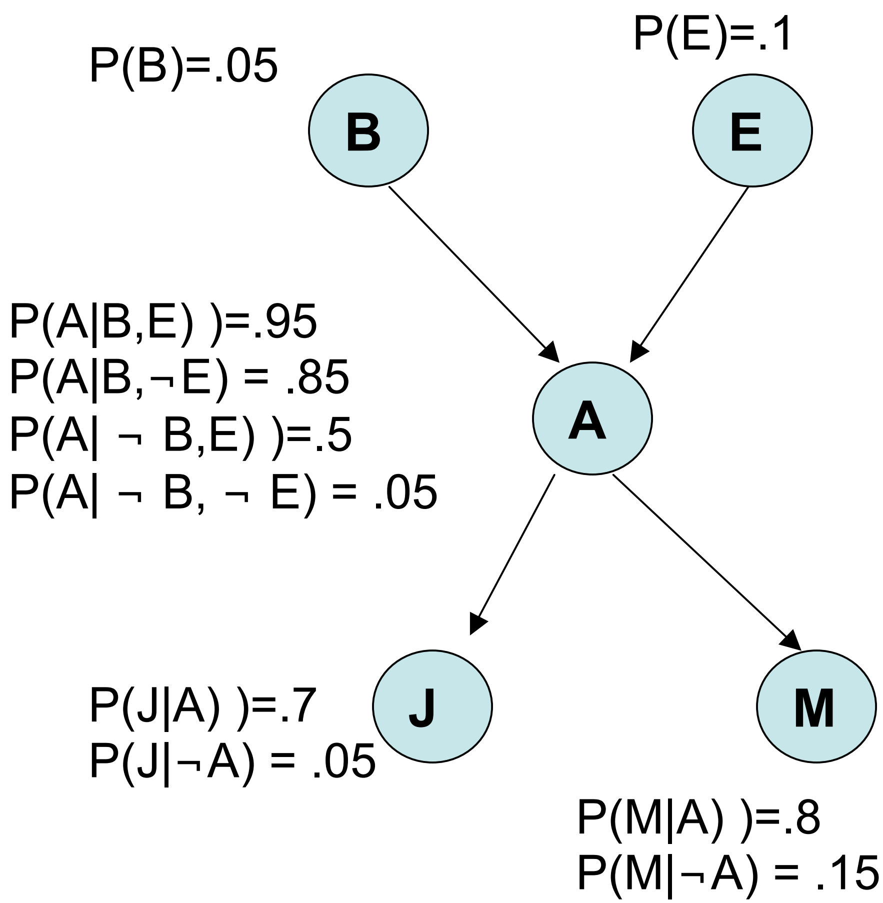
15-780: Graduate Artificial Intelligence

Density estimation

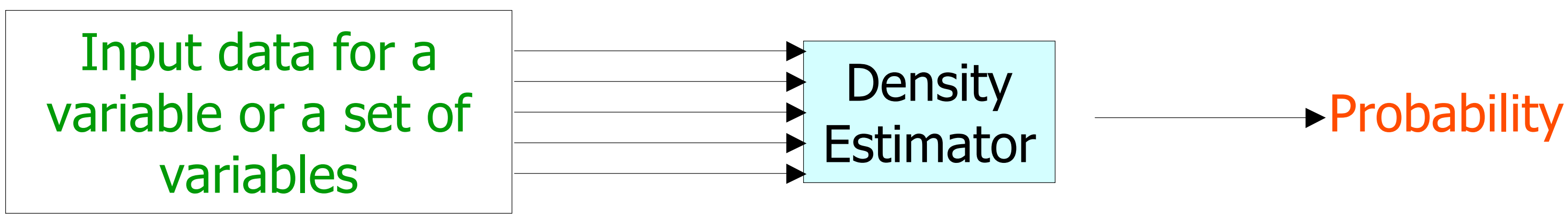
Conditional Probability Tables (CPT)

But where do we get them?



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.,

Not just for Bayesian networks …

- 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

Density estimation

- Binary and discrete variables:

Easy: Just count!

- Continuous variables:

Harder (but just a bit): Fit a model

Learning a density estimator

$$\hat{P}(x[i]=u) = \frac{\#\text{records in which } x[i]=u}{\text{total number of records}}$$

A trivial learning algorithm!

Course evaluation

P(summer) = #Summer / # records

= 23/151 = 0.15

P(Evaluation = 1) = #Evaluation=1 / # records
= 49/151 = 0.32

P(Evaluation = 1 | summer) = P(Evaluation = 1 & summer) / P(summer) = 2/23 = 0.09

But why do we count?

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

⋮

Computing the joint likelihood of the data

P(summer) = #Summer / # records

= 23/151 = 0.15

Summer?	Size	Evaluation

$$\hat{P}(\text{dataset}|M) = \hat{P}(\mathbf{x}_1 \wedge \mathbf{x}_2 \ldots \wedge \mathbf{x}_R|M) = \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M)$$

The next slide presents one of the most important ideas in probabilistic inference. It has a huge number of applications in many different and diverse problems

Maximum Likelihood Principle

- We can fit 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 (rows in the table) are assumed to be independent)
- For a binary random variable A with P(A=1)=q

  argmax_q = #1/#samples
- Why?

