# COMP 424 - Artificial Intelligence
# Lecture 17: Learning Bayesian Networks

Instructor: Jackie CK Cheung (jcheung@cs.mcgill.ca)

Readings: R&N Ch 20

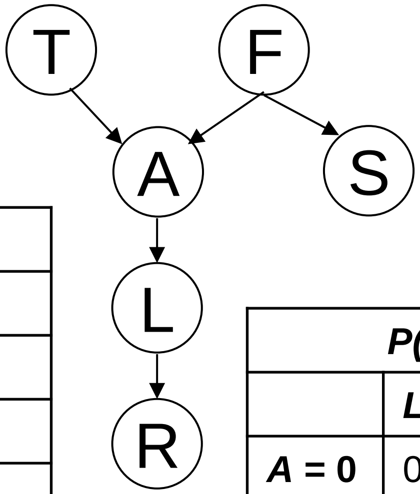
# Review: Inference in Bayes nets

- Bayes nets encode information about conditional independence between variables.

$$P(x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} P(x_i | \text{parents}(X_i))$$

- **Variable elimination** algorithm gives us dynamic programming approach for inferences in Bayes.
- Complexity of inference depends a lot on network's structure.
  - Inference is efficient (poly-time) for tree-structured networks.
  - In worse-case, inference is NP-complete.
- Can leverage DAG structure to facilitate inference.

# Constructing Belief Nets: CPDs




| P(T) | |
|---|---|
| ***T* = 1** | ***T* = 0** |
| 0.02 | 0.98 |

| P(F) | |
|---|---|
| ***F* = 1** | ***F* = 0** |
| 0.01 | 0.99 |

| P(S \| F) | | |
|---|---|---|
| | ***S* = 1** | ***S* = 0** |
| ***F* = 0** | 0.01 | 0.99 |
| ***F* = 1** | 0.90 | 0.10 |

| P(A\| T, F) | | |
|---|---|---|
| | ***A* = 1** | ***A* = 0** |
| ***T*=0, *F*=0** | 0.0001 | 0.9999 |
| ***T*=0, *F*=1** | 0.99 | 0.01 |
| ***T*=1, *F*=0** | 0.85 | 0.15 |
| ***T*=1, *F*=1** | 0.5 | 0.5 |

| P(L \| A) | | |
|---|---|---|
| | ***L* = 1** | ***L* = 0** |
| ***A* = 0** | 0.001 | 0.999 |
| ***A* = 1** | 0.88 | 0.12 |

| P(R \| L) | | |
|---|---|---|
| | ***R* = 1** | ***R* = 0** |
| ***L* = 0** | 0.01 | 0.99 |
| ***L* = 1** | 0.75 | 0.25 |

**Where do these numbers come from?**

# Parameter Estimation

**Option 1: Ask an expert**

- Use experts to select Bayes net structure and parameters
  - Experts are often scarce and expensive.
  - Experts can be inconsistent.
  - Experts can be non-existent!

**Option 2: Estimate parameters from data**

- Estimate how the world works by observing samples!

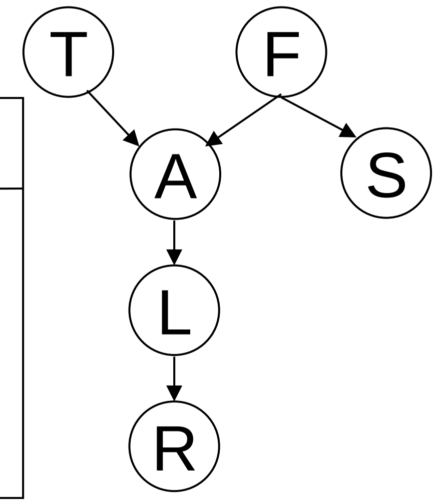
# Outline

- Learning with complete data
  - Supervised learning
  - Maximum likelihood estimation
  - Laplace smoothing
- Learning with incomplete data
  - Expectation maximization

# Learning in Bayesian networks

- Given **data** in the form of instances:

| Tampering | Fire | Smoke | Alarm | Leaving | Report |
|---|---|---|---|---|---|
| No | No | No | No | No | No |
| No | Yes | Yes | Yes | Yes | No |
| … | … | | … | … | … |




- Create a complete Bayes net!
  1. **Parameter estimation**: Given a graph structure, compute the Conditional Probability Distributions (CPDs). *Do this today!*
  2. **Structure learning**: Figure out the graph structure as well as the numbers in the CPDs. *Much harder to do!*

# Parameter estimation with complete data

- Given:
  - A Bayes network structure *G*
  - A choice of representation for the CPDs: $P(X_i \mid Parents(X_i))$
- Goal:
  - Learn the CPD in each node, such that the network is "closest" to the probability distributions that generated the data.
- For simplicity, assume all random variables in graph are binary.

# Solving a Detective Mystery

- Your friend just flipped a coin 10 times:

  *H, T, H, H, H, T, H, H, H, T (7 heads out of 10 tosses)*

- They then mixed the coin up with a bunch of other coins. You know the biases of the coins. Which one did they flip?

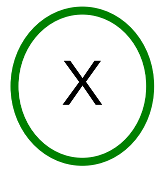
# Coin toss example

- Your friend just flipped a coin 10 times:

  *H, T, H, H, H, T, H, H, H, T*     *(7 heads out of 10 tosses)*
- Which coin did they flip? The one with P(H) =
  - 0.2
  - 0.5
  - 0.7
  - 0.9
- Which of these values is possible? Probable? Likely?

