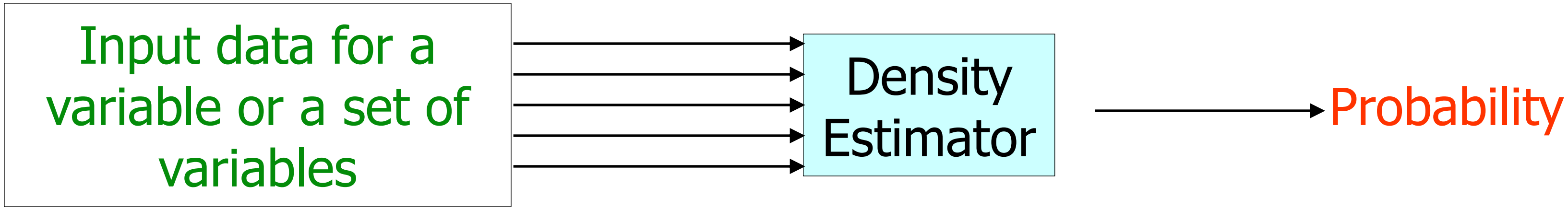
10-601
Machine Learning

Density estimation

Density Estimation

- A Density Estimator learns a mapping from a set of attributes to a Probability



Density estimation

- Estimate the distribution (or conditional distribution) of a random variable
- Types of variables:

  - Binary

    coin flip, alarm

  - Discrete

    dice, car model year

  - Continuous

    height, weight, temp.,

When do we need to estimate densities?

- Density estimators can do many good things…
  - Can sort the records by probability, and thus spot weird records (anomaly detection)
  - Can do inference: P(E1|E2)

    Medical diagnosis / Robot sensors
  - Ingredient for Bayes networks and other types of ML methods

Density estimation

- Binary and discrete variables:

Easy: Just count!

- Continuous variables:

Harder (but just a bit): Fit a model

Learning a density estimator for discrete variables

$$\hat{P}(x_i = u) = \frac{\#\text{records in which } x_i = u}{\text{total number of records}}$$

A trivial learning algorithm!

But why is this true?

Maximum Likelihood Principle

We can define the likelihood of the data given the model as follows:

$$\hat{P}(\text{dataset} \mid M) = \hat{P}(x_1 \wedge x_2 \cdots \wedge x_n \mid M) = \prod_{k=1}^{n} \hat{P}(x_k \mid M)$$

M is our model (usually a collection of parameters)

For example M is

- The probability of ‘head’ for a coin flip

- The probabilities of observing 1,2,3,4 and 5 for a dice

- etc.

Maximum Likelihood Principle

$$\hat{P}(\text{dataset} \mid M) = \hat{P}(x_1 \wedge x_2 \ldots \wedge x_n \mid M) = \prod_{k=1}^{n} \hat{P}(x_k \mid M)$$

- Our goal is to determine the values for the parameters in *M*
- We can do this by maximizing the probability of generating the observed samples
- For example, let Θ *be the probabilities for a coin flip*
- Then

$$L(x_1, \ldots, x_n \mid \Theta) = p(x_1 \mid \Theta) \ldots p(x_n \mid \Theta)$$

- The observations (different flips) are assumed to be independent
- For such a coin flip with *P(H)=q* the best assignment for Θ_h is

$$argmax_q = \#H/\#samples$$

- Why?

Maximum Likelihood Principle: Binary variables

- For a binary random variable A with P(A=1)=q
  argmax_q = #1/#samples
- Why?