Maximum Likelihood Principle

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

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

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

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

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

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

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

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

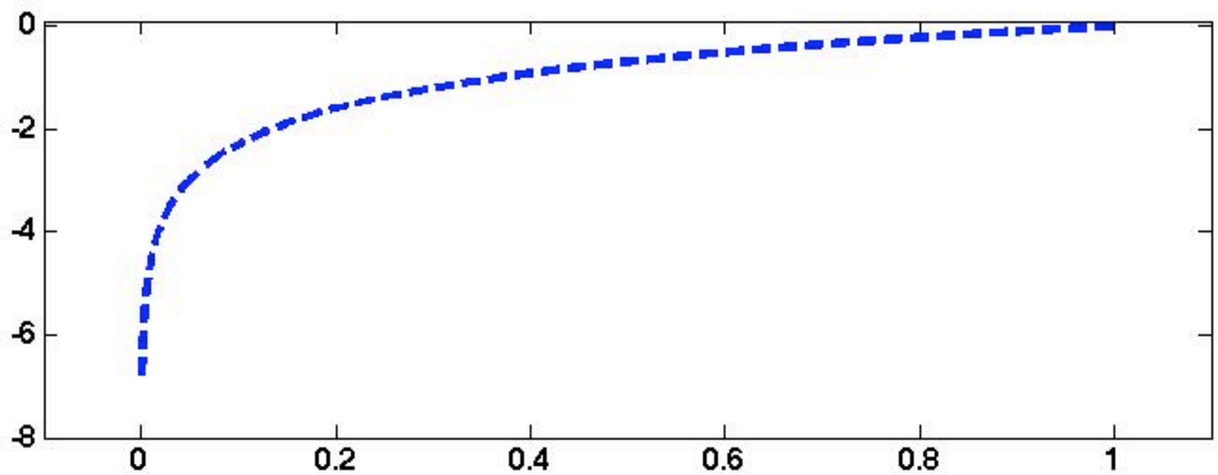
$$q = \frac{n_1}{n_1 + n_2}$$

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'

$$\log \hat{P}(\text{dataset}|M) = \log \prod_{k=1}^{R} \hat{P}(\mathbf{x}_k|M) = \sum_{k=1}^{R} \log \hat{P}(\mathbf{x}_k|M)$$

Log values between 0 and 1



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.

Joint estimation, revisited

Assuming independence we can compute each probability independently

P(Summer) = 0.15

P(Evaluation = 1) = 0.32

P(Size > 20) = 0.63

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

How do we do on the joint?

P(Summer & Evaluation = 1) = 0.09

P(Summer)P(Evaluation = 1) = 0.05

Not bad !

P(size > 20 & Evaluation = 1) = 0.23
P(size > 20)P(Evaluation = 1) = 0.20

Joint estimation, revisited

Assuming independence we can compute each probability independently

P(Summer) = 0.15

P(Evaluation = 1) = 0.32

P(Size > 20) = 0.63

How do we do on the joint?

P(Summer & Size > 20) = 0.026

P(Summer)P(Size > 20) = 0.094

Summer?	Size	Evaluation
1	19	3
1	17	3
0	49	2
0	33	1
0	55	3
1	20	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 … model.
- We as … 5 tails and 5 h
- Thus p
- Advan … ent

  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

Density estimation

- Binary and discrete variables:

  Easy: Just count! √

- Continuous variables:

  Harder (but just a bit): Fit a model

Conditional Probability Tables (CPT)

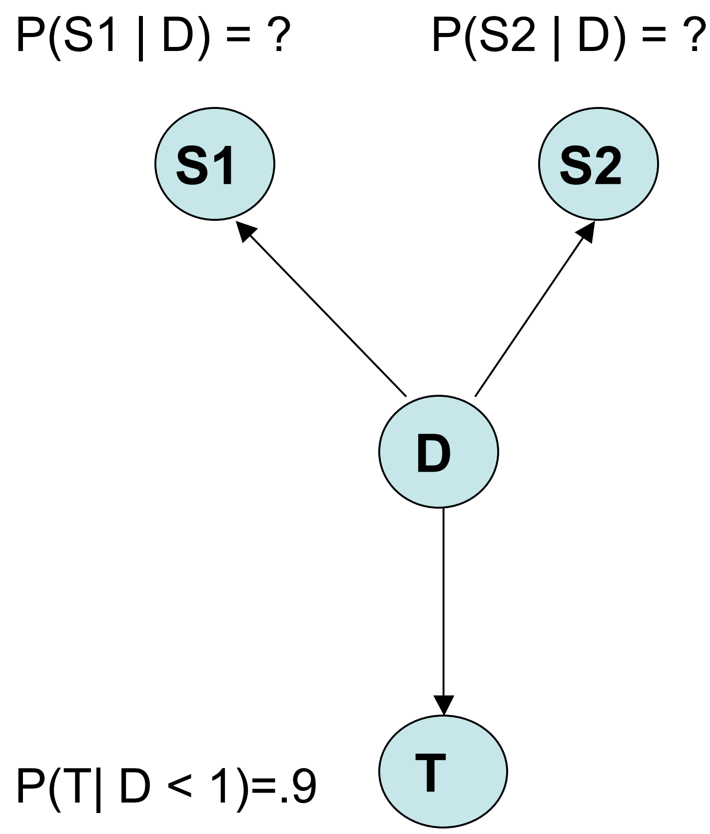
What do we do with continuous variables?

S1 – sensor 1

S2 – sensor 2

D – distance to wall

T – too close



Conditional Probability Tables (CPT)

