LEARNING WITH BAYESIAN NETWORKS

Author: David Heckerman

Presented by: Dilan Kiley

Adapted from slides by: Yan Zhang - 2006, Jeremy Gould – 2013, Chip Galusha -2014

May 6th, 2016

OUTLINE

1. Bayesian Approach
2. Bayesian Networks
3. Bonus
4. Exam Questions

OUTLINE

1. **Bayesian Approach**
2. Bayesian Networks
3. Bonus
4. Exam Questions

OUTLINE

1. **Bayesian Approach**
    - **Bayesian vs Classical probability schools**
    - **Bayes Theorem Review**
    - **Example**
2. Bayesian Networks
3. Conclusion
4. Exam Questions

BAYESIAN VS CLASSICAL PROBABILITY

- Classical probability refers to the true or ***actual*** probability of an event occurring. It is not concerned with observed behavior, and addresses these probabilities more like physical properties.
- Bayesian probability refers to a ***degree of belief*** in an event occurring. It is a property of the person who assigns the probability. Still conforms to rules of probabilities.

IS THIS MAN A MARTIAN ???

Steve Buscemi

STEVE BUSCEMI. MARTIAN?

Consider TWO concepts

1. Hypothesis(H) = He either is or is not a martian.
2. Data (D) = Some set of information about Steve. Financial data, phone records, tabloids, favorite dining establishments…

STEVE BUSCEMI. MARTIAN?

Hypothesis, H = Steve is a martian

Frequentist Approach

Given H there is a probability P of seeing this data

P(D | H)

Considers absolute ground truth.

Bayesian Approach

Given this data there is a probability P of hypothesis, H begin true.

P(H | D)

Probability indicates our level of *belief* in the hypothesis.

TWO SCENARIOS.

Consider the following examples:

1. Repeated tosses of a sugar cube into a puddle.
    A. Classic: Hard time measuring the probability (needs repeated trials.
    B. Bayesian: Restrict attention to next toss, assign belief.
2. Will the Steelers win the Super Bowl this year?
    C. Classic: ***crickets***
    D. Bayesian: Assign a probability to the scenario.

BAYESIAN BOOTCAMP

- Suppose we have observed N events. The Bayesian approach restricts its prediction to the next (N+1) occurrence of this event.
- A classical approach would predict the likelihood of any event regardless of the number of occurrences.

NOTE!

Bayesian approach can be **updated** as new data is observed.

BAYES THEOREM

We are familiar with the concept of Bayes Theorem

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

P(A) and P(B) are probabilities of A and B independently

P(A | B) is the probability of A *given* that B is true

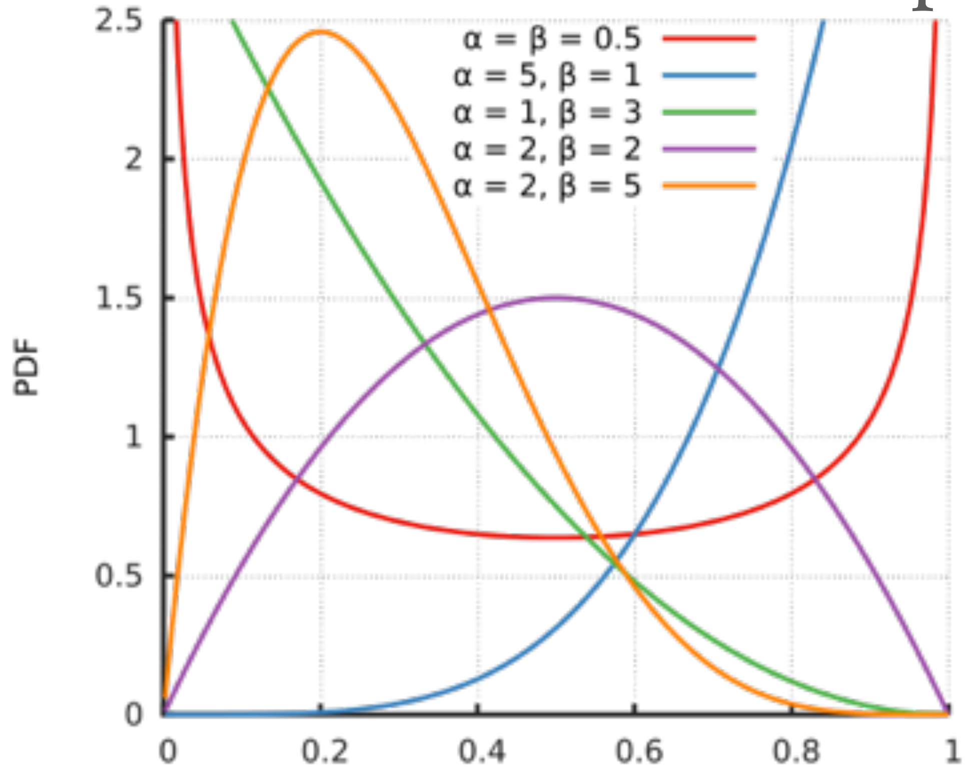
TECHNICAL EXAMPLE – COIN TOSSING

Suppose I am going to toss a coin 100 times.

We will denote the *random variable X* as the outcome of *one flip*.

Before I do this (flipping) experiment I have some belief in mind. This is called the **prior probability.**

Here we will use a Beta distribution as our prior belief in the outcome.



TECHNICAL EXAMPLE - COIN TOSSING

Back to Bayes Theorem we have

$$P(H|D) = \frac{P(H)P(D|H)}{P(D)}$$

We replace P(D|H) with the likelihood function for binomial sampling: $x^h(1-x)^t$

Bringing us to: $P(H|D) = \dfrac{P(H)H^h(1-H)^t}{P(D)}$

P(H) is our **prior**, P(H|D) is our **posterior**.

TECHNICAL EXAMPLE - COIN TOSSING

Really, P(H) is P(H|state of information). By assuming a Beta prior (and supposing we anticipate a fair coin) we have

$$P(H|\mu) = \mathrm{Beta}(H|\alpha_t, \alpha_h) = \frac{\Gamma(\alpha)}{\Gamma(\alpha_h)\Gamma(\alpha_t)} H^{\alpha_h - 1}(1 - H)^{\alpha_t - 1}$$

Thankfully, Beta functions work out quite nicely to

$$\mathrm{Beta}(H|\alpha_h + h, \alpha_t + t)$$

Granted, this is still just our **prior**!

$$\alpha_t, \alpha_h = \text{parameters of Beta function}$$

$$h, t = \text{actual observed number of heads/tails}$$

TECHNICAL EXAMPLE - COIN TOSSING

To calculate the probability that the next flip will be heads, we integrate (average) over all possible values the coin can take to determine the probability that the N+1 toss will come up heads.