# A network with one node

- Can model as a one-node Bayesian network, X={*head*, *tail*}.




- Let $P(X) = \theta$ be the (unknown) probability of landing on *head*.

- In this case, *X* is a **Bernoulli** random variable.

- Given sequence of tosses $x_1, \ldots, x_m$, how can we estimate *P(X)*?

# Statistical parameter fitting

- Given instances $x_1, \ldots, x_m$ that are **independently identically distributed (i.i.d.)**.
  - Set of possible values for each variable in each instance is known.
  - Each instance is obtained independently of the other instances.
  - Each instance is sampled from the same distribution.

- **The learning problem**:

  Find a set of parameters $\theta$ such that the data can be summarized by a probability $P(x_j \mid \theta)$
  - $\theta$ depends on the family of probability distributions we consider (e.g. Bernoulli: $\theta=\{p\}$, Gaussian: $\theta=\{\mu, \sigma^2\}$, etc.).

# How good is a parameter set?

- It depends on how likely it is to generate the observed data.
- Let $D$ be the data set (all the instances).
- The **likelihood** of parameter set $\theta$ given data set $D$ is defined as:

$$L(\theta \mid D) = P(D \mid \theta)$$

- If the **instances are i.i.d.** we have:

$$L(\theta \mid D) = P(D \mid \theta) = P(x_1, x_2, \ldots, x_m \mid \theta) = \Pi_{j=1:m} P(x_j \mid \theta)$$

# Example: Coin tossing

- Suppose you see the following data: $D = H, T, H, T, T$

  Recall $\theta = Pr$ *(Coin lands on Head)*

- What is the likelihood of this sequence for parameter $\theta$ ?

  $$L(\theta \mid D) = \theta (1 - \theta) \theta (1 - \theta) (1 - \theta)$$

- Likelihood has a familiar form:

  $$L(\theta \mid D) = \theta^{N(H)} (1 - \theta)^{N(T)}$$

  where $N(H)$ and $N(T)$ are numbers of heads and tails observed.

# Sufficient statistics

- To compute the likelihood in the coin tossing example, we only need to know *N(H)* and *N(T)* (number of heads and tails), not the full dataset.

- Here *N(H)* and *N(T)* are sufficient statistics for this probabilistic model (Bernoulli distribution).
  - A sufficient statistic of the data is a function of the data that summarizes enough information to compute the likelihood.

- Formally, *s(D)* is a sufficient statistic if, for any two datasets *D* and *D'*:

$$s(D) = s(D') \quad \Rightarrow \quad L(\theta | D) = L(\theta | D')$$

# Maximum likelihood estimation (MLE)

- Choose parameters that maximize the likelihood function.
- We want to maximize:

  $$L(\theta \mid D) = \prod_{j=1:m} P(x_j \mid \theta)$$

  - This is a product and products are hard to maximize!

- Instead, we can maximize:

  $$\log L(\theta \mid D) = \sum_{j=1:m} \log P(x_j \mid \theta)$$

  - To maximize, take the derivates of this function with respect to $\theta$ and set them to 0 (calculus!).

# MLE applied to the Bernoulli model

- The likelihood is:

$$L(\theta \mid D) = \theta^{N(H)} (1 - \theta)^{N(T)}$$

- The log-likelihood is:

$$\log L(\theta \mid D) = N(H) \log \theta + N(T) \log (1 - \theta)$$

- Take the derivative of the log-likelihood and set it to 0:

$$d \log L(\theta \mid D) / d\theta = N(H) / \theta - N(T) / (1 - \theta) = 0$$

- Solving this gives:

$$\theta = N(H) / ( N(H) + N(T) )$$

- This is a nice, intuitive answer – it is simply the proportion of times the coin comes up heads!

# MLE applied to a categorical distribution

- Can show, with more calculus (including Lagrange multipliers – eek!), that the intuitive MLE answer generalizes to the case when there are k outcomes.

- Assume outcomes are {1, 2, …, k}, then parameters are $\theta = \{\theta_1, \theta_2, \dots, \theta_k\}$, where $\sum_i \theta_i = 1$

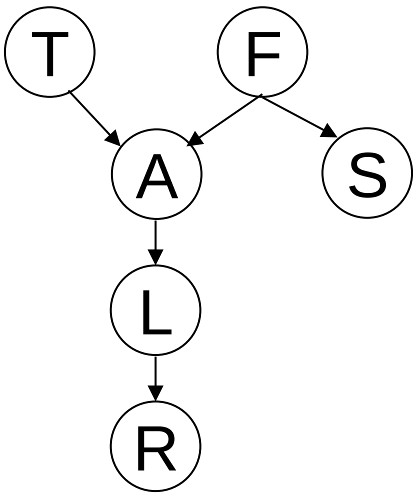
- In a data set of N samples,

$$\theta_i^{MLE} = N_i / N$$

# Parameter estimation in a Bayes net

- Instances are of the form:

$$x_j = \langle t_j, f_j, a_j, s_j, l_j, r_j \rangle, j = 1,\ldots m$$

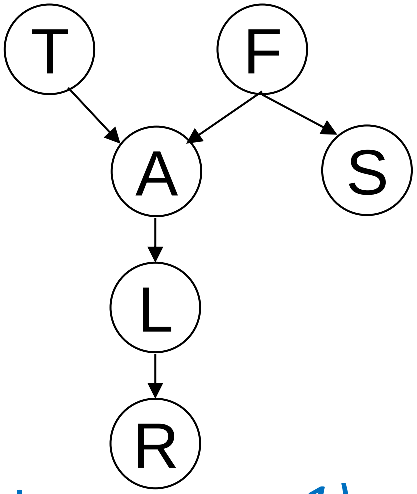



- What parameters are we trying to estimate?

$$\theta = \{ P(T), P(F), P(A|T,F), P(A|T,\neg F), P(A|\neg T,F), P(A|\neg T,\neg F),$$
$$P(S|F), P(S|\neg F), P(L|A), P(L|\neg A), P(R|L), P(R|\neg L) \}$$

i.e., Set of parameters in the conditional probability distributions

# Alarm example




- Given *m* instances, how do we compute $P(A)$ ?