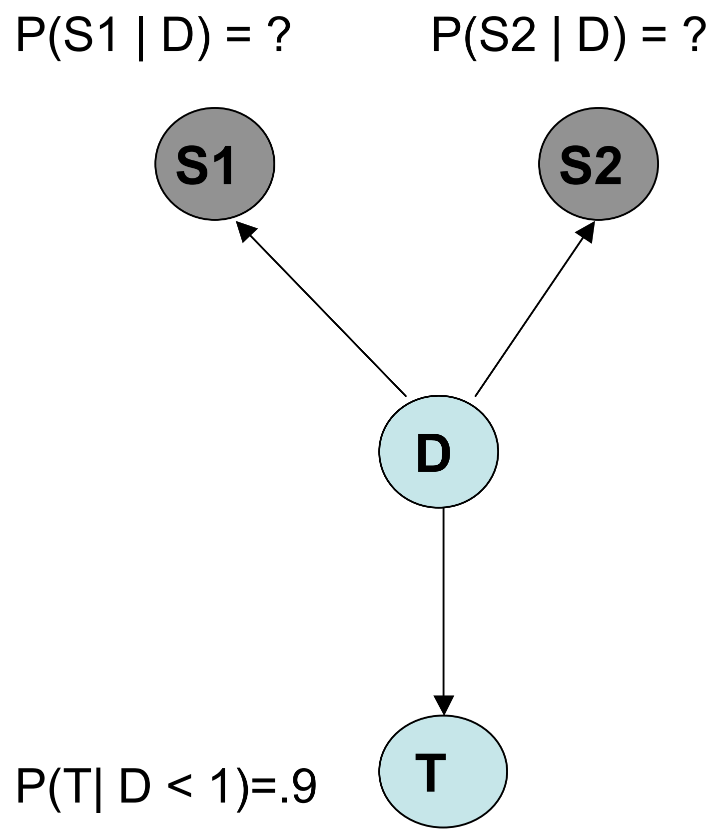
What do we do with continuous variables?

S1 – sensor 1

S2 – sensor 2

D – distance to wall

T – too close



Elementary Concepts

- **Population:** the ideal group whose properties we are interested in and from which the samples are drawn

  e.g., graduate students at CMU

- **Random sample:** a set of elements drawn at random from the population

  e.g., students in grad AI

Elementary Concepts

- **Statistic:** a number computed from the data

  e.g., Average time of sleep

Sample Statistics

- **Sample mean:** $$\overline{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

where n is the number of samples.

- **Sample variance:**

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

- **Sample covariance:**

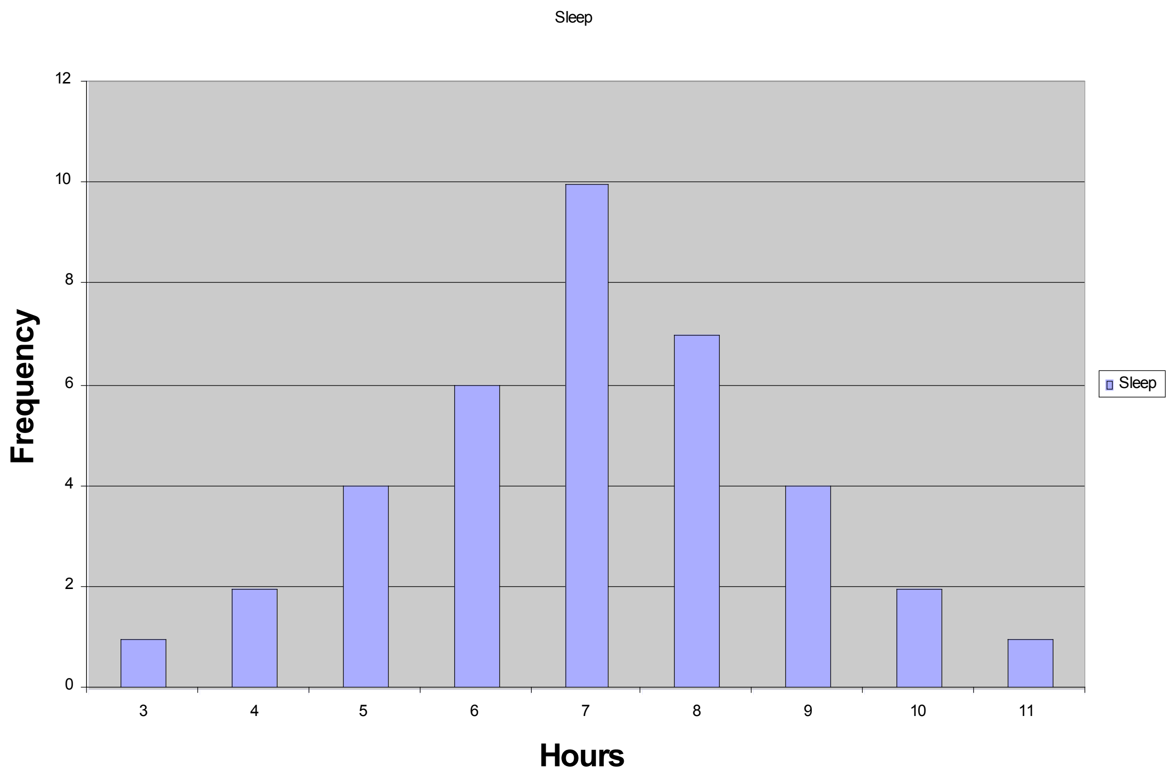
$$\overline{\mathrm{cov}(x_1, x_2)} = \frac{1}{n}\sum_{i=1}^{n}\left(x_{1,i} - \overline{\mu_1}\right)\left(x_{2,i} - \overline{\mu_2}\right)$$

How much do grad students sleep?