In our case (finally using Beta functions pays off) this simplifies greatly to

$$P(X_{N+1} = \text{heads}|D) = \frac{\alpha_h + h}{\alpha_h + \alpha_t + N}$$

TECHNICAL EXAMPLE - COIN TOSSING

Suppose that we saw 45 heads and 55 tails.

Then $$P(X_{N+1} = \text{heads}|D) = \frac{\alpha_h + h}{\alpha_h + \alpha_t + N}$$

becomes

$$\frac{5+45}{5+5+100} \approx .46$$

We then could update our beliefs off of this finding the NEXT time we want to infer the the probability of heads. i.e. updating our prior.

VOCABULARY REVIEW

Prior = P(H|belief): Prior probability of a particular hypothesis given no observed data.

Posterior = P(H|D): Probability of the hypothesis given that we have observed the data in D.

Likelihood = P(D|H): Likelihood of the data, D being observed given our hypothesis.

ADVANTAGES OF BAYESIAN THINKING

The following will become useful when we learn Bayesian Networks and start to use them from a data mining standpoint.

1. Chain Rule applies to Bayesian probabilities!

$$P(A, B, C) = P(A|B, C) \times P(B|C) \times P(C)$$

2. Using the Bayesian model we can **update our beliefs** based on new data we receive.

OUTLINE

1. ~~Bayesian Approach~~
2. **Bayesian Networks**
    - Structure
    - Inference
    - Learning Probabilities
    - Unknowns
    - Learning structure
    - Example
3. Bonus
4. Exam Questions

BAYESIAN NETWORKS…WHAT ARE THEY GOOD FOR?

A Bayesian Network is a ***probabilistic graphical model*** which represents a set of **random variables** and their **conditional dependancies** on a graph (network) structure.

To construct a Bayes Net we need…

1. A set of random variables X ={X1,X2,….Xn}
2. A Network Structure
3. Conditional Probability Table for the random variables.

It is possible to begin with missing/incomplete data

SLOGGING THROUGH NOTATION

S : The network structure

S^h : The hypothesis corresponding to the network

S_c : A Complete network structure

X_i : A variable, and its corresponding node

Pa_i : The variable or node corresponding to the parents of node X_i

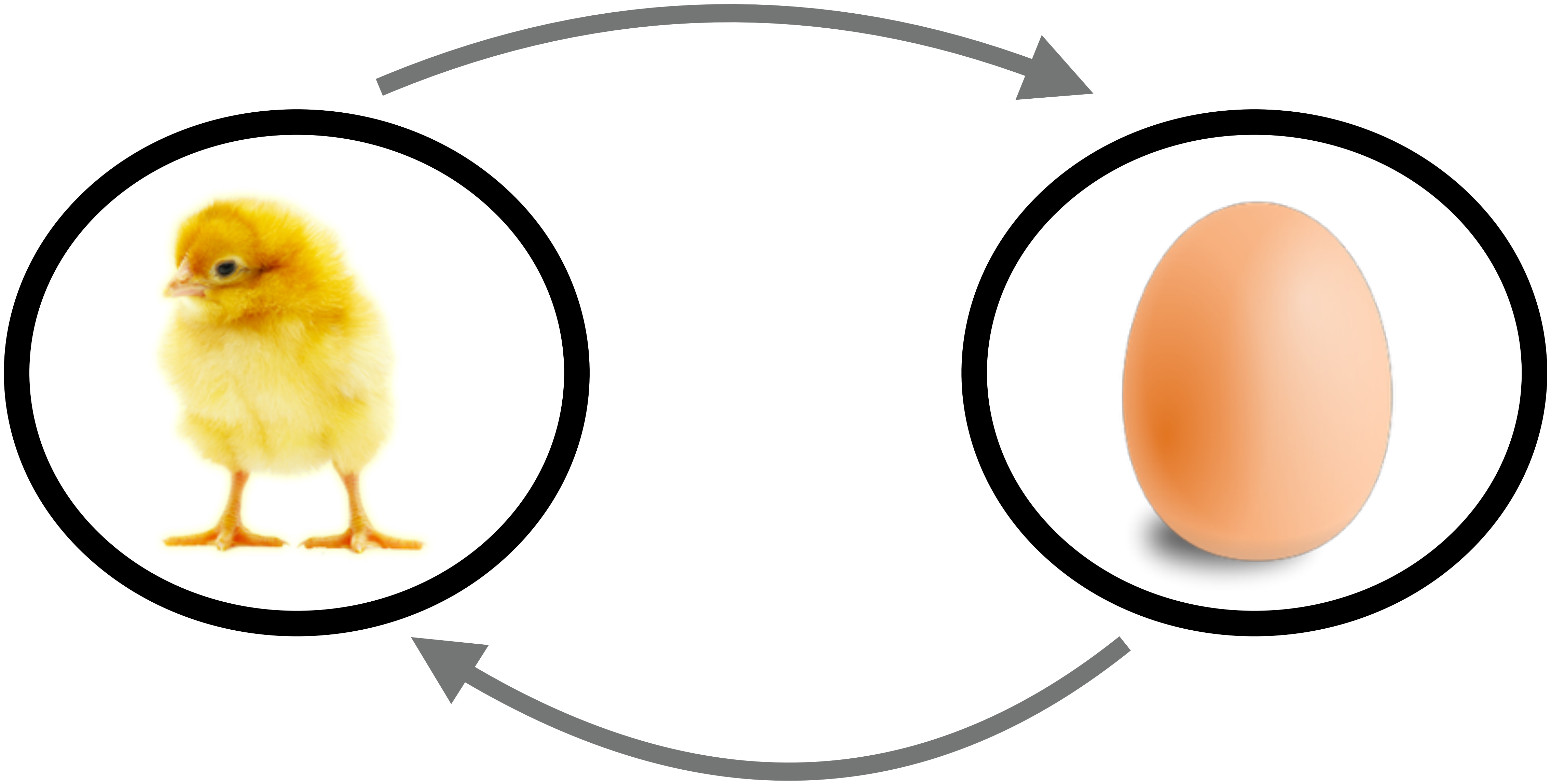
D : Data

Each random variable will be represented with a node.