Data likelihood: $P(D \mid M) = q^{n_1}(1-q)^{n_2}$

We would like to find: $\arg\max_q q^{n_1}(1-q)^{n_2}$

Maximum Likelihood Principle

Data likelihood: $P(D|M) = q^{n_1}(1-q)^{n_2}$

We would like to find: $\arg\max_q q^{n_1}(1-q)^{n_2}$

$$\frac{\partial}{\partial q} q^{n_1}(1-q)^{n_2} = n_1 q^{n_1-1}(1-q)^{n_2} - q^{n_1} n_2 (1-q)^{n_2-1}$$

$$\frac{\partial}{\partial q} = 0 \Rightarrow$$

$$n_1 q^{n_1-1}(1-q)^{n_2} - q^{n_1} n_2 (1-q)^{n_2-1} = 0 \Rightarrow$$

$$q^{n_1-1}(1-q)^{n_2-1}(n_1(1-q) - qn_2) = 0 \Rightarrow$$

$$n_1(1-q) - qn_2 = 0 \Rightarrow$$

$$n_1 = n_1 q + n_2 q \Rightarrow$$

$$q = \frac{n_1}{n_1 + n_2}$$

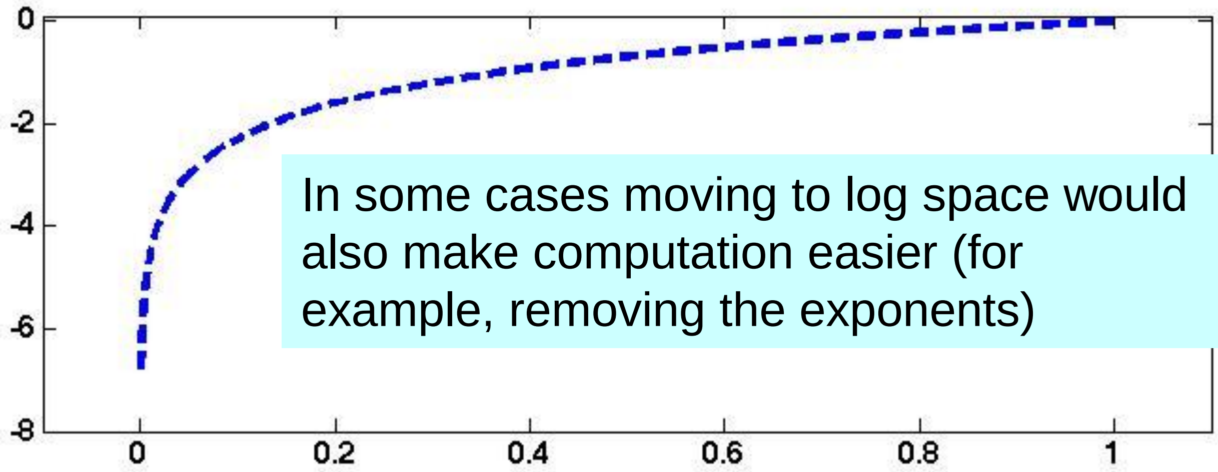
Log Probabilities

When working with products, probabilities of entire datasets often get too small. A possible solution is to use the log of probabilities, often termed 'log likelihood'

$$\log \hat{P}(\text{dataset} \mid M) = \log \prod_{k=1}^{n} \hat{P}(x_k \mid M) = \sum_{k=1}^{n} \log \hat{P}(x_k \mid M)$$

Maximizing this likelihood function is the same as maximizing P(dataset | M)

Log values between 0 and 1



In some cases moving to log space would also make computation easier (for example, removing the exponents)

Density estimation

- Binary and discrete variables:

Easy: Just count!

- Continuous variables:

Harder (but just a bit): Fit a model

But what if we only have very few samples?

The danger of joint density estimation

P(summer & size > 20 & evaluation = 3) = 0

- No such example in our dataset

Summer?	Size	Evaluation
1	19	3
1	17	3
0	49	2
0	33	1
0	55	3
1	20	1

Now lets assume we are given a new (often called 'test') dataset. If this dataset contains the line

Summer	Size	Evaluation
1	30	3

Then the probability we would assign to the *entire* dataset is 0

Naïve Density Estimation

The problem with the Joint Estimator is that it just mirrors the training data.

We need something which generalizes more usefully.

The naïve model generalizes strongly:

Assume that each attribute is distributed independently of any of the other attributes.