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

Possible statistics

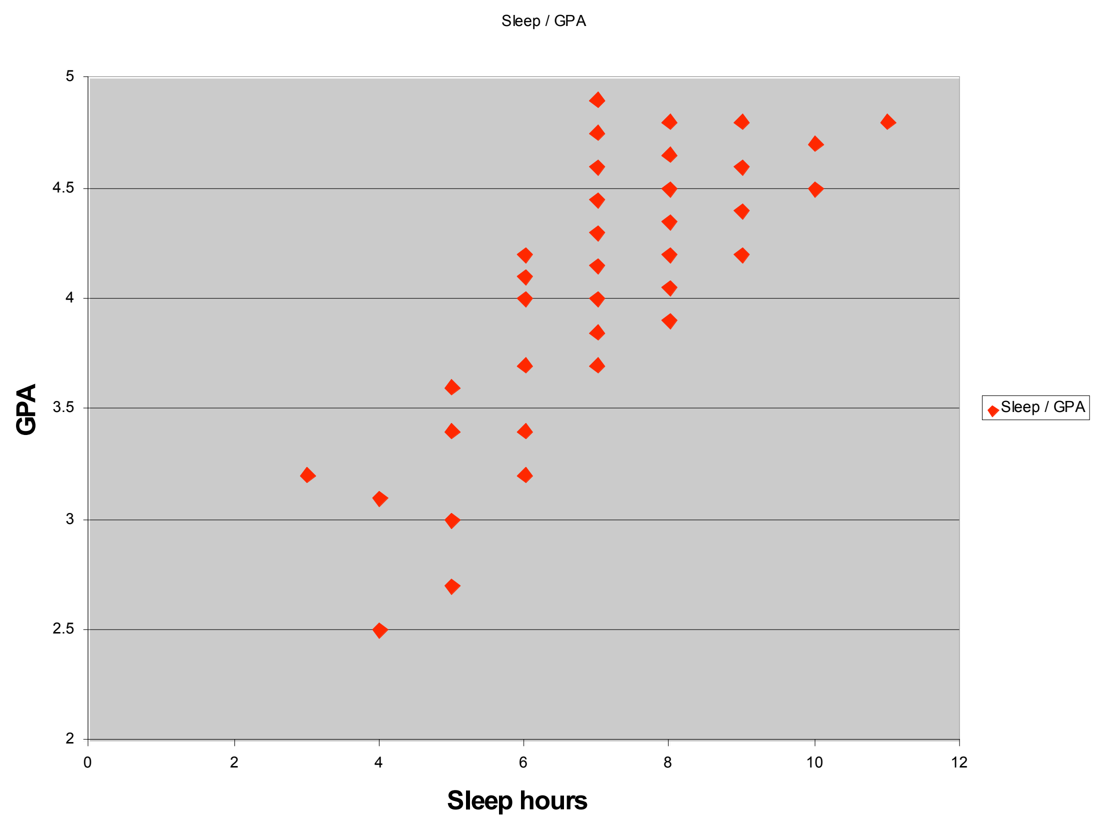
- **X**
  Sleep time
- **Mean of X:**
  E{X}
  7.03
- **Variance of X:**
  Var{X} = E{(X-E{X})^2}
  3.05



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)