Edges are drawn to indicate cause and effect.

Network must be ***directed acyclic***.

WHY MUST THE GRAPH BE DIRECTED ACYCLIC???



SETTING UP YOUR BAYES NET

In general when applying Bayesian Networks to a problem there are four (4) major tasks:

1. Correctly identify the goal (predict,update,infer)
2. Identify many possible observations that may be relevant
3. Determine what subset of these observations is worth modeling
4. Organize observations into variables **and** determine order of network.

APPLICATION: CREDIT CARD FRAUD

Consider the scenario in which we are attempting to detect credit card fraud.

We first identify our variables:

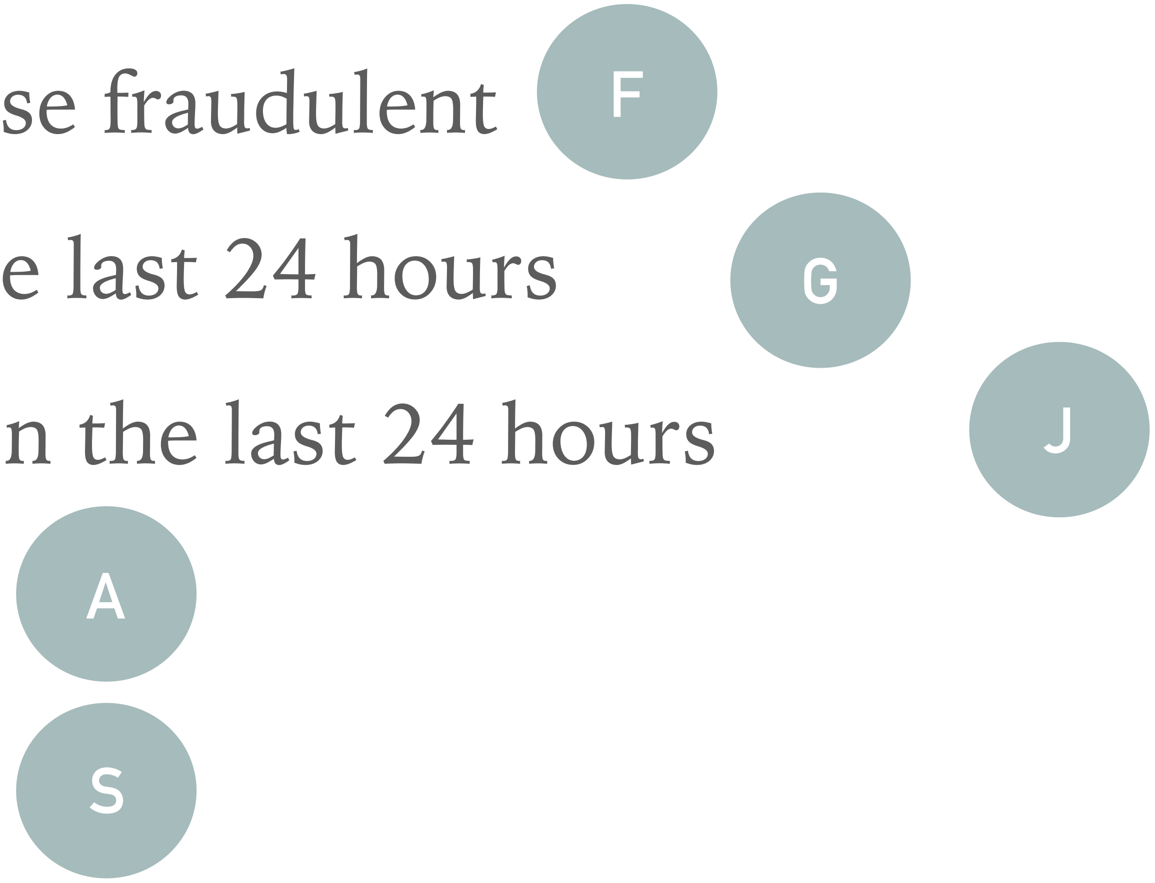
Fraud (F) : is the current purchase fraudulent

