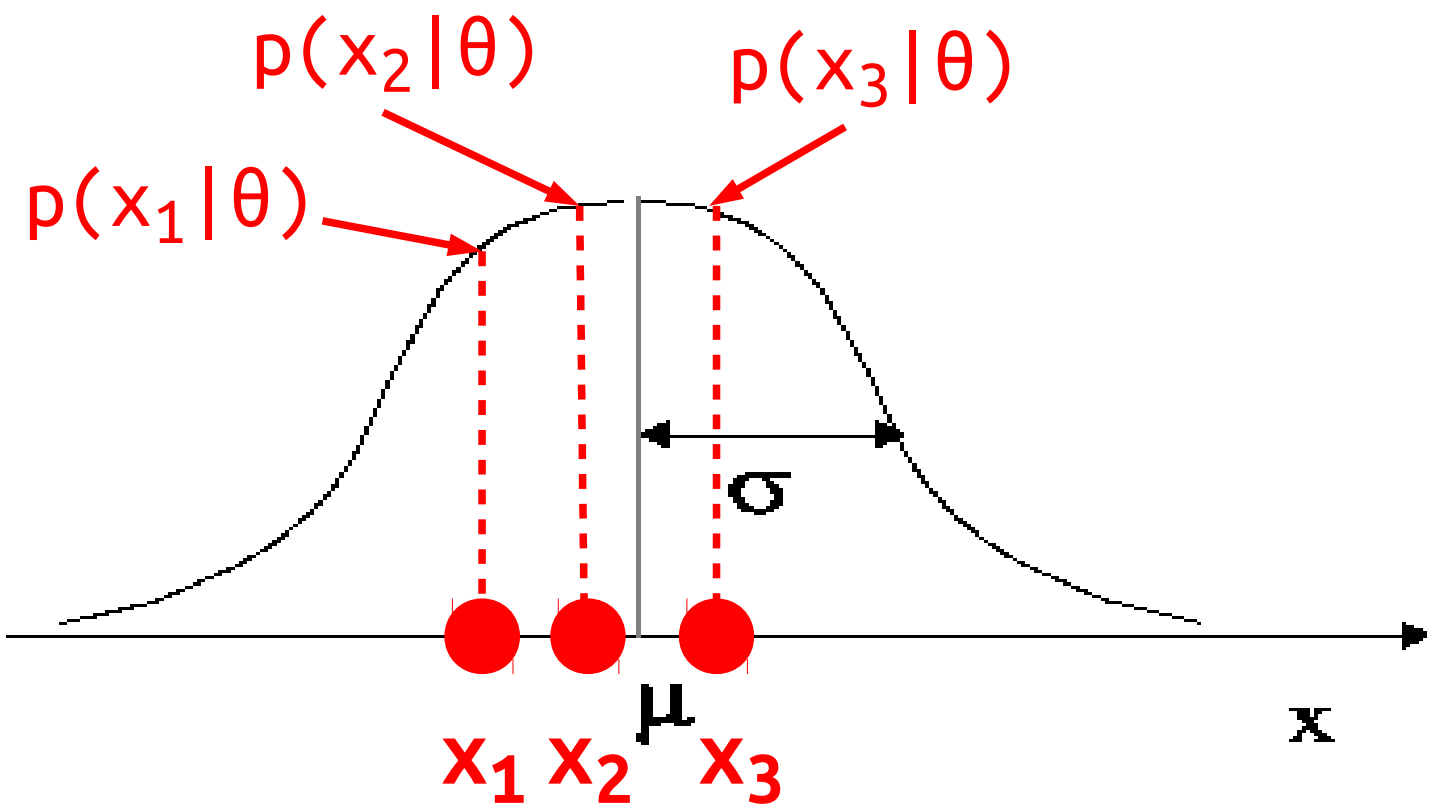
$$\theta = \{\mu, \sigma\}$$



$$P(\mathbf{D}|\theta) = p(x_1, x_2, x_3|\theta) = p(x_1|\theta)p(x_2|\theta)p(x_3|\theta)$$

x_1,x_2 and x_3 are **independent** and **identically distributed**
(i.e. they follow the same distribution, the Gaussian)

Note: if we change the parameters θ, also the likelihood changes!



$$P(\mathbf{D}|\theta) = p(x_1, x_2, x_3|\theta) = p(x_1|\theta)p(x_2|\theta)p(x_3|\theta)$$

In this case the likelihood is larger: the dataset is explained in a better way by the Gaussian identified by this θ!