If two variables are independent then
p(A,B) = p(A)p(B)

Joint estimation, revisited

Assuming independence we can compute each probability independently

P(Summer) = 0.5

P(Evaluation = 1) = 0.33

P(Size > 20) = 0.66

Not bad !

How do we do on the joint?

P(Summer & Evaluation = 1) = 0.16

P(Summer)P(Evaluation = 1) = 0.16

P(size > 20 & Evaluation = 1) = 0.33
P(size > 20)P(Evaluation = 1) = 0.22

Summer?	Size	Evaluation
1	19	3
1	17	3
0	49	2
0	33	1
0	55	2
1	21	1

Joint estimation, revisited

Assuming independence we can compute each probability independently

P(Summer) = 0.5

P(Evaluation = 3) = 0.33

P(Size > 20) = 0.66

How do we do on the joint?

P(Summer & Eval = 3) = 0.33

P(Summer)P(Eval = 3) = 0.16

Summer?	Size	Evaluation
1	19	3
1	17	3
0	49	2
0	33	1
0	55	2
1	21	1

We must be careful when using the Naïve density estimator

Contrast

Joint DE	Naïve DE
Can model anything	Can model only very boring distributions
No problem to model “C is a noisy copy of A”	Outside Naïve’s scope
Given 100 records and more than 6 Boolean attributes will screw up badly	Given 100 records and 10,000 multivalued attributes will be fine

Dealing with small datasets

- We just discussed one possibility: Naïve estimation
- There is another way to deal with small number of measurements that is often used in practice.
- Assume we want to compute the probability of heads in a coin flip

  - What if we can only observe 3 flips?

  - 25% of the times a maximum likelihood estimator will assign probability of 1 to either the heads or tails

Pseudo counts

- What if we can only observe 3 flips?
- 25% of the times a maximum likelihood estimator will assign probability of 1 to either the heads or tails

- In these cases we can use prior belief about the 'fairness' of most coins to influence the resulting model.
- We assume that we have observed 10 flips with 5 tails and 5 heads
- Thus p(heads) = (#heads+5)/(#flips+10)
- Advantages: 1. Never assign a probability of 0 to an event

  2. As more data accumulates we can get very close to the real distribution (the impact of the pseudo counts will diminish rapidly)

Pseudo counts

- What if we can only observe 3 flips?
- 25% of the times a maximum likelihood estimator will assign probability of 1 to either the he

- In thes
  'fairnes ... nodel.
- We as ... 5 tails and 5 h
- Thus p
- Advan ... nt

  2. As m ... real distribution (the impact of the pseudo counts will diminish rapidly)

Some distributions (for example, the Beta distribution) can incorporate pseudo counts as part of the model