Gas (G) : was gas bought in the last 24 hours

Jewelry (J): was jewelry bought in the last 24 hours

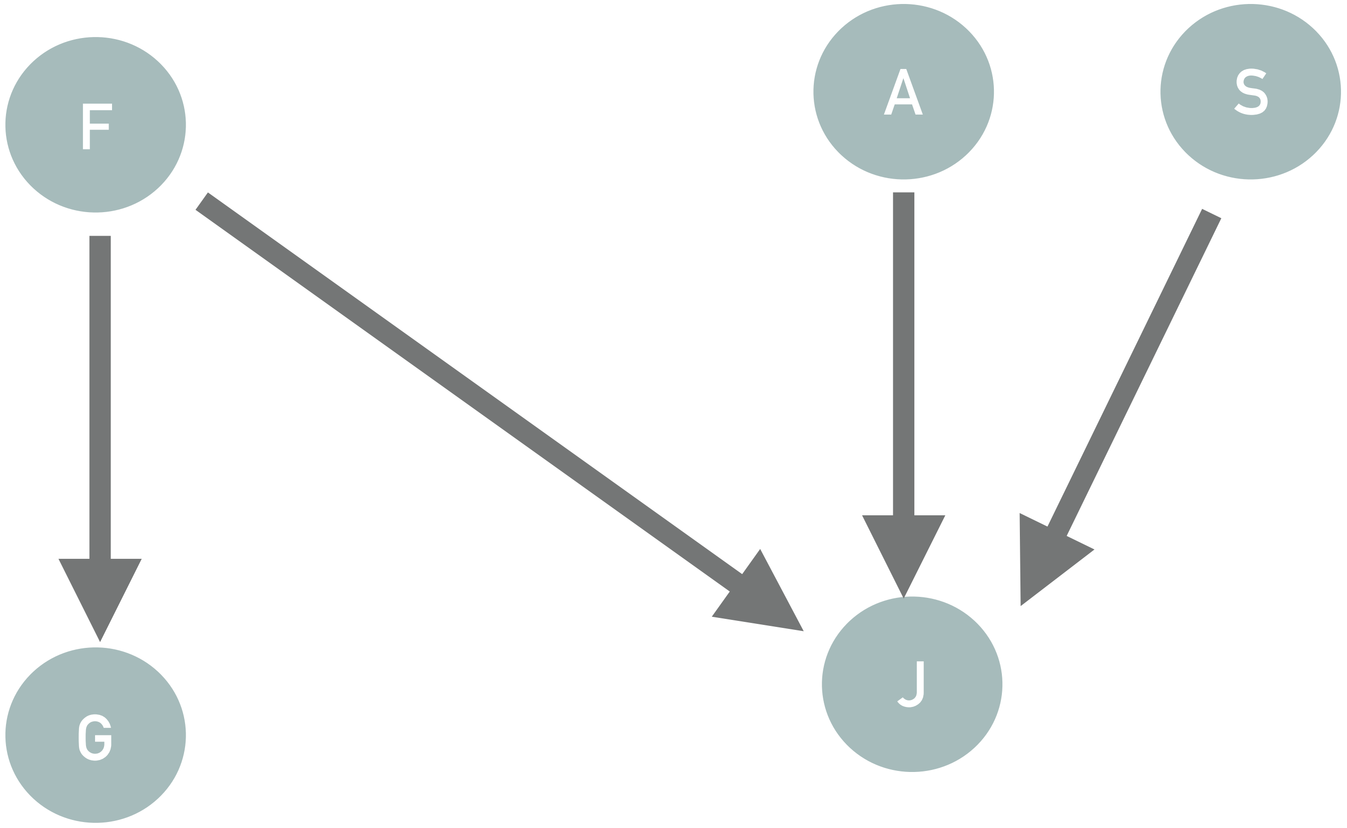
Age (A) : age of card holder

Sex (S) : sex of card holder



CONSTRUCTING THE NETWORK

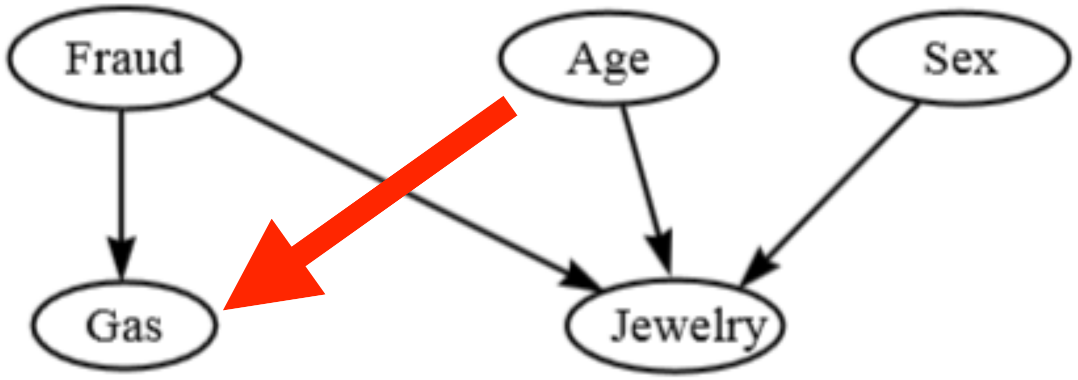
Set up nodes to indicate the causal relationships between variables.



UNPACKING CONDITIONAL DEPENDANCIES

Given an ordering of the nodes we can infer the following conditional dependancies and independencies.

Dependancies: shown on graph



Independencies:

$$P(A|F) = P(A)$$
$$P(S|F, A) = P(S)$$
$$P(G|F, A, S) = P(G|F)$$
$$P(J|F, A, S, G) = P(J|F, A, S)$$

This uses the ordering (F,A,S,G,J)

What about the red arrow?

P(G|A) = P(G)

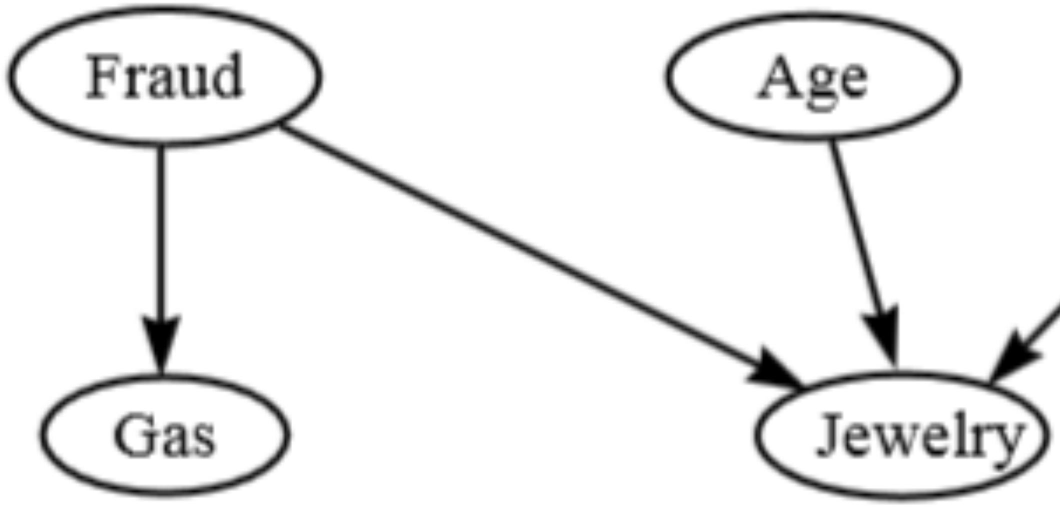
REALIZE THE LOCAL PROBABILITY DISTRIBUTIONS

$p(f=\text{yes}) = 0..00001$

$p(a=<30) = 0.25$

$p(a=30\text{-}50) = 0.40$

$p(s=\text{male}) = 0.5$



$p(g=yes|f=yes) = 0.2$

$p(g=yes|f=no) = 0.01$

$p(j=yes|f=yes,a=*,s=*) = 0.05$

$p(j=yes|f=no,a=<30,s=male) = 0..0001$

$p(j=yes|f=no,a=30\text{-}50,s=male) = 0.0004$

$p(j=yes|f=no,a=>50,s=male) = 0.0002$

$p(j=yes|f=no,a=<30,s=female) = 0..0005$

$p(j=yes|f=no,a=30\text{-}50,s=female) = 0.002$

$p(j=yes|f=no,a=>50,s=female) = 0.001$

USING THE BAYESIAN NETWORK: INFERENCE

Suppose we wanted to use our Bayesian Network to calculate our the probability of Fraud given observations of the other variables.