Maximum Likelihood estimation

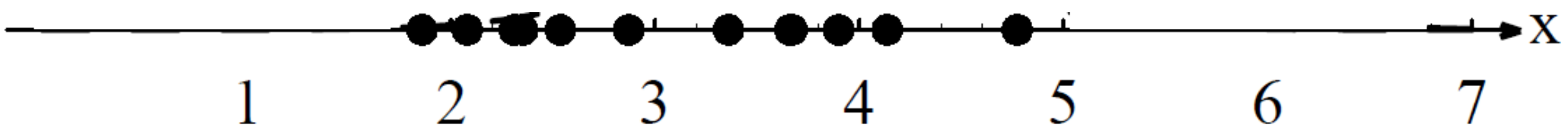
- Given a parameter θ, we can consider the likelihood P(**D**|θ) as a measure of "**how well** the training set is **explained** by the model defined by the parameter θ"
  - The likelihood P(**D**|θ) is a **function of θ** (**D** is fixed)

The Maximum Likelihood estimate of the parameter of the model is defined as the parameter θ_{ML} which **maximizes** the likelihood P(D|θ)

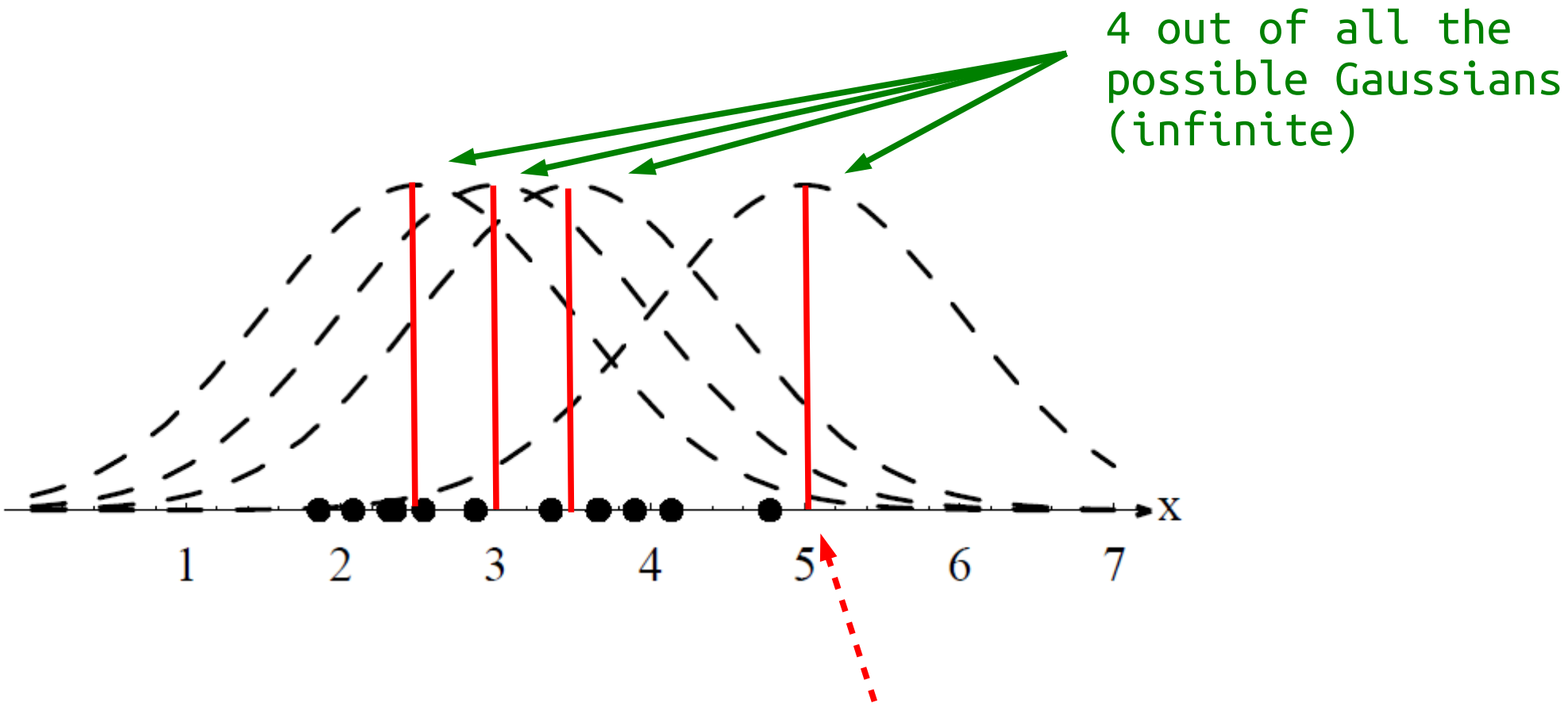
$$\theta_{ML} = \arg\max_{\theta} P(D|\theta)$$