  $P(A)$ = (# instances with $a_j = 1$) / $m$

- How do we compute $P(L=1 \mid A=1)$?

  $P(L \mid A) = P(L, A) \ / \ P(A)$

  = (# instances $l_j=1$ and $a_j = 1$) / (# instances $a_j = 1$)

- How do we compute $P(A=1 \mid T=1, F=0)$ ?

  $P(A \mid T, \neg F) = P(A, T, \neg F) \ / \ P(T, \neg F)$

# Parameter estimation for general Bayes nets

- Generalizing, for any Bayes net with variables $X_1, \ldots, X_n$:

$L(\theta \mid D) = \Pi_{j=1\ldots m} P(X_1(j), \ldots, X_n(j) \mid \theta)$ from i.i.d.

$= \Pi_{j=1\ldots m} \Pi_{i=1\ldots n} P(X_i(j) \mid Parents(X_i(j)), \theta)$ factorization

$= \Pi_{i=1\ldots n} \Pi_{j=1\ldots m} P(X_i(j) \mid Parents(X_i(j)), \theta_i)$ *simplification*

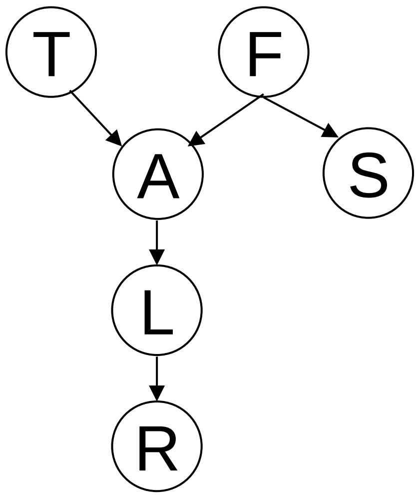
$= \Pi_{i=1\ldots n} L(\theta_i \mid D)$

- The likelihood function **decomposes** according to the **structure of the network**, which creates independent estimation problems.

# Coin tossing revisited

- Suppose you observed 3 coin tosses, and all come up with tails.
- What is the maximum predictor for $\theta$ ?
- Is this a good prediction?

# A problem: Zero probabilities




- For problems with lots of variables, it is possible that not all possible values are seen in the data.
  - Especially for very rare events.

- What is the MLE for the corresponding parameters?

  E.g. Prob(Heads) after seeing Tails times.

- **If a value is not seen, the corresponding MLE value for its parameters is 0.**

# Laplace smoothing

- Instead of: $\theta = N(H) / ( N(H) + N(T) )$
- Use: $\theta = ( N(H) + 1 ) / ( N(H) + N(T) + 2 )$

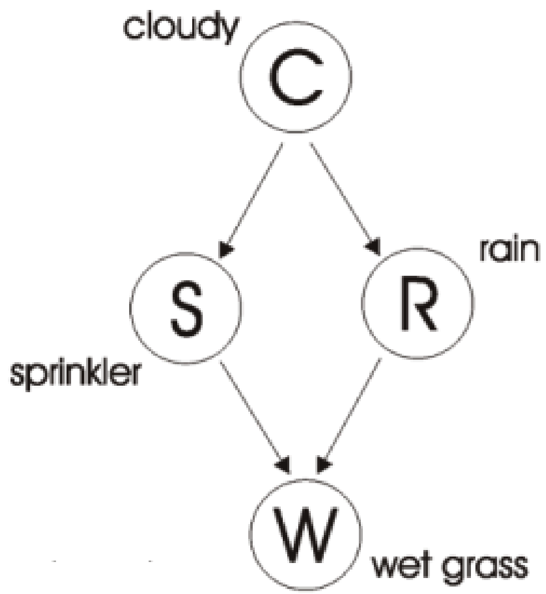
- Imagine that you have seen at least 1 instance of <u>each type</u>.
  - If you have no data, this estimate is: $\theta = 0.5$
  - With 3 tails, the estimate is: $\theta = 0.2$
  - With 98 tails, the estimate is: $\theta = 0.01$

*+1 for Heads*
*+1 for Tails*

- If $\theta$ is not a Bernoulli, the "+2" changes, e.g. for a categorical with *k* possible outcomes, add *+k* in the denominator (and +1 for each possible outcome).