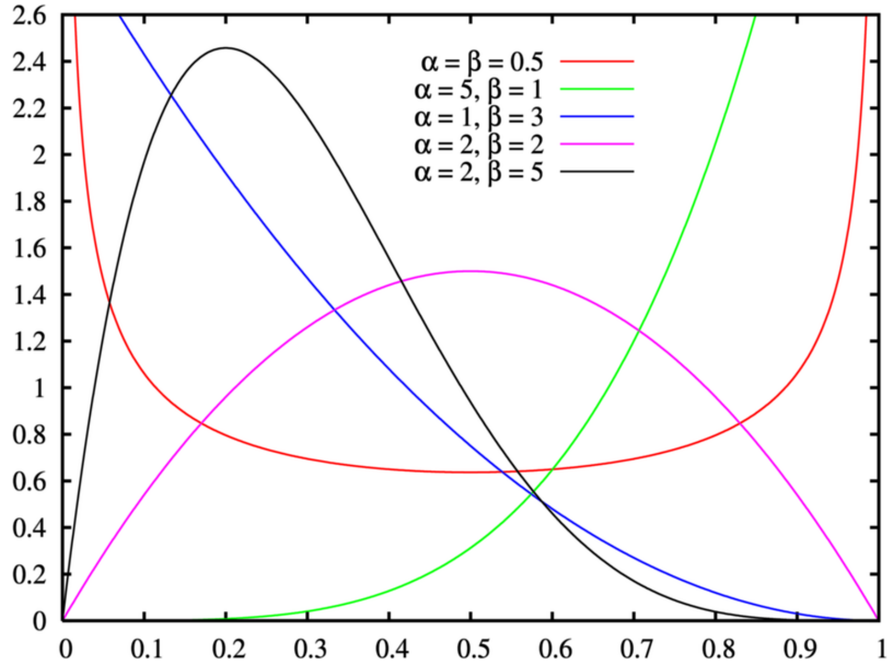
Beta distribution

- The beta distribution provides an easy way to incorporate prior knowledge in the form of pseudo-counts

$$p(\Theta;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\Theta^{\alpha-1}(1-\Theta)^{\beta-1}$$

- Where Γ is defined (for discrete values of x):

  $\Gamma(x+1) = x\Gamma(x) = x!$



Beta distribution

$$p(\Theta;\alpha,\beta)=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\Theta^{\alpha-1}(1-\Theta)^{\beta-1}$$

Assume we observed n coin flips of which n1 are heads and n2 are tails then the likelihood of Θ is:

$$P(\Theta\,|\,x_1\ldots x_n)=\frac{P(x_1\ldots x_n\,|\,\Theta)P(\Theta)}{P(x_1\ldots x_n)}\propto\Theta^{n1}(1-\Theta)^{n2}\Theta^{\alpha-1}(1-\Theta)^{\beta-1}$$

$$=\Theta^{n1+\alpha-1}(1-\Theta)^{n2+\beta-1}=P(\Theta;\alpha+n1,\beta+n2)$$

-Note the similarity of the posterior to the prior

- Such priors are termed conjugate priors

- α and β are termed hyperparameters (parameters of the prior) and correspond to the number of pseudo counts from each class

Density estimation

- Binary and discrete variables:

  Easy: Just count! √

- Continuous variables:

  Harder (but just a bit): Fit a model

How much do grad students sleep?

- Lets try to estimate the distribution of the time students spend sleeping (outside class).