Maximum Likelihood estimation

- **Example**: **D** contains some points to be modelled with a one-dimensional Gaussian
  - The variance is known, the only parameter is the mean ($\theta = \{\mu\}$)

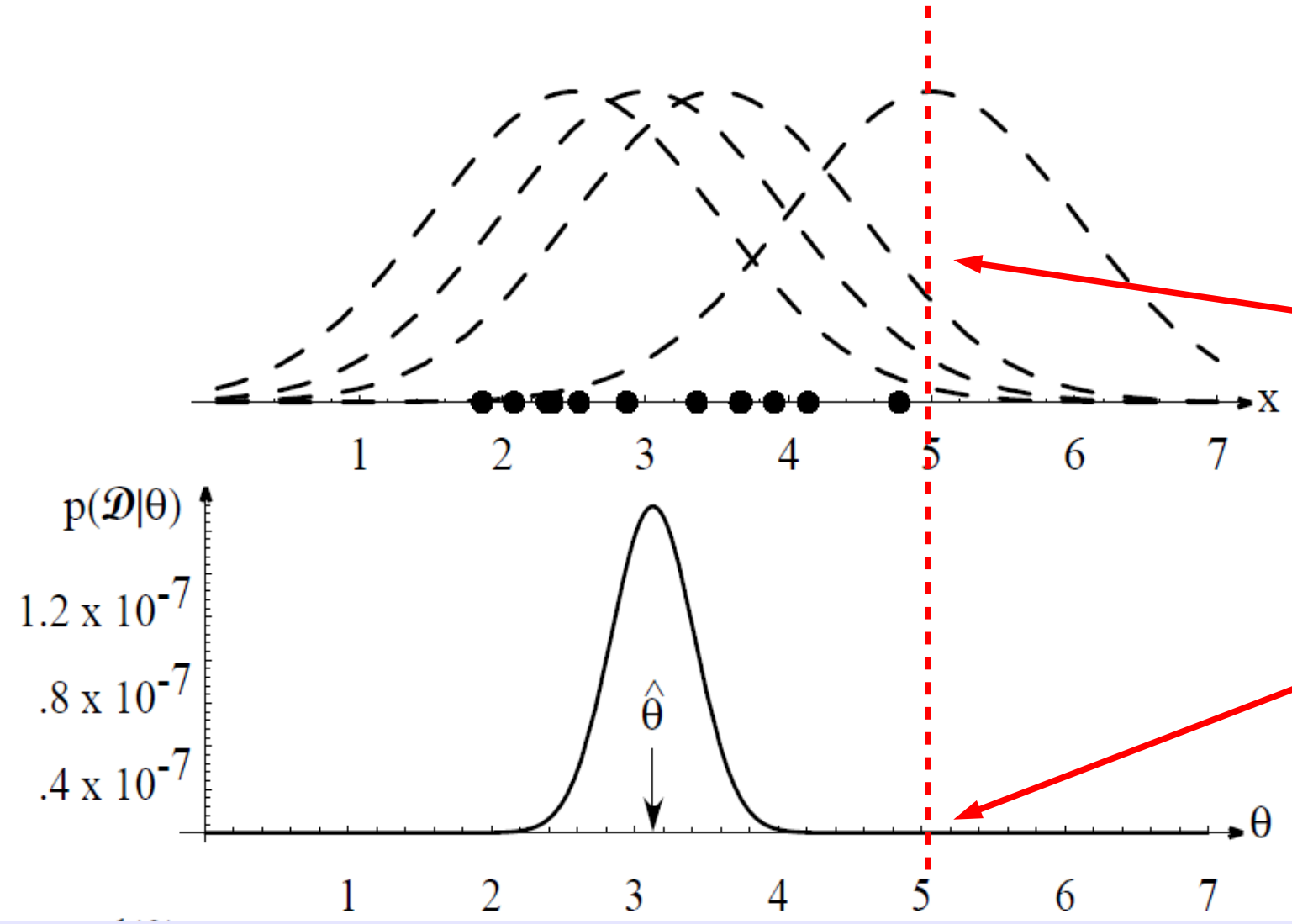