# Exercise

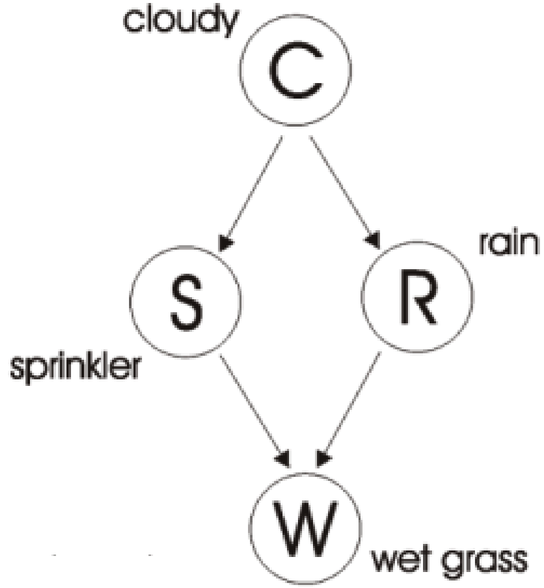
| C | S | R | W |
|---|---|---|---|
| T | T | F | T |
| T | T | T | T |
| T | F | F | F |
| F | T | F | T |
| F | F | F | F |




What is the MLE of this Bayes net?
The estimates after Laplace smoothing?

# Answers: MLE

| C | S | R | W |
|---|---|---|---|
| T | T | F | T |
| T | T | T | T |
| T | F | F | F |
| F | T | F | T |
| F | F | F | F |




**MLE:**

$P(C) = 3/5$

$P(S|C) = 2/3$ $P(S|\sim C) = 1/2$

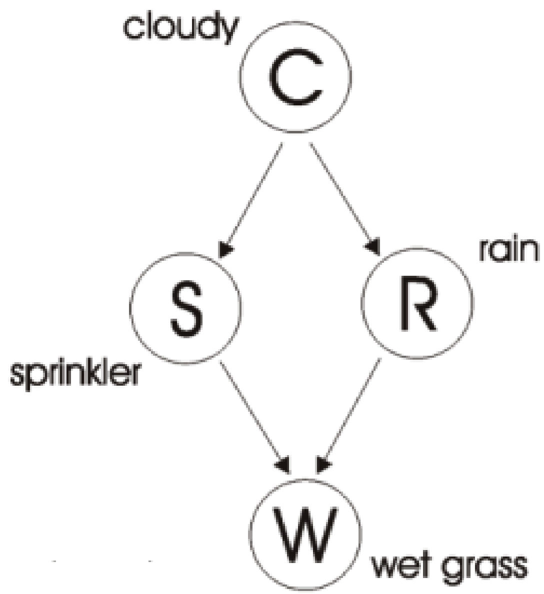
$P(R|C) = 1/3$ $P(R|\sim C) = 0/2$

$P(W|S,R) = 1/1$ $P(W|S,\sim R) = 2/2$

$P(W|\sim S,R) = 0/0$ $P(W|\sim S,\sim R) = 0/2$

# Answers: Laplace Smoothing

| C | S | R | W |
|---|---|---|---|
| T | T | F | T |
| T | T | T | T |
| T | F | F | F |
| F | T | F | T |
| F | F | F | F |