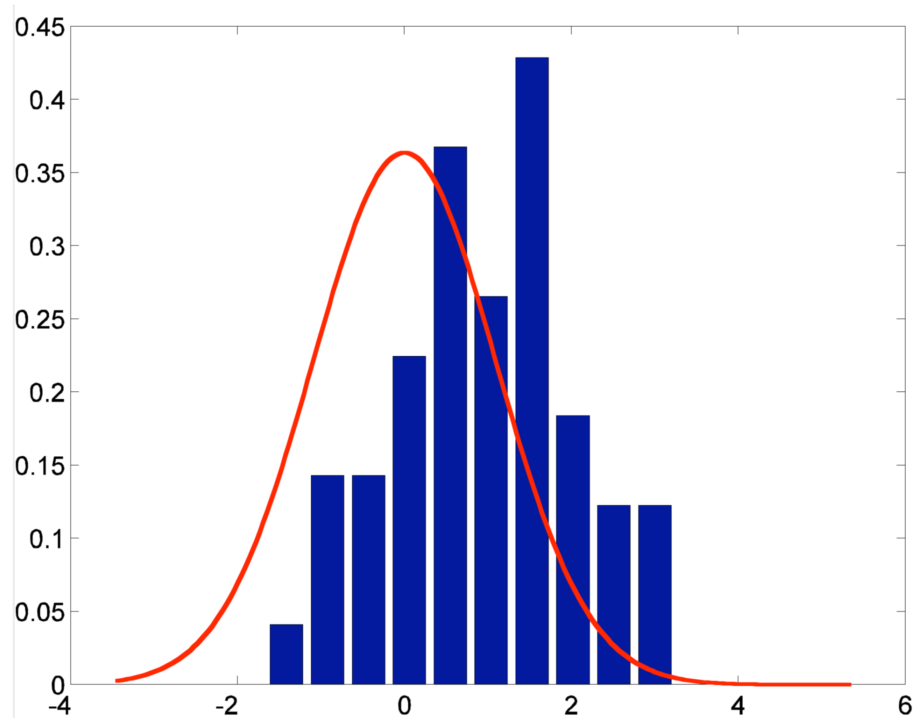
where

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

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

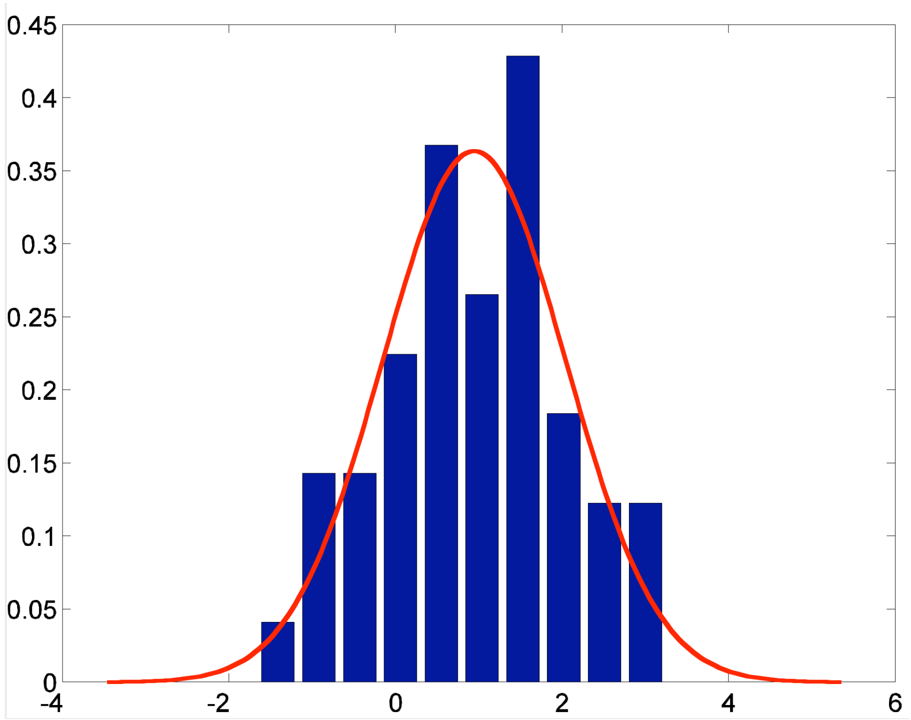
The Parameters of Our Model

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



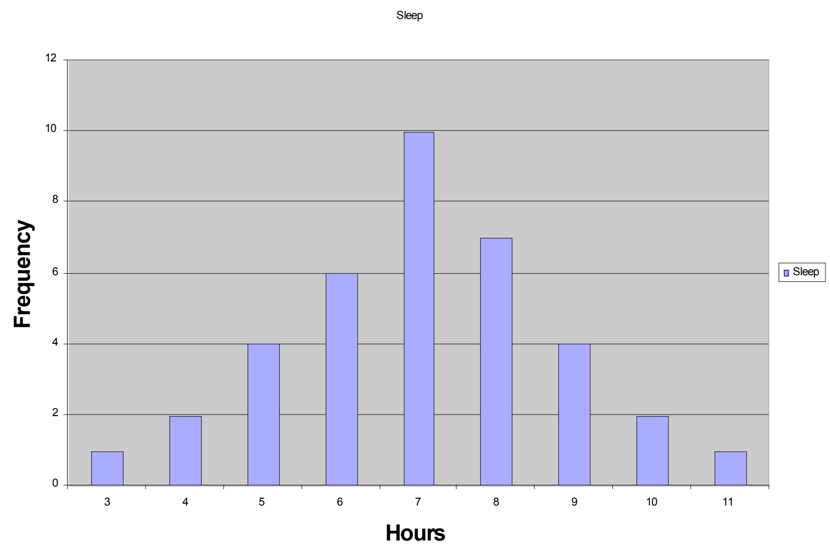
The Parameters of Our Model

- A statistical model is a **collection** of distributions; the **parameters** specify individual distributions x ~ N(μ,σ^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)

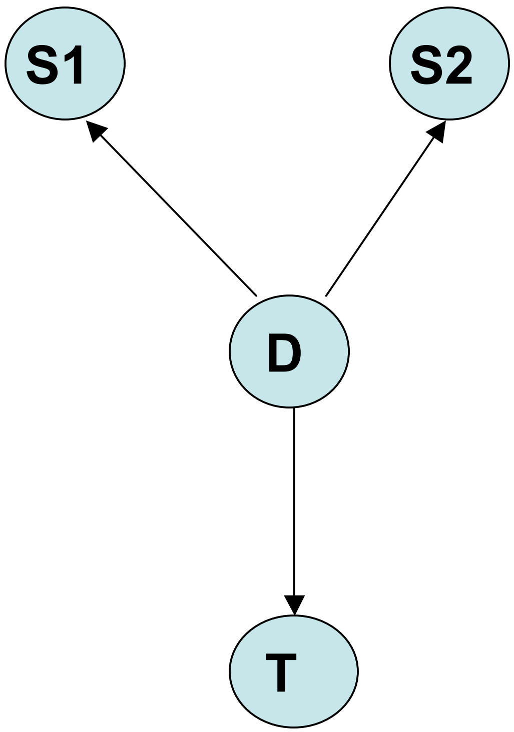
- In the Gaussian case we simply set the mean and the variance to the sample mean and the sample variance:

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

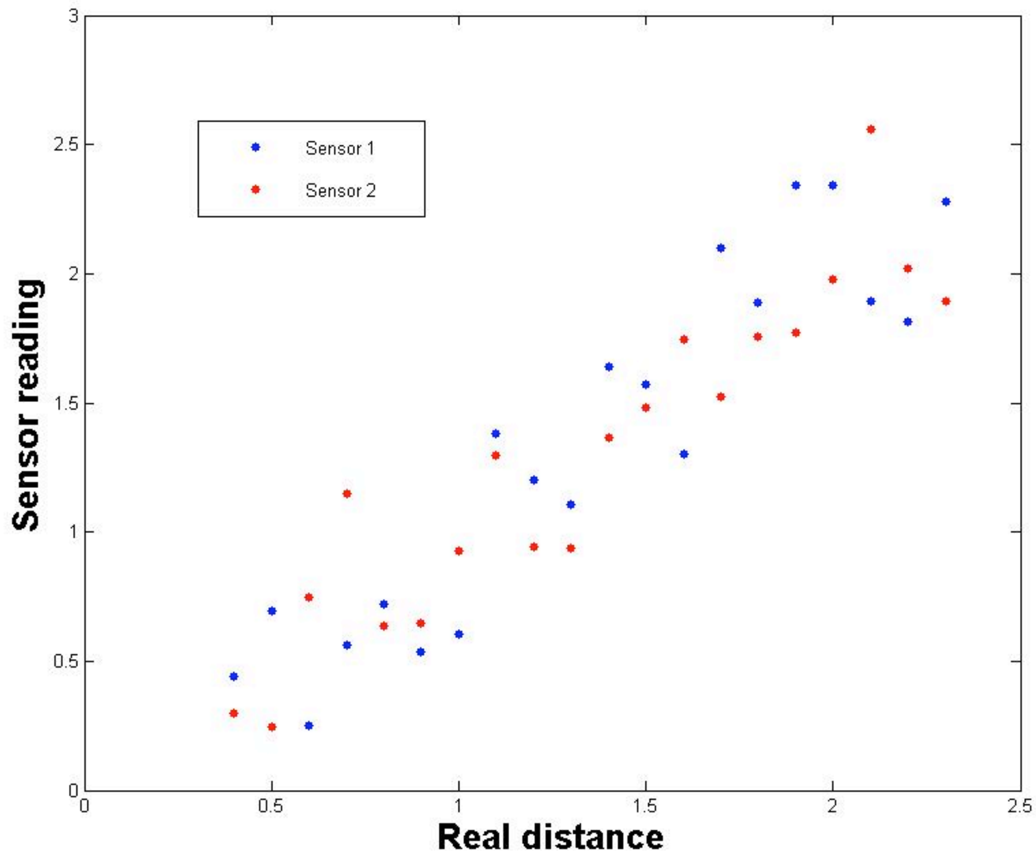
Why?