Maximum Likelihood estimation



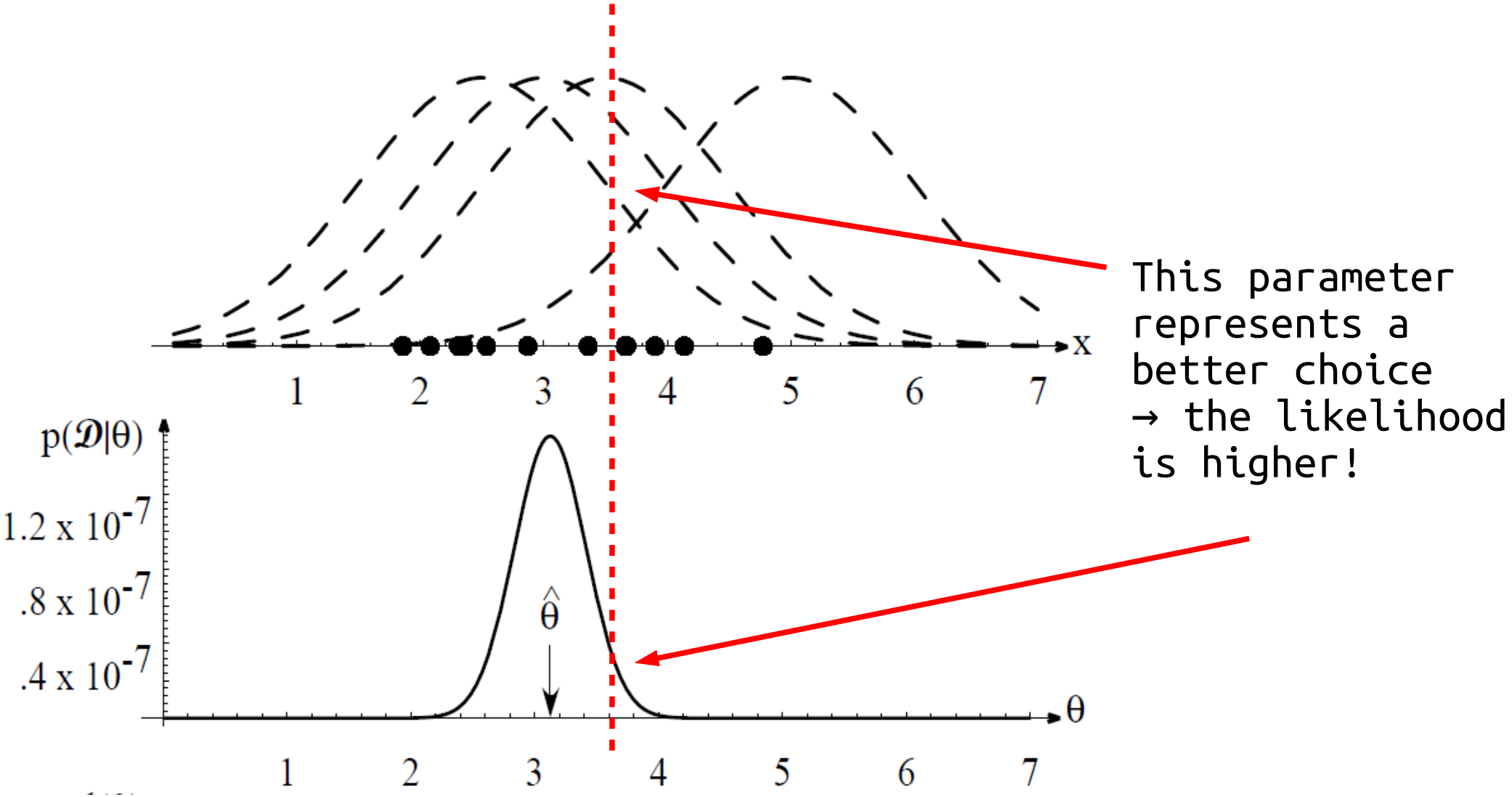
Maximum Likelihood estimation

Let's compute the likelihood, i.e. "**how well** the training set is **explained** by the model defined by a specific parameter θ"