Possible statistics

- **X**

  Sleep time

- **Mean of X:**

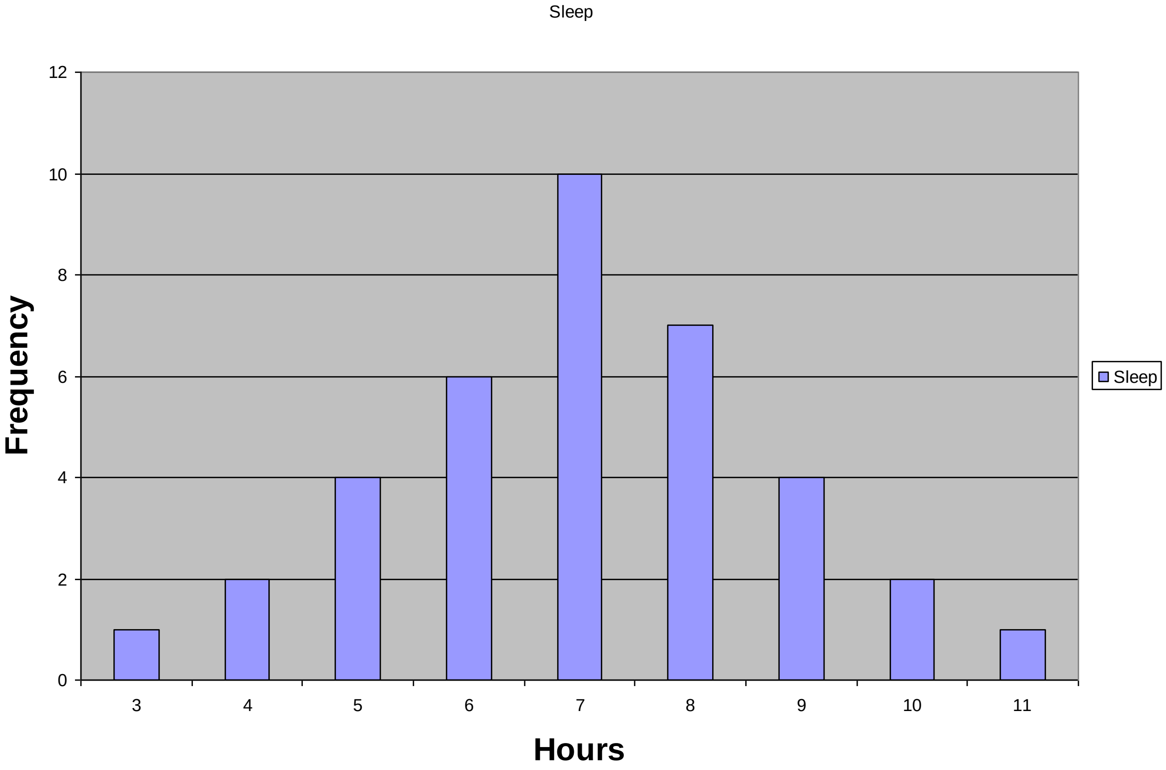
  E{X}

  7.03

- **Variance of X:**

  Var{X} = E{(X-E{X})^2}

  3.05

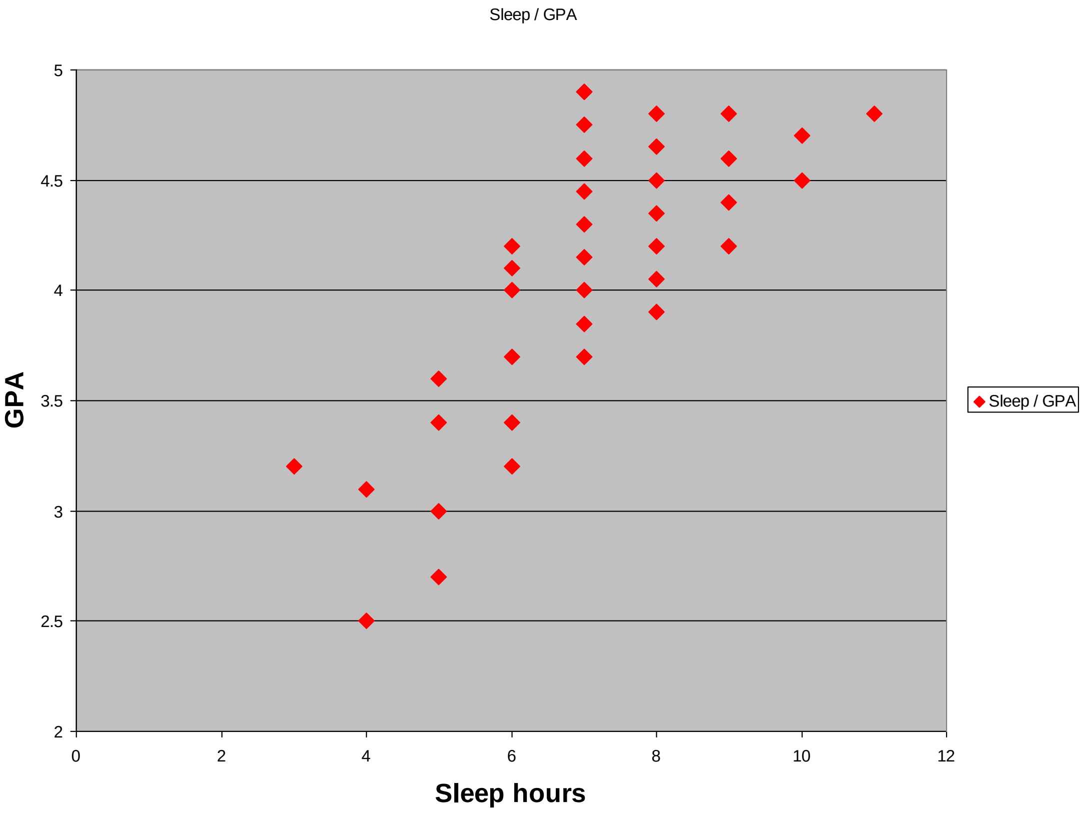


Covariance: Sleep vs. GPA

- **Co-Variance of X1, X2:**

  Covariance{X1,X2} = E{(X1-E{X1})(X2-E{X2})}

  = 0.88



Statistical Models