**Laplace:**

P(C) = 4/7

P(S|C) = 3/5   P(S|~C) = 2/4

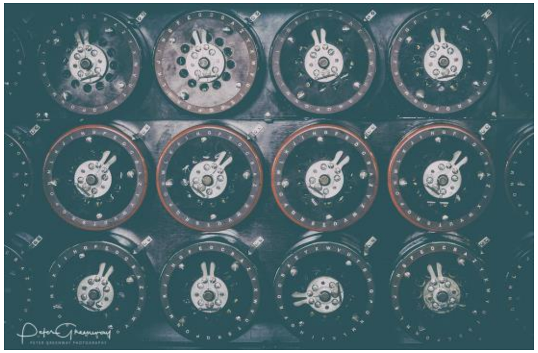
P(R|C) = 2/5   P(R|~C) = 1/4

P(W|S,R) = 2/3   P(W|S,~R) = 3/4

P(W|~S,R) = 1/2 P(W|~S,~R) = 1/4

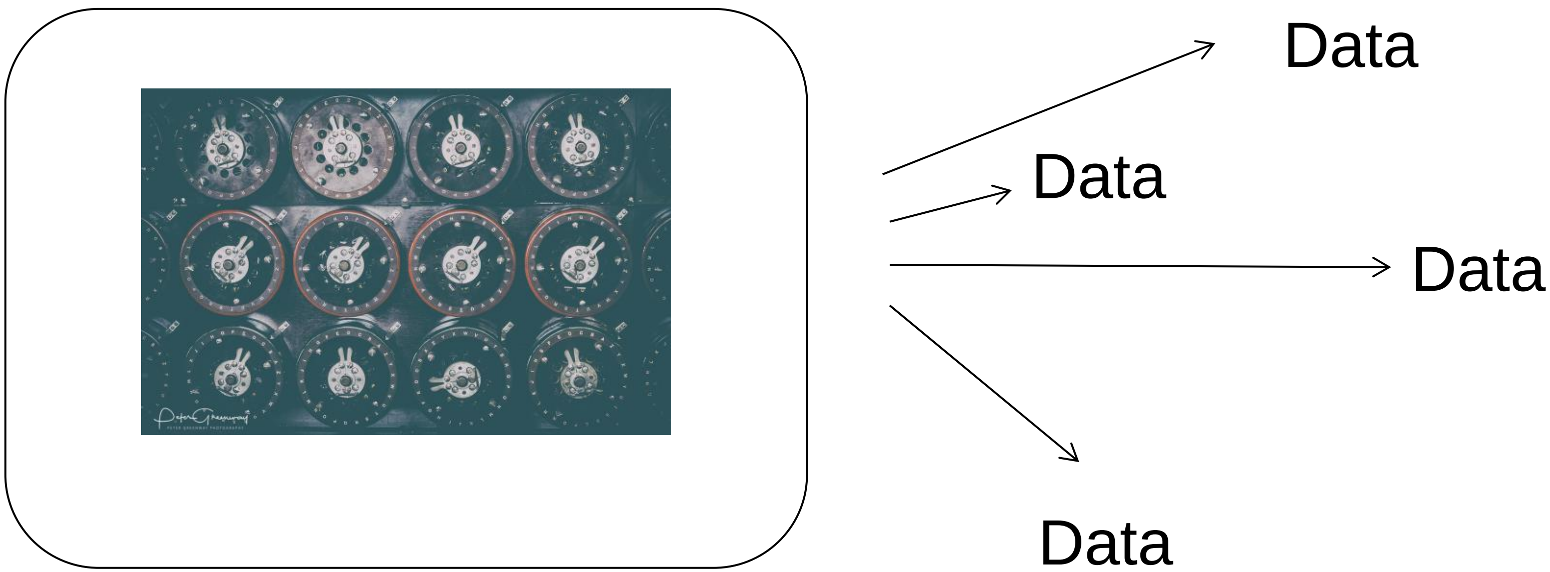
# An Intuitive Analogy

- A Bayes net structure is like having a machine with many dials, which correspond to its parameters



# Learning

- We know that this machine was used to generate some observed samples, but we don't know what the dial settings were. Learning is to figure out the setting.

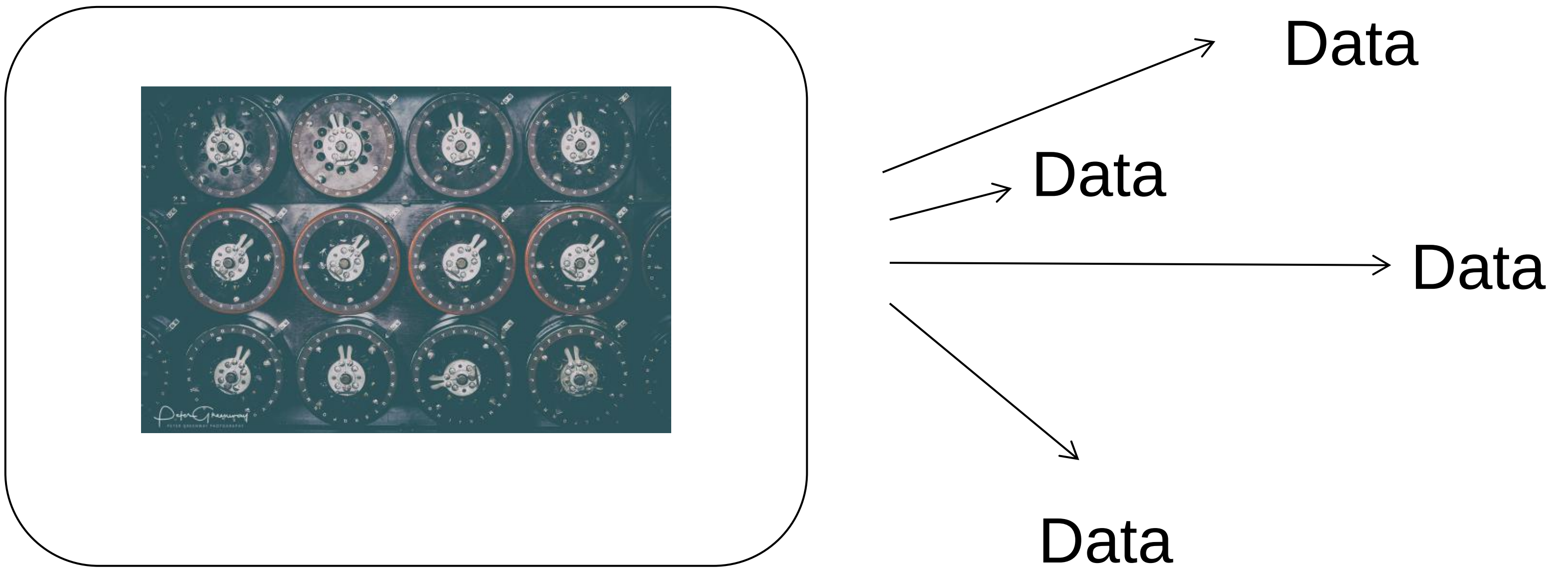


# MLE vs Laplace

- MLE and Laplace are two alternative criteria to select the dial setting.