I will leave these derivation to you …

Sensor data



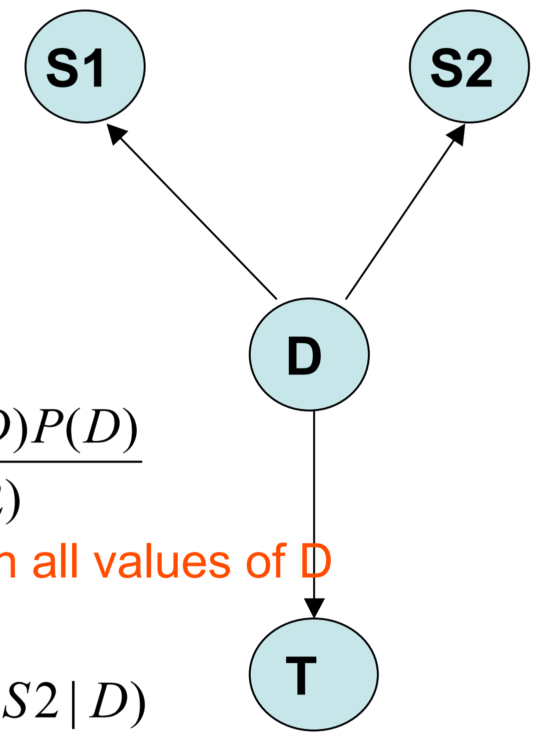
S1
S2
D
T



Sensor 1
Sensor 2
Sensor reading
Real distance
0
0.5
1
1.5
2
2.5
3

What value would we infer for D given S1,S2?

- We will write the general terms and then use the network model to simplify it.
- The important issue is how to work with Gaussians



Bayes rule

$$P(D \mid S1, S2) = \frac{P(S1 \mid D, S2)P(D \mid S2)}{P(S1 \mid S2)} = \frac{P(S1 \mid D, S2)P(S2 \mid D)P(D)}{P(S1 \mid S2)P(S2)}$$

Using network structure

Assuming equal prior on all values of D

$$\arg\max_D \frac{P(S1 \mid D)P(S2 \mid D)P(D)}{P(S1 \mid S2)P(S2)} = \arg\max_D P(S1 \mid D)P(S2 \mid D)$$

$$P(S1 \mid D)P(S2 \mid D) = \frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(D-S1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(D-S2)^2}{2\sigma_2^2}}$$

Model for sensor data

$$\log\left(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(D-S1)^2}{2\sigma_1^2}} \frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(D-S2)^2}{2\sigma_2^2}}\right) = \log\left(\frac{1}{\sqrt{2\pi\sigma_1^2}} \frac{1}{\sqrt{2\pi\sigma_2^2}}\right) - \frac{(D-S1)^2}{2\sigma_1^2} - - \frac{(D-S2)^2}{2\sigma_2^2}$$

$$\frac{\partial}{\partial D} \log\left(\frac{1}{\sqrt{2\pi\sigma_1^2}} \frac{1}{\sqrt{2\pi\sigma_2^2}}\right) - \frac{(D-S1)^2}{2\sigma_1^2} - - \frac{(D-S2)^2}{2\sigma_2^2} = -2\frac{(D-S1)}{2\sigma_1^2} - 2\frac{(D-S2)}{2\sigma_2^2}$$