$$P(F|A, S, G, J) = \frac{\text{joint probability}}{\text{marginal probability}} = \frac{P(F, A, S, G, J)}{P(A, S, G, J)}$$

USING THE BAYESIAN NETWORK: INFERENCE

We can implement the Chain Rule and our conditional independencies to simplify

$$P(F|A,S,G,J) = \frac{\text{joint probability}}{\text{marginal probability}} = \frac{P(F,A,S,G,J)}{P(A,S,G,J)}$$

$$\frac{P(F,A,S,G,J)}{P(A,S,G,J)} = \frac{P(F,A,S,G,J)}{\Sigma_{F'} P(F',A,S,G,J)}$$

BOARD DEMO!

$$\boxed{\frac{P(F,A,S,G,J)}{\Sigma_{F'}P(F',A,S,G,J)}}$$

Conditional Independencies

$$\boxed{\begin{aligned} P(A|F) &= P(A) \\ P(S|F,A) &= P(S) \\ P(G|F,A,S) &= P(G|F) \\ P(J|F,A,S,G) &= P(J|F,A,S) \end{aligned}}$$

Chain Rule

$$\boxed{P(A,B,C) = P(A|B,C) \times P(B,C)}$$

Bayes Theorem

$$\boxed{P(A|B)P(A) = P(B|A)P(B)}$$

$$\boxed{\frac{P(F)P(G|F)P(J|F,A,S)}{\sum_{F'}P(F')P(G|F')P(J|F',A,S)}}$$

USING THE BAYESIAN NETWORK: LEARNING

Thus far we have not taken advantage of the fact that we can update our Bayesian Network with new data. Lets change that!

First we assemble the joint probability distribution for X(set of nodes) encoded in our network, S

$$P(x|\theta_s, S^h) = \prod_{i=1}^{n} P(x_i|pa_i, \theta_i, S^h)$$

S^h : current hypothesis that the joint probability can be factor under S

θ_i : vector of parameters for the distribution $P(x_i|pa_i, \theta_i, S^h)$

θ_s : vector of parameters $\theta_1 ... \theta_n$

USING THE BAYESIAN NETWORK: LEARNING

Now that we established that, we can can state our problem of *learning probabilities in a Bayesian Network* as:

Given a random sample of data, D compute the posterior distribution $P(\theta_s, |D, S^h)$

This relies on two (2) assumptions:

1. There is no missing data in D.
2. The parameter vectors are independent.

$$P(X, Y) = P(X)P(Y)$$
$$P(X, Y|D) = P(X|D)P(Y|D)$$

USING THE BAYESIAN NETWORK: LEARNING

Given these assumptions we can write

$$\text{Prior distribution} = P(\theta_s|S^h) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} P(\theta_{ij}|S^h)$$

$$\text{Posterior distribution} = P(\theta_s|D, S^h) = \prod_{i=1}^{n}\prod_{j=1}^{q_i} P(\theta_{ij}|D, S^h)$$

So we can update each vector of theta **independently** just like in the one variable case.

here q_i corresponds to the indexing of the parents of each of the n nodes

USING THE BAYESIAN NETWORK: LEARNING

Assuming a Dirichlet prior distribution for each vector we arrive with

$$\text{Posterior distribution} = P(\theta_{i,j}|D,S^h) = \text{Dir}(\theta_{ij}|\alpha_{ij1}+N_{ij1},...,\alpha_{i,jr_i}+N_{ijr_i})$$

As with the coin toss example we can use this to infer the probability of the next event given our data

where N_{ijk} is the number of cases in D where $X^i = x_i^k$ and $Pa_i = pa_i^j$

$$P(X_{N+1}|D,S^h)$$

Really this just boils down to cranked up version of inferring the probability that the next flip of a coin will be a heads

ADDRESSING OUR ASSUMPTIONS

This is all well and good under our assumptions, but what is we are dealing with missing data.

First we need to consider if the **missing data** is dependent or independent on variable states.

eg: dependent: missing data in a drug study might indicate the became too sick — possibly as a result of the drug — to continue to provide data.

If independent…

- Monte-Carlo Methods
- Gaussian Approximations

ADDRESSING OUR ASSUMPTIONS

In addition to our assumption concerns there could be several unknowns, namely:

1. Set of variables
   - *What should we measure, and why?*
2. Conditional Relationship Data
   - Begin with our best guess as to the causal belief in a relationship and let our model refine the values under the methods shown in the last section.
3. Network Structure
   - Don't know where/which way the edges go…

LEARNING THE NETWORK STRUCTURE

Suppose we are unsure about what the causal relationship between our variables (nodes) are.

Theoretically we can learn it using a Bayesian approach to get the posterior

$$P(S^h|D) = \frac{P(D|S^h)P(S^h)}{P(D)}$$

Unfortunately, the number of possible network structures grows exponentially! In other words we have to compute the marginal likelihood of the data $P(D|S^h)$ for every possible configuration.

LEARNING THE NETWORK STRUCTURE

In practice there are two approaches to address this combinatorial explosion:

1. Model Selection
   - Select a ‘good’ model from all possible structure and assume it is correct
2. Selective Model Averaging
   - To select a hand-full of ‘good’ model and assume they represent all possibilities.