- Statistical models attempt to characterize properties of the population of interest

- For example, we might believe that repeated measurements follow a normal (Gaussian) distribution with some mean μ and variance σ^2 , x ~ N(μ,σ^2)

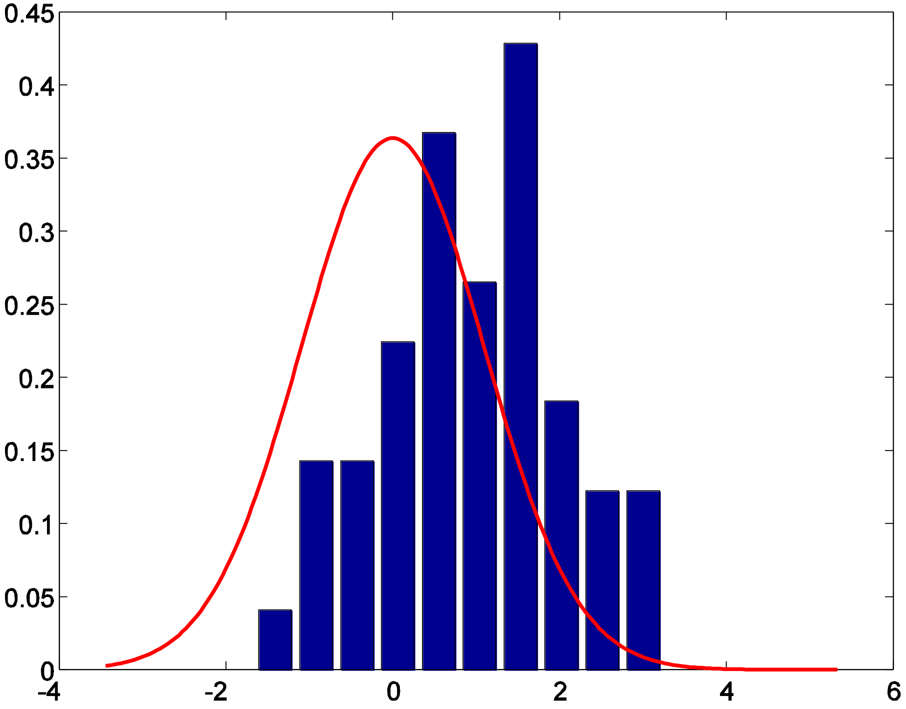
where

$$p(x \mid \Theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

and $\Theta=(\mu,\sigma^2)$ defines the parameters (mean and variance) of the model.