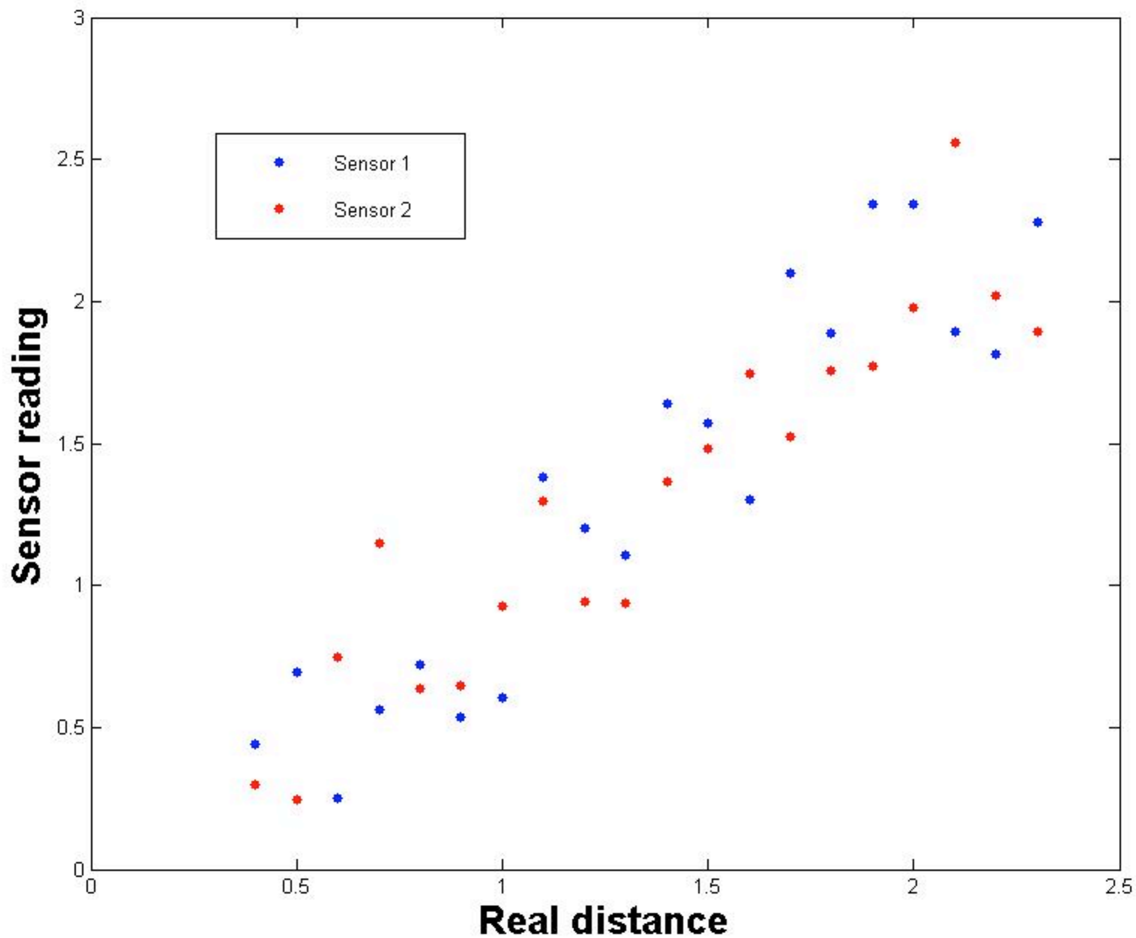
$$\Rightarrow -2\frac{(D-S1)}{2\sigma_1^2} - 2\frac{(D-S2)}{2\sigma_2^2} = 0 \Rightarrow$$

$$D = \frac{S1\sigma_2^2 + S2\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \Rightarrow$$

Only if $\sigma_1 = \sigma_2$

$$D = \frac{S1 + S2}{2}$$

Sensor data



$$D = \frac{S1 + S2}{2}$$

Lets go back to Naïve vs.full model

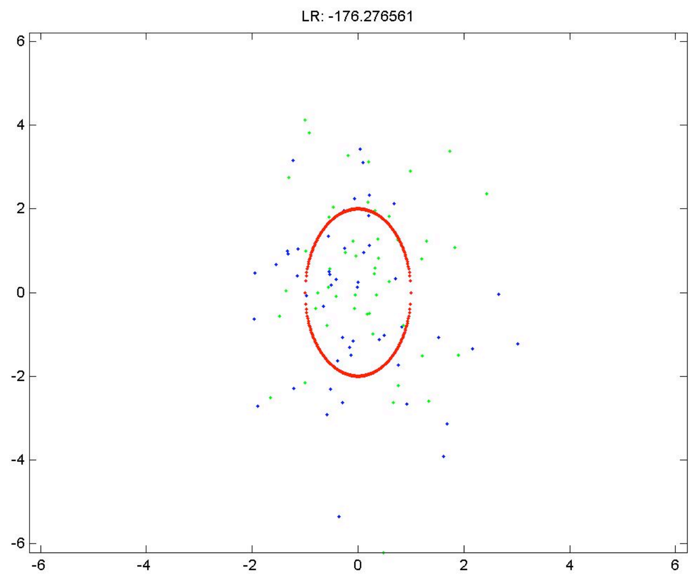
What should I use?

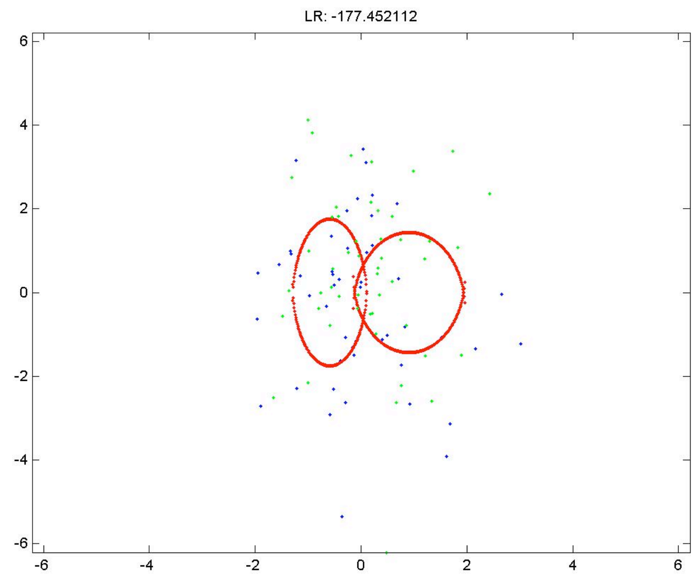
This can be determined based on:

- Training data size
- Cross validation
- Likelihood ratio test

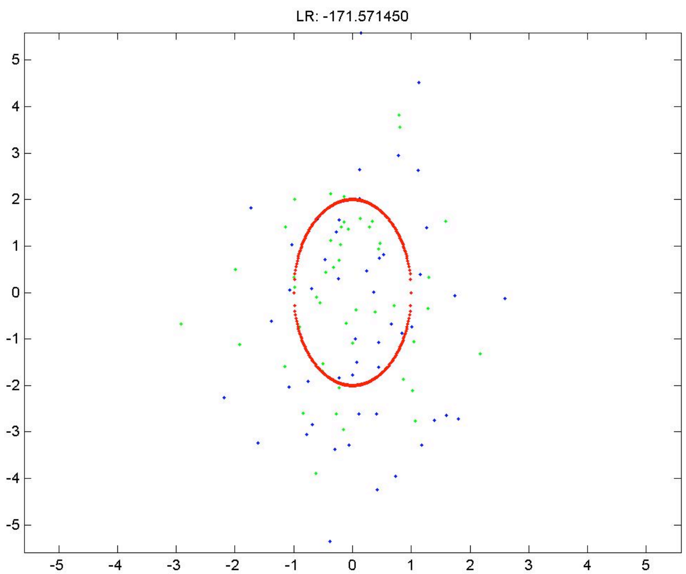
Cross validation is one of the most useful tricks in model fitting

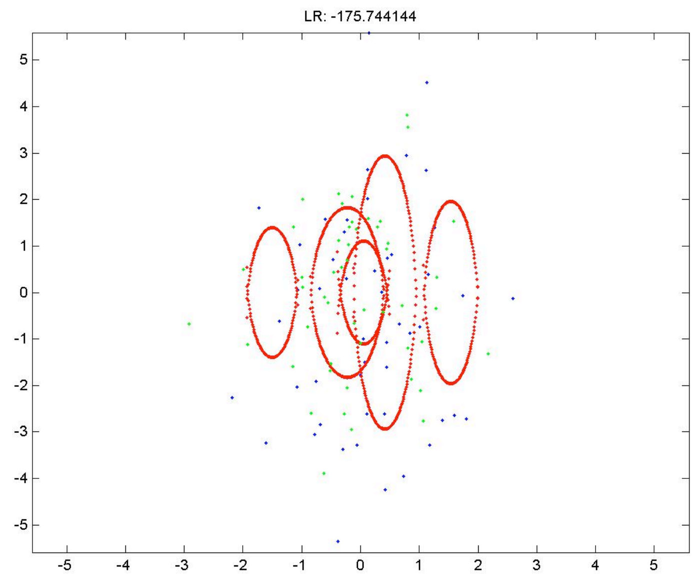
Cross validation





Cross validation





Multi-Variate Gaussian

- A multivariate Gaussian model: **x** ~ N(μ,Σ) where

$$p(\mathbf{x} \mid \Theta) = \frac{1}{2\pi^{p/2} \mid \Sigma \mid^{1/2}} e^{-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)}$$

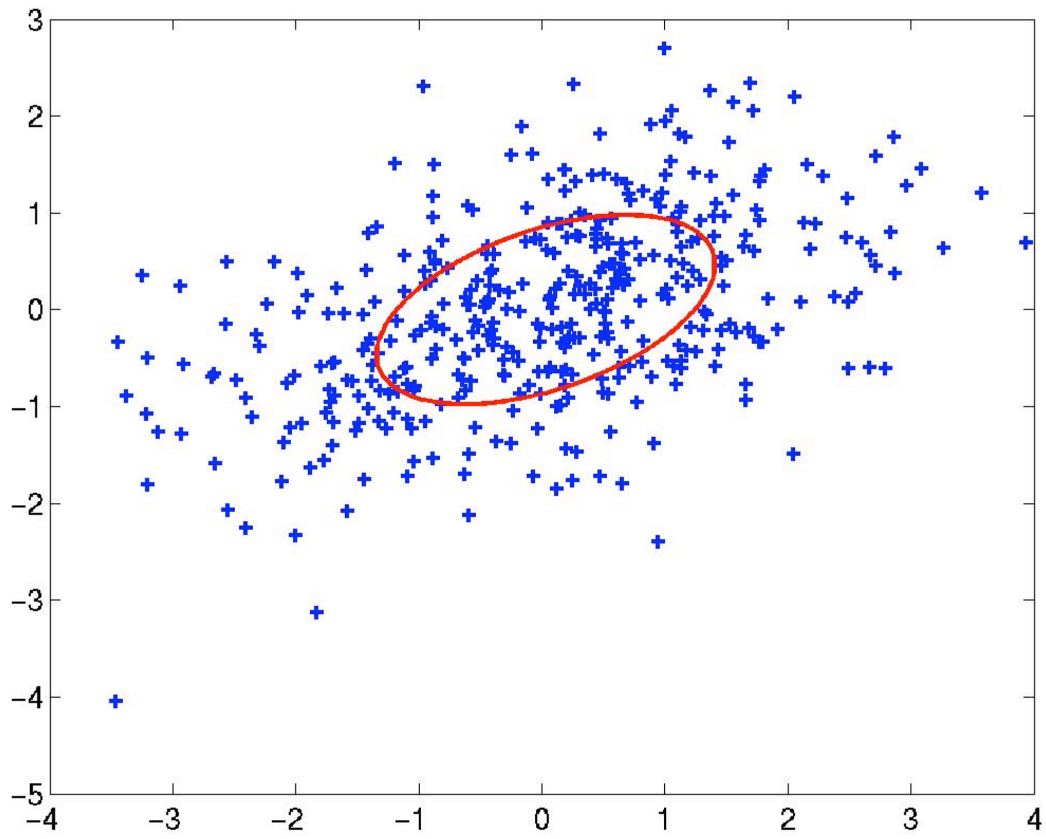
Here μ is the mean vector and Σ is the covariance matrix

$\mu = \{\mu_1, \mu_2\}$ Σ =

var(x_1)	cov(x_1,x_2)
cov(x_1,x_2)	var(x_2)

- The covariance matrix captures linear dependencies among the variables

Example



Important points

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