  **MLE**: pick $\theta$ to maximize $P(D|\theta)$

  **Laplace**: also care about generalization

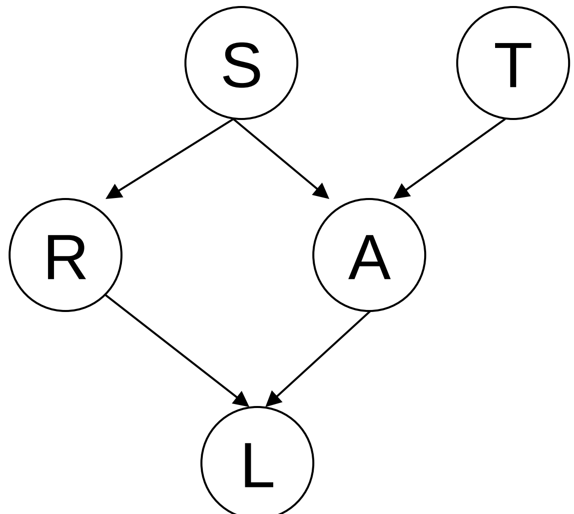


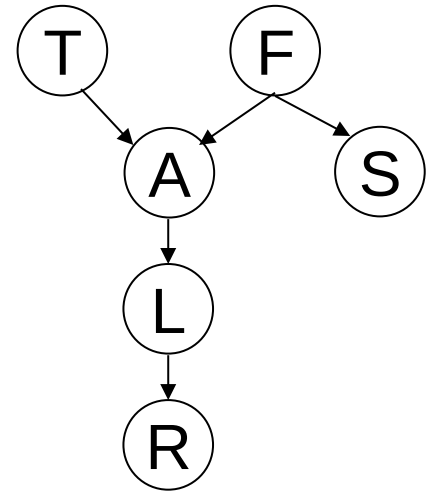
# Summary of learning in Bayes Nets

- **Learning** is process of acquiring a model from a data set.
- In Bayes nets, we can learn the parameters of the networks (numbers in the CPDs) or its structure.
- The maximum likelihood principle says that parameters should make the observed data as likely as possible.
- Parameters can be found by taking the gradient of the likelihood function and setting it to 0.
- In the case of simple distributions, the solution is to use “empirical probabilities”, based on counting the data.

# What have we left out?

- Everything about choosing variables and structure learning!



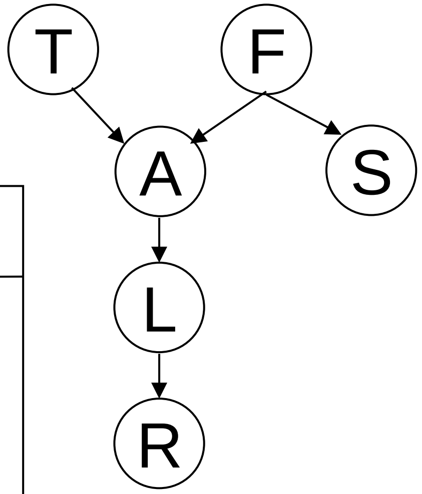

- Search over model structures (i.e. adding, reversing, deleting arcs) to maximize $L(\theta_S, S \mid D) = P(D \mid \theta_S, S)$
- Need to trade-off between model complexity and data fidelity. Hard!

# Summary of parameter estimation

- I.i.d. assumption
- Sufficient statistics
- Computing the maximum likelihood
- Laplace smoothing
- Extension to standard probability distributions (see textbook):
  - e.g., categorical, Gaussian, Poisson, exponential

# Learning in Bayesian networks

- Given data in the form of instances:

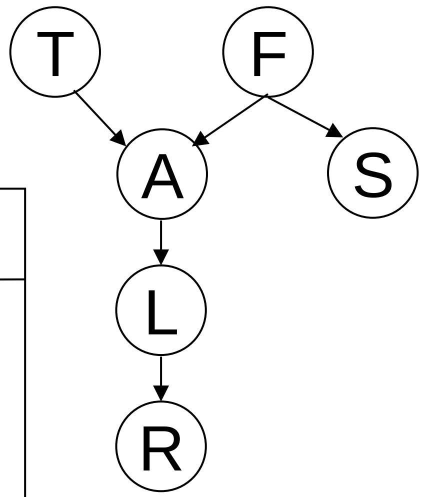



| Tampering | Fire | Smoke | Alarm | Leaving | Report |
|---|---|---|---|---|---|
| No | No | No | No | No | No |
| No | Yes | Yes | Yes | Yes | No |
| … | … | | … | … | … |

- Goal: Find parameters of the Bayes net.
- We discussed how to do this using maximum likelihood.

# Learning in Bayesian networks

- **Plot twist**: Suppose some values are missing!




| Tampering | Fire | Smoke | Alarm | Leaving | Report |
|---|---|---|---|---|---|
| ? | No | No | No | No | No |
| No | Yes | Yes | Yes | ? | No |
| … | … | | … | … | … |

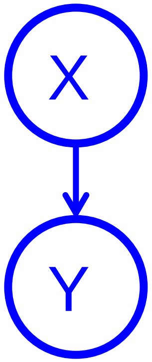
- Can we still use MLE?
  - How do we deal with the missing data?

# Why do we get incomplete data?

- Some variables may not be assigned values in *some* instances.
  - E.g. not all patients undergo all medical tests.
- Some variables may not be observed in *any* of the data items.
  - E.g. viewer preferences for a show may depend on their metabolic cycle (what time they are awake) - which is not usually measured.
- **Problem**: the fact that a value is missing may be indicative of what the value actually is.
  - E.g. patient did not undergo X-ray because she had no bone problems; so X-ray would likely have come out negative.

# Why missing values make life hard

- Consider a simple network $X \rightarrow Y$, and suppose we want to learn its parameters from samples $<x_1, y_1>, \ldots, <x_m, y_m>$.




- Which parameters do we need?

- Given all samples values, maximize log-likelihood of:

$L(\theta_X, \theta_{Y|X=0}, \theta_{Y|X=1}) = (\theta_X)^{N1} (1-\theta_X)^{N0} (\theta_{Y|X=0})^{N01} (1-\theta_{Y|X=0})^{N00} (\theta_{Y|X=1})^{N11} (1-\theta_{Y|X=1})^{N10}$

- Suppose now that $x_1$ is missing and $y_1=1$. What can we do?