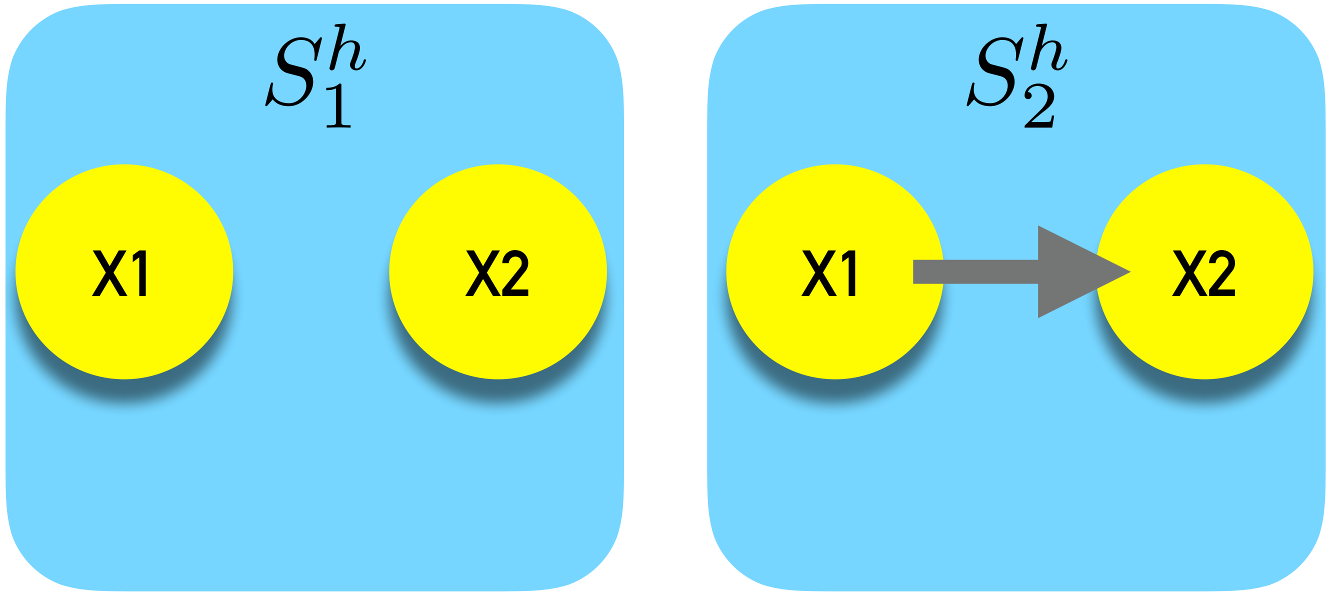
LEARNING THE NETWORK STRUCTURE

In practice there are two approaches to address this combinatorial explosion:

1. Model Selection
    - Select a ‘good’ model from all possible structure and assume it is correct
2. Selective Model Averaging
    - To select a hand-full of ‘good’ model and assume they represent all possibilities.

LEARNING THE NETWORK STRUCTURE

We consider a toy example with…you guessed it…coins!



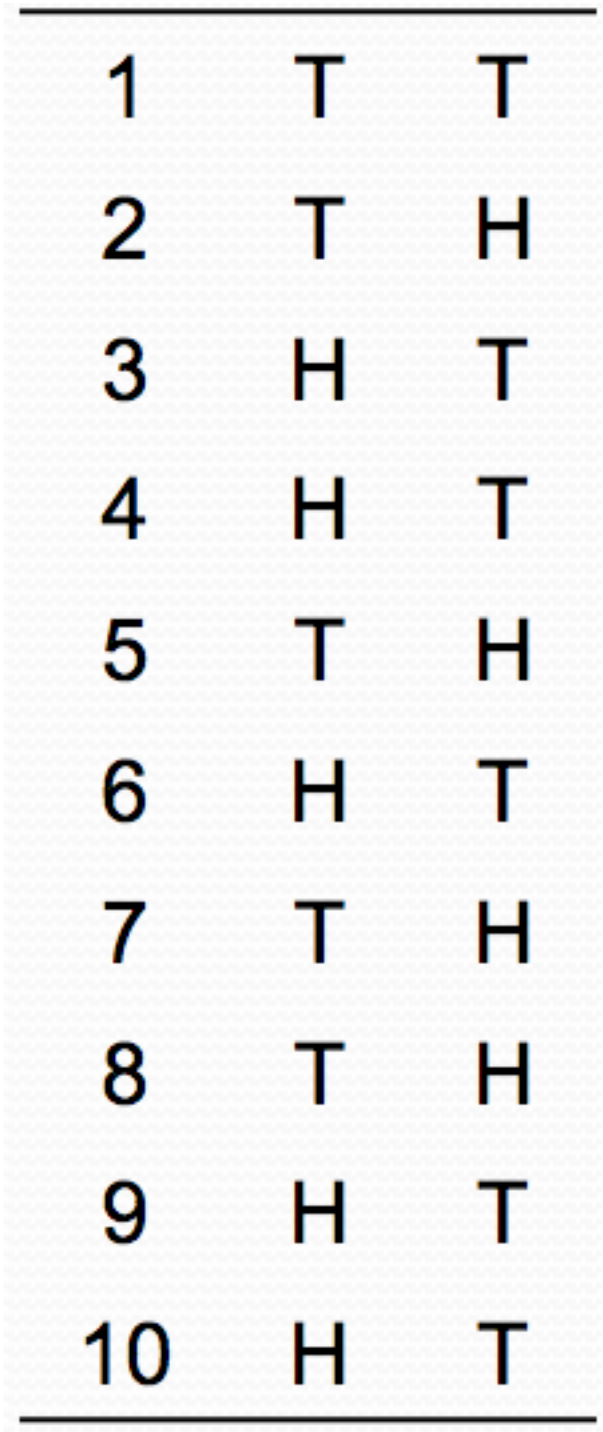
$P(X_2|X_1)$

p(H|H)	=	0.1
p(T|H)	=	0.9
p(H|T)	=	0.9
p(T|T)	=	0.1

Flip coin X and coin Y. Does coin X influence coin Y?

Begin by assuming that both networks are equally likely (prior).