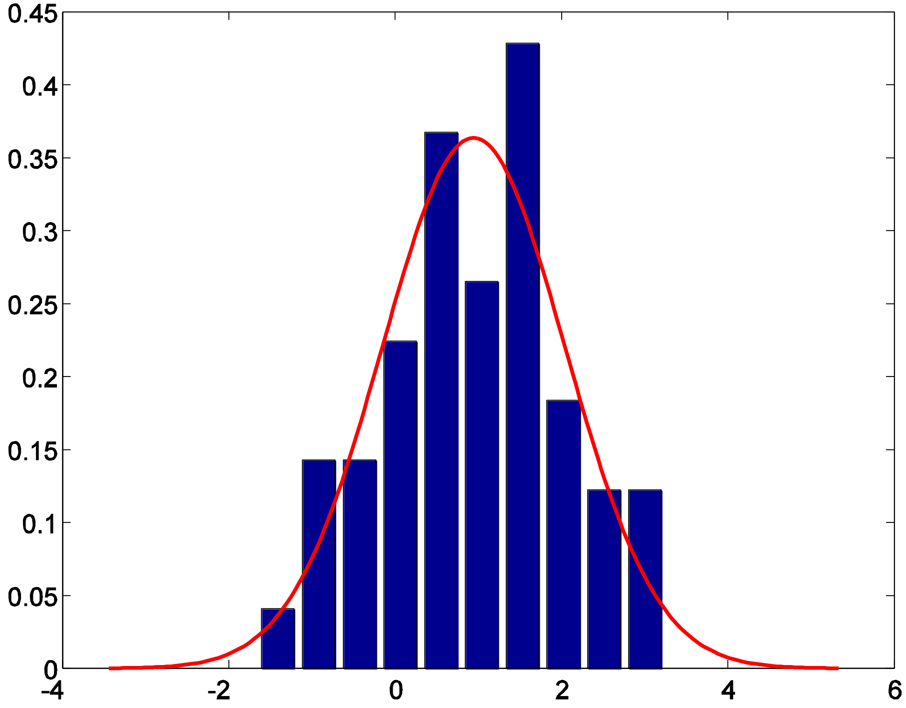
The Parameters of Our Model

- A statistical model is a **collection** of distributions; the **parameters** specify individual distributions x ~ N(μ,σ^2)
- We need to adjust the parameters so that the resulting distribution **fits** the data well



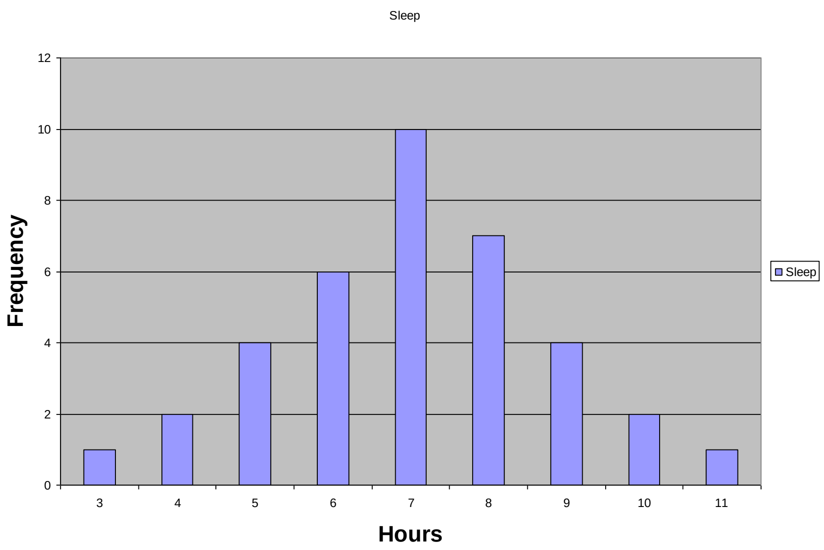
The Parameters of Our Model

- A statistical model is a **collection** of distributions; the **parameters** specify individual distributions $x \sim N(\mu, \sigma^2)$
- We need to adjust the parameters so that the resulting distribution **fits** the data well



Computing the parameters of our model

- Lets assume a Guassian distribution for our sleep data
- How do we compute the parameters of the model?



Maximum Likelihood Principle

• We can fit statistical models by maximizing the probability of generating the observed samples:
$L(x_1, \ldots, x_n \mid \Theta) = p(x_1 \mid \Theta) \ldots p(x_n \mid \Theta)$
(the samples are assumed to be independent)

• In the Gaussian case we simply set the mean and the variance to the sample mean and the sample variance:

$$\overline{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \overline{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{\mu})^2$$

Why?

Important points

- Maximum likelihood estimations (MLE)
- Pseudo counts
- Types of distributions
- Handling continuous variables