# Why missing values make life hard (2)

- We can consider both settings: $x_1=0$, $x_1=1$.
- For each setting we get a different likelihood.
- Overall likelihood combines both settings (weighted by probability of that setting).

$$L(\theta_X, \theta_{Y|X=0}, \theta_{Y|X=1}) = (1- \theta_X)\, Pr\, (<0,y_1>, <x_2,y_2>, \ldots, <x_m,y_m> \mid \theta_X, \theta_{Y|X=0}, \theta_{Y|X=1}) + \theta_X\, Pr\, (<1,y_1>, <x_2,y_2>, \ldots, <x_m,y_m> \mid \theta_X, \theta_{Y|X=0}, \theta_{Y|X=1})$$

- **Problem**: If we have values missing for $x_1$ and $x_2$, we have to consider all possible values for both instances! Etc.

# Missing at random assumption

- The probability that the value of $X_i$ is missing is independent of its actual value, given the observed data.

- If this is not true, for variable $X_i$, we can introduce an additional Boolean variable, $X_i^{Observed}$ and satisfy the assumption.

# Effects of missing data

| Complete data | Missing data |
| --- | --- |
| • Parameters of model can be estimated locally and independently.<br>• Log-likelihood has a unique maximum.<br>• Under certain assumptions, there is a nice closed-form solution for parameters. | • Parameters cannot be estimated independently.<br>• Many local maxima. Maximizing likelihood becomes non-linear optimization problem.<br>• No closed-form solution. |

# Two solutions for maximizing likelihood

1. Gradient ascent: Use hill-climbing search through the space of parameters, following the gradient of the likelihood with respect to the parameters.

# Gradient ascent

- **Basic idea**: Move parameters in the direction of the log-likelihood.
- Pros:
  - Flexible: allows different forms for the Conditional Prob. Distributions.
  - Easy to compute the gradient at any parameter setting.
  - Closely related to other learning methods (e.g., neural nets).
- Cons:
  - Solution needs to be projected on space of legal parameters (in our case, need to ensure that we get probability distributions.)
  - Sensitive to parameters (e.g., learning rate).
  - Slow!

# Two solutions for maximizing likelihood

1. Gradient ascent: hill-climbing search through the space of parameters, following the gradient of the likelihood with respect to the parameters.

2. Expectation maximization: use the current parameter settings to construct a local approximation of the likelihood which is "nice" and can be optimized easily.

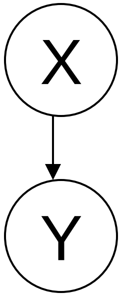
# Expectation Maximization (EM)

- General purpose method for learning from incomplete data (not only Bayes nets), **whenever an underlying distribution is assumed**.
- Main idea: Alternate between two steps
  1. (**E-step**): For all the instances of missing data, we will "fantasize" how the data should look based on the current parameter setting.
     - This means we compute **<u>expected sufficient statistics</u>**.
  2. (**M-step**): Then **<u>maximize parameter setting</u>**, based on these statistics.

# Outline of EM

- **Initialization**:
  - Start with some initial parameter setting (e.g. *P(T), P(F), P(A|F,T)*, etc.).
  - These can be estimated from all complete data instances.
- **Repeat**:
  1. <u>Expectation (E-step)</u>: Complete the data by assigning "values" to the missing items based on current parameter setting.
  2. <u>Maximization (M-step)</u>: Compute the maximum likelihood parameter setting based on the completed data. This is what we did earlier this lecture.
- **Convergence**:
  - Nothing changes in E-step or M-step between 2 consecutive rounds.

# EM in our example




- To start, guess the parameters of the network $\theta$ (using the known data):

  E.g. $\theta_X = N_{x=1}(2:m) / (m-1)$

  $\theta_{Y|x=0} = N_{Y=1,X=0}(2:m) / N_{X=0}(2:m)$

  $\theta_{Y|x=1} = N_{Y=1,X=1}(2:m) / N_{X=1}(2:m)$

- **E-step**: Using initial $\theta$, compute: $P(x_1=0 \mid y_1), P(x_1=1 \mid y_1)$

  (Note that this step requires *exact inference* - so not cheap!)

  Complete dataset with most likely value of $x_1$.

  Call new dataset D*.

- **M-step**: Compute new parameter vector $\theta$, which maximizes the likelihood given the completed data: $L(\theta|D^*) = P(D^* \mid \theta)$

  E.g. $\theta_X = N_{x=1}/m$

  $\theta_{Y|x=0} = N_{Y=1,X=0} / N_{X=0}$

  $\theta_{Y|x=1} = N_{Y=1,X=1} / N_{X=1}$

- **Repeat E-step and M-step until the parameter vector converges.**

# Two version of the algorithm

- **Hard EM**: for each missing data point, assign the value that is most likely.

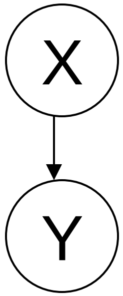
  (This is the version we just saw.)

- **Soft EM**: for each missing data point, put a weight on each value, equal to its probability, and use the weights as counts.

  (This is the most common version.)

Then these numbers are used as real counts, to provide a maximum likelihood estimate for $\theta$.