LEARNING THE NETWORK STRUCTURE: LIKELIHOODS



	$X1$	$X2$
1	T	T
2	T	H
3	H	T
4	H	T
5	T	H
6	H	T
7	T	H
8	T	H
9	H	T
10	H	T

$$P(D|S^h) = \prod_{d=1}^{10} \prod_{i=1}^{2} P(X_{di}|P\alpha_i, S^h)$$

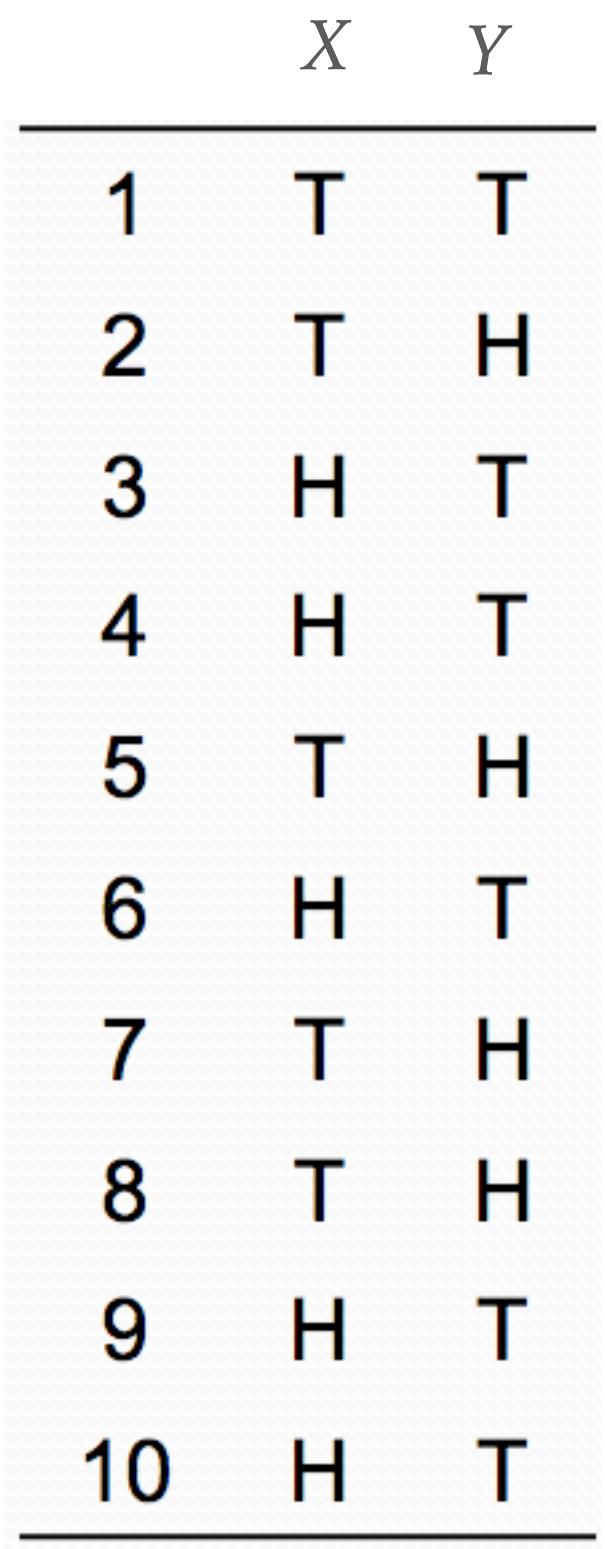
For our first hypothesis this is simple, as we assumed the coins had no effect on each other.

$$P(X_2|X_1) = P(X_2)$$

Therefore the calculation is the same for all cases thus,

$$P(D|S^h) = (0.5^2)^{10}$$

LEARNING THE NETWORK STRUCTURE: LIKELIHOODS



	X	Y
1	T	T
2	T	H
3	H	T
4	H	T
5	T	H
6	H	T
7	T	H
8	T	H
9	H	T
10	H	T

$$P(D|S^h) = \prod_{d=1}^{10} \prod_{i=1}^{2} P(X_{di}|Pa_i, S^h)$$

For our second hypothesis this different for each combination of heads and tails

p(H|H) = 0.1
p(T|H) = 0.9
p(H|T) = 0.9
p(T|T) = 0.1

Therefore, our calculation of the likelihood is different.

$$P(D|S^h) = (0.5)^{10} \times (0.1)^1 \times (0.9)^9$$

LEARNING THE NETWORK STRUCTURE: JUDGEMENT TIME

We can now calculate the posterior for each hypothesis

$$P(S^h|D) = \frac{P(D|S^h)P(S^h)}{\sum_{i=1}^{2} P(D|S_h = S_i^h)P(S_i^h)}$$

Hypothesis 1:

$$P(S_1^h|D) = \frac{P(D|S_1^h)P(S_1^h)}{P(D|S_1^h)P(S_1^h) + P(D|S_2^h)P(S_2^h)}$$

$$P(S_1^h|D) = \frac{\Big((0.5^2)^{10}\Big) \times 0.5}{\Big((0.5^2)^{10}\Big)0.5 + \Big(0.5^{10}(0.1)(0.9)^9\Big)0.5}$$

Hypothesis 2:

$$P(S_2^h|D) = \frac{P(D|S_2^h)P(S_2^h)}{P(D|S_1^h)P(S_1^h) + P(D|S_2^h)P(S_2^h)}$$

$$P(S_2^h|D) = \frac{\Big(0.5^{10}(0.1)(0.9)^9\Big) \times 0.5}{\Big((0.5^2)^{10}\Big)0.5 + \Big(0.5^{10}(0.1)(0.9)^9\Big)0.5}$$

$$P(S_1^h|D) = 2.5\%, P(S_2^h|D) = 97.5\%$$

OUTLINE

1. ~~Bayesian Approach~~
2. ~~Bayesian Networks~~
3. **Bonus Round**
4. Exam Questions

BONUS: EM

- The Expectation Maximization algorithm is an iterative method for maximum a posteriori (MAP) estimates for parameters.
- Implemented when there is missing data, when missing data is suspected, or when the model is simplified under the assumption that there is unobserved data.
- Relies on these ***latent variables*** and parameters which we want to determine.