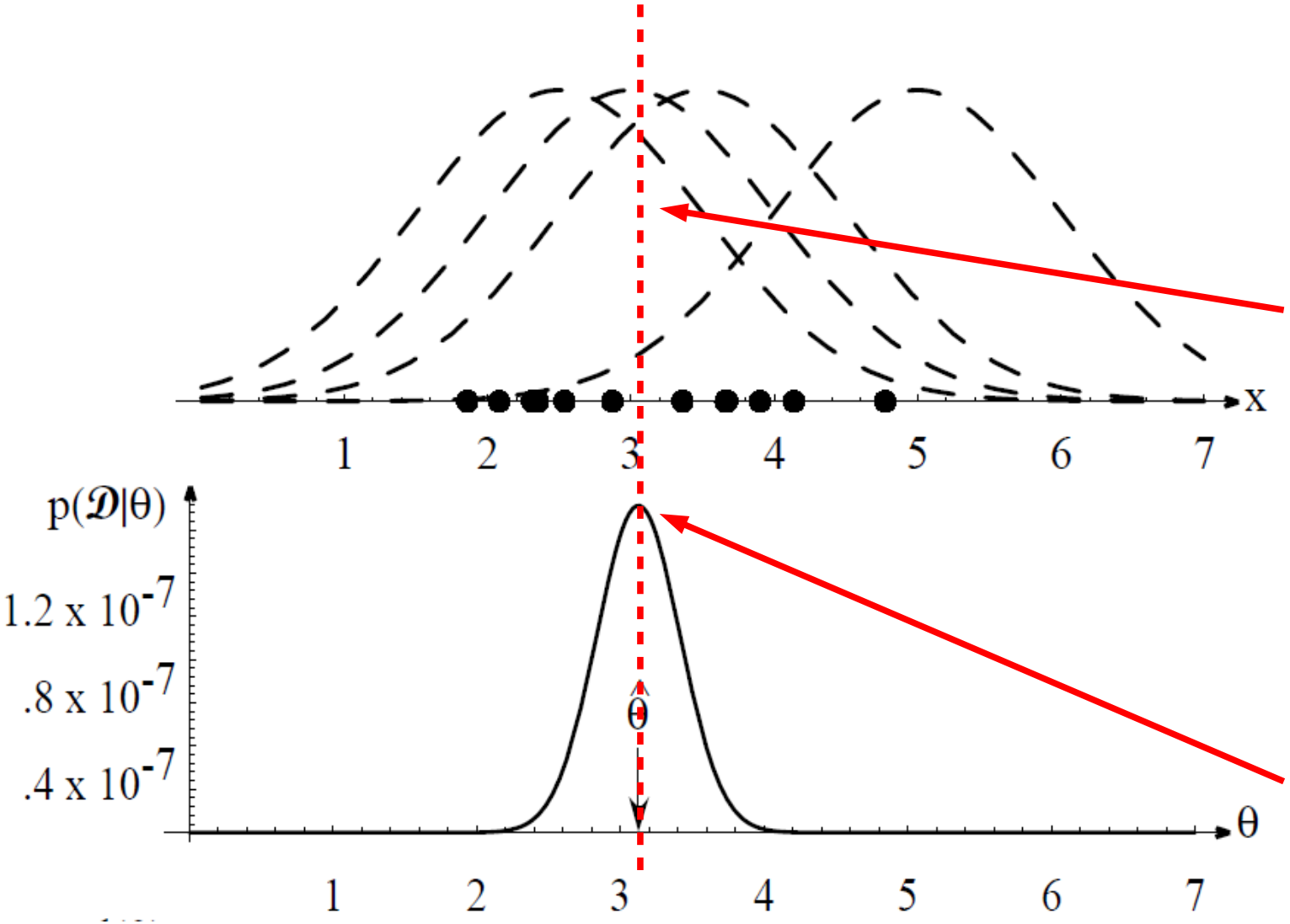
This parameter (which identifies a Gaussian) does not expain well the dataset
→ low likelihood

Maximum Likelihood estimation



This parameter represents a better choice → the likelihood is higher!