# Soft EM in our example




- To start, guess the parameters of the network $\theta$ (using the known data):

  E.g. $\theta_X = N_{x=1}(2{:}m) / (m-1)$

  $\theta_{Y|x=0} = N_{Y=1,X=0}(2{:}m) / N_{X=0}(2{:}m)$

  $\theta_{Y|x=1} = N_{Y=1,X=1}(2{:}m) / N_{X=1}(2{:}m)$

- **E-step**: Using initial $\theta$, compute: $w_0 = P(x_1=0 \mid y_1)$ $w_1 = P(x_1=1 \mid y_1)$

  Now hypothesize two datasets: $D_0 = \langle w_0, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_m, y_m\rangle$

  $D_1 = \langle w_1, y_1\rangle, \langle x_2, y_2\rangle, \ldots, \langle x_m, y_m\rangle$

- **M-step**: Compute new parameter vector $\theta$, which maximizes the expected likelihood given the completed data: $L(\theta|D^*) = w_0 P(D_0 \mid \theta) + w_1 P(D_1 \mid \theta)$

  E.g. $\theta_X = (N_{X=1}(2{:}m) + w_1) / m$

  $\theta_{Y|x=0} = (N_{Y=1,X=0}(2{:}m) + w_0) / (N_{X=0}(2{:}m) + w_0)$

  $\theta_{Y|x=1} = (N_{Y=1,X=1}(2{:}m) + w_1) / (N_{X=1}(2{:}m) + w_1)$

  **Repeat E and M steps!**

# Comparison of hard EM and soft EM

- Soft EM does not commit to specific value for the missing item.
  - Instead, it considers all possible values, with some probability.
  - This is a pleasing property, given the uncertainty in the value.
- Complexity:
  - Hard EM requires computing most probable values.
  - Soft EM requires computing conditional probabilities for completing the missing values.
  - Same complexity: both require full probabilistic inference - which can be expensive!

# Properties of EM

- Likelihood function is guaranteed to improve (or stay the same) with each iteration.
  - Algorithm can be stopped when no more improvement is achieved between iterations.
- EM is guaranteed to converge to a local optimum of the likelihood function.
  - Starting with different values of initial parameters is necessary (random re-starts, to avoid local optimum).
- EM is a widely used algorithm in practice!

# A harder example

Suppose we have the simple Bayes net $A \to B \to C$, where each node is associated with a Bernoulli random variable. Further suppose we have the following sample data:

(i) A=1,B=?,C=1

(ii) A=0,B=1,C=0

(iii) A=1,B=0,C=0

(iv) A=1,B=1,C=0

(v) A=1,B=1,C=0

(vi) A=0,B=0,C=?

# A harder example

Suppose we have the simple Bayes net $A \rightarrow B \rightarrow C$, where each node is associated with a Bernoulli random variable. Further suppose we have the following sample data:

(i) A=1,B=?,C=1

(ii) A=0,B=1,C=0

(iii) A=1,B=0,C=0

(iv) A=1,B=1,C=0

(v) A=1,B=1,C=0

(vi) A=0,B=0,C=?

**E-step**

$$
\begin{aligned}
w_{B_1=1} &= P(B_1 = 1 | A_1, C_1) \\
&= P(B = 1 | A = 1, C = 1) \\
&= \frac{P(A = 1, B = 1, C = 1)}{P(A = 1, C = 1)} \\
&= \frac{P(A = 1, B = 1, C = 1)}{\sum_{b \in 0,1} P(A = 1, C = 1, B = b)} \\
&= \frac{\theta_A \theta_{B|A=1} \theta_{C|B=1}}{\theta_A \theta_{B|A=1} \theta_{C|B=1} + \theta_A (1 - \theta_{B|A=1}) \theta_{C|B=0}} \\
&= \frac{(0.5)(0.5)(0.5)}{(0.5)(0.5)(0.5) + (0.5)(0.5)(0.5)} \\
&= 0.5
\end{aligned}
$$

$$
\begin{aligned}
w_{B_1=0} &= P(B_1 = 0 | A_1, C_1) \\
&= P(B = 0 | A = 1, C = 1) \\
&= (1 - P(B = 1 | A = 1, C = 1)) \\
&= (1 - w_{B_1=1}) \\
&= 0.5
\end{aligned}
$$

$$
\begin{aligned}
w_{C_6=1} &= P(C_6 = 1 | A_6, B_6) \\
&= P(C_6 = 1 | B_6) \qquad \text{by conditional independence} \\
&= P(C = 1 | B = 0) \\
&= 0.5
\end{aligned}
$$

$$
\begin{aligned}
w_{C_6=0} &= (1 - w_{C_6=1}) \qquad \text{same reasoning as } w_{B_1=0} = (1 - w_{B_1=1}) \\
&= 0.5
\end{aligned}
$$

**M-step**

$$\theta_A^{ML} = \frac{N_{A=1}(1{:}6)}{6} = \frac{4}{6} \approx 0.667$$

$$\theta_{B|A=1}^{ML} = \frac{N_{B=1|A=1)}(2{:}6) + w_{B_1=1}}{4} = \frac{2+0.5}{4} = 0.625$$

$$\theta_{B|A=0}^{ML} = \frac{N_{B=1|A=1)}(2{:}6)}{2} = \frac{1}{2} = 0.5$$

$$\theta_{C|B=1}^{ML} = \frac{N_{C=1|B=1}(2{:}4) + w_{B_1=1}}{3 + w_{B_1=1}} = \frac{0.5}{3.5} \approx 0.143$$

$$\theta_{C|B=0}^{ML} = \frac{N_{C=1|B=0}(2{:}4) + w_{B_1=0} + w_{C_6=1}}{2 + w_{B_1=0}} = \frac{0.5+0.5}{2.5} = 0.4$$

And repeat…