BONUS: EM

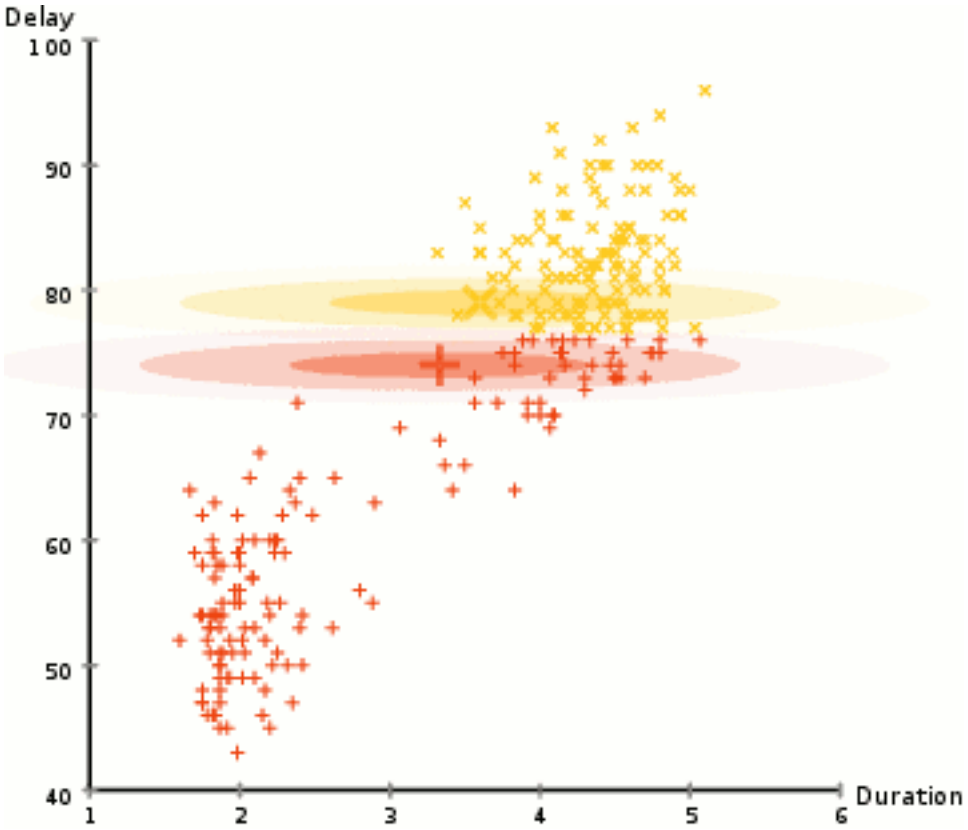
- The basic idea is to use the observed data and the assumption of latent variables to determine the parameters.
- However…doing so involves derivatives of likelihood functions (we fondly recall a few recently) of the **unknown** variables.
- In general, this is analytically unsolvable. Consider a pair of coupled equations where the solutions to the parameters depends on the solution to the latent variables, and vice versa — a **deadlock scenario**.

BONUS: EM

- In an attempt to solve this deadlock problem, the EM algorithm seeds arbitrary values for one set (i.e. the parameters or the variables).
- Next, it uses these values the estimate values for the other set.
- The cycle then repeats, using the estimations for the second set to improve the estimates for the first.
- Continue doing so until the values for both sets converge.

BONUS: EM

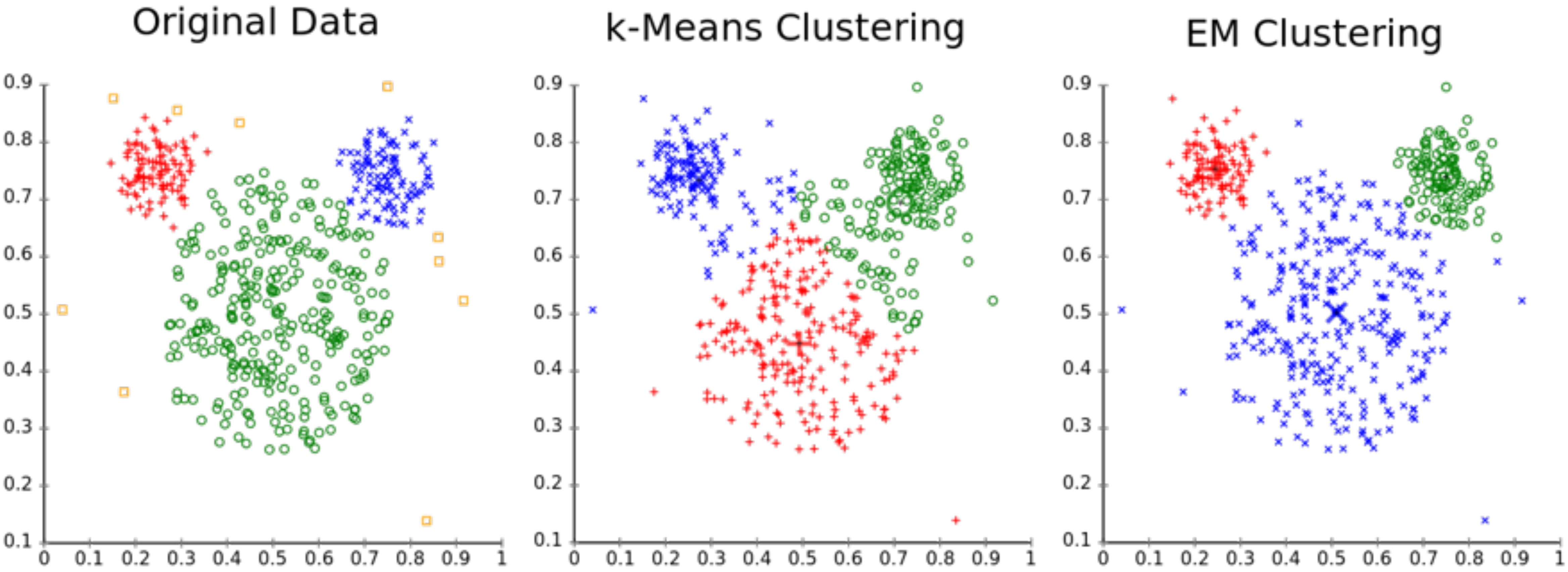
- EM is commonly used as a means of clustering



BONUS: EM

- EM is commonly used as a means of clustering

Different cluster analysis results on "mouse" data set:



Original Data

k-Means Clustering

EM Clustering

OUTLINE

1. ~~Bayesian Approach~~
2. ~~Bayesian Networks~~
3. ~~Bonus Round~~
4. **Exam Questions**

EXAM QUESTIONS

1. Describe the three main components of calculating Bayesian probabilities: Prior, Posterior, Likelihood.

Prior = P(H|belief): Prior probability of a particular hypothesis given no observed data.

Posterior = P(H|D): Probability of the hypothesis given that we have observed the data in D. (What we hope to compute)

Likelihood = P(D|H): Likelihood of the data, D being observed given our hypothesis.

EXAM QUESTIONS

2. Briefly describe/explain Bayesian Networks and list two uses of these networks.

A graphical model which encodes probabilistic relationships among variables of interest.

These network can be used to…

1. Perform inference (probability of fraud)
2. Learn probabilities (update the local probability distributions)

EXAM QUESTIONS

3. What are three possible scenarios when one might turn to use EM (Expectation Maximization)

- EM is all about **missing data**. Thus we might use it when (1) we know data is missing, (2) we suspect there is missing data, or (3) when the proposed model is simplified by assuming there is missing data.

THE END

QUESTIONS?