Maximum Likelihood estimation



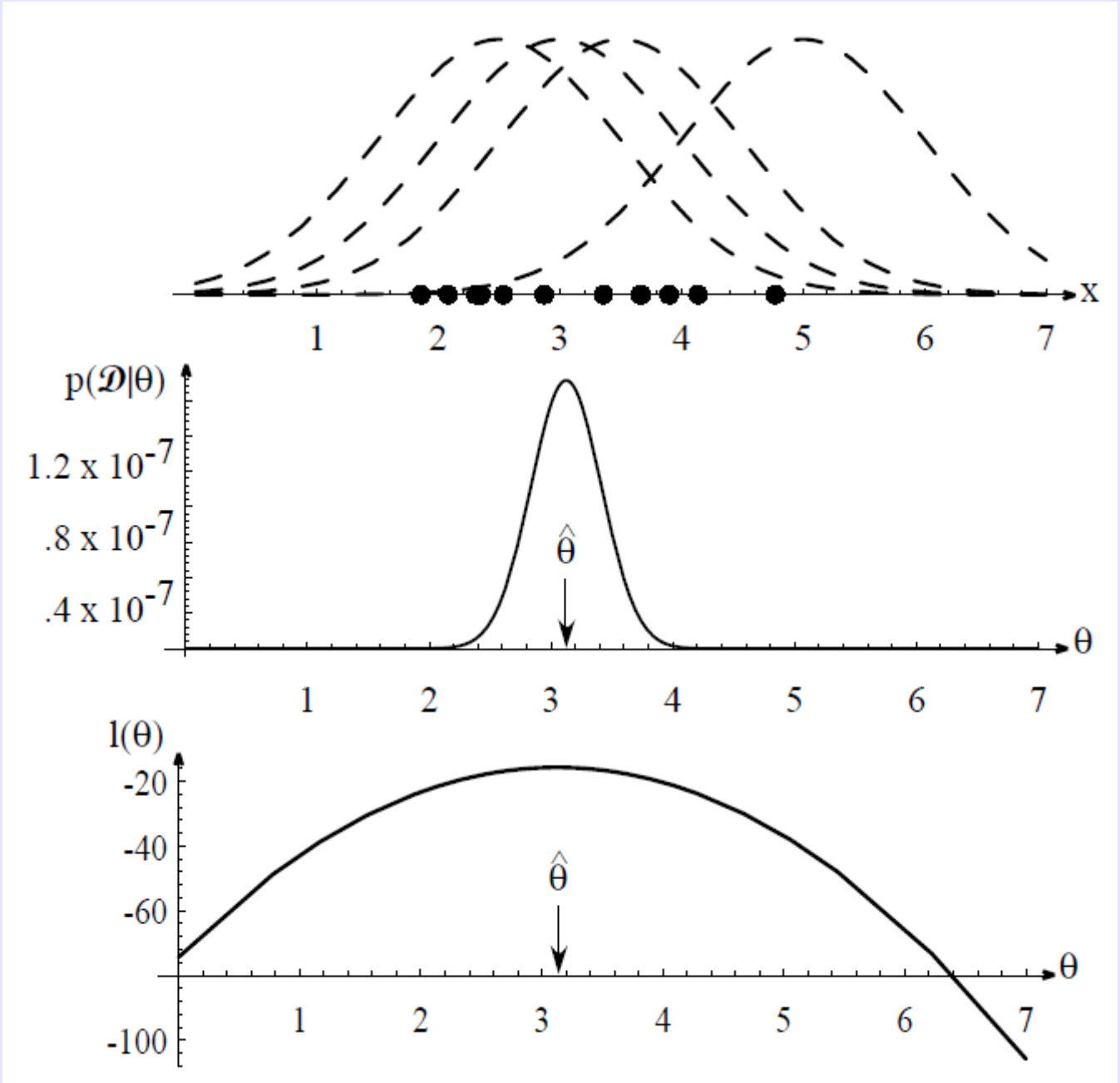
This parameter represents the best choice: the corresponding Gaussian is the one explaining in the best way the dataset
→ maximum of the likelihood!

Maximum Likelihood estimation

- Important note: the likelihood P(**D**|θ) is a **function of θ**, whereas the probability p(x|θ) is a **function of x** (p(x|θ) is the probability of a point given a model specified with the parameter θ)
- In order to maximize the likelihood, it is often mathematically convenient to work with the logarithm of the likelihood, often called **log-likelihood**

$$l(\theta) = \ln P(\mathbf{D}|\theta) = \ln \prod_{i=1}^{N} p(x_i|\theta) = \sum_{i=1}^{N} \ln P(x_i|\theta)$$

Maximum Likelihood estimation



The θ maximizing P(**D**|θ) also maximizes l(θ)

The logarithm is a monotonic function

This trick is useful when we have to compute derivatives to maximize the likelihood (the derivative of a sum is more manageable than the derivative of a product)

Maximum Likelihood estimation

- In many cases the ML estimate provides a **sufficiently good** value for the parameter
  - This is especially true when the training set is reasonably **large**

- One example: ML estimation when $p(x|\theta)$ is a Gaussian with known variance and unknown mean

  (board)

Bayesian estimation

- Alternative approach to estimate the p(x).

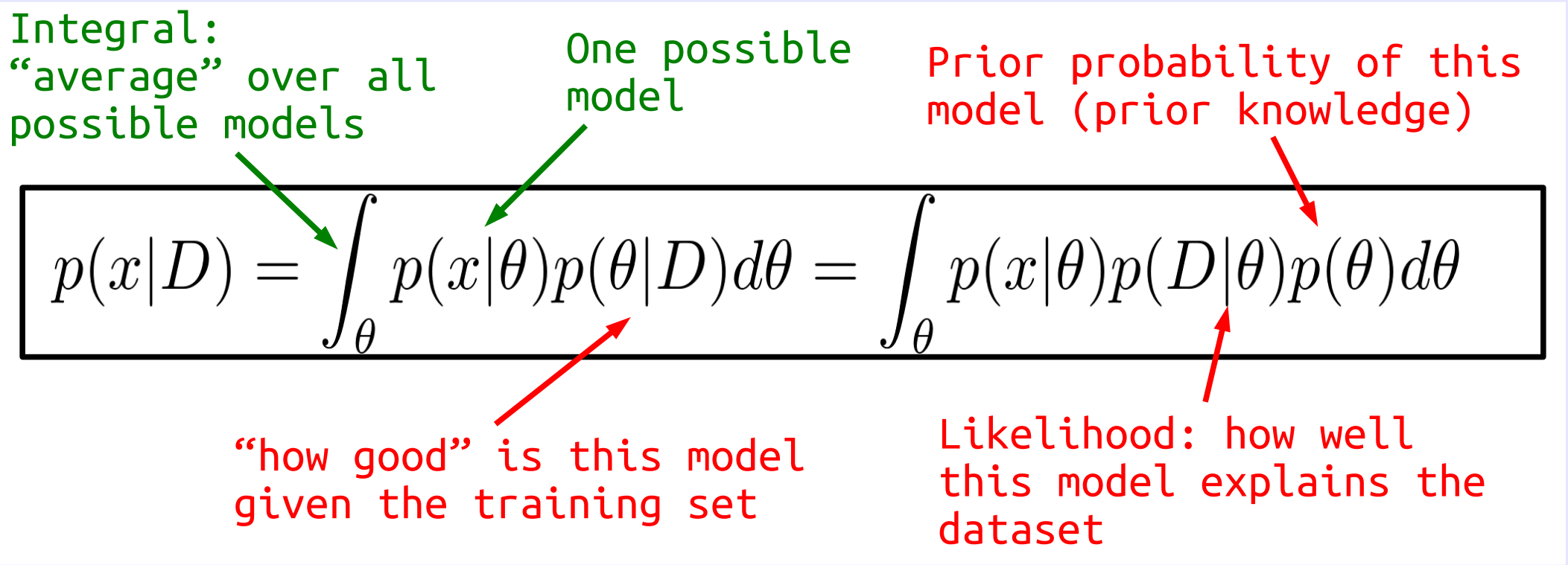
  In a nutshell:

  - Instead of considering that there exists a **single** optimal parameter θ which permits to explain the dataset, the Bayes estimate takes into account **all the possible values** of θ to define the p(x), each one with its own probability

Bayesian estimation

(board)

Bayesian estimation: summary



- Powerful approach, but difficult in practical applications
  - computing the integral may be impossible (intractable integral)
  - defining the priors p(θ) may be problematic