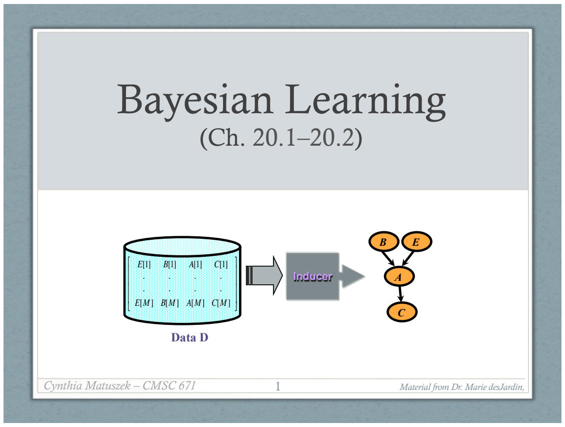
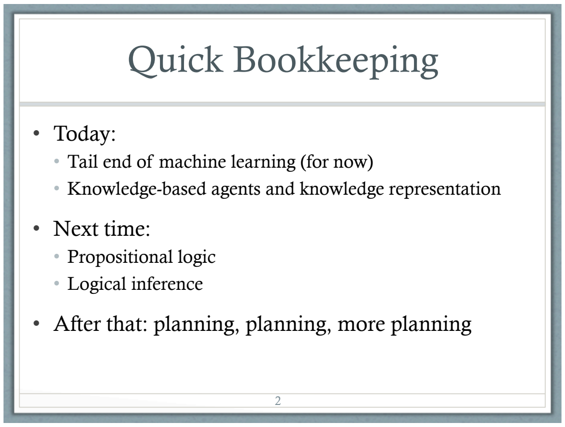


Bayesian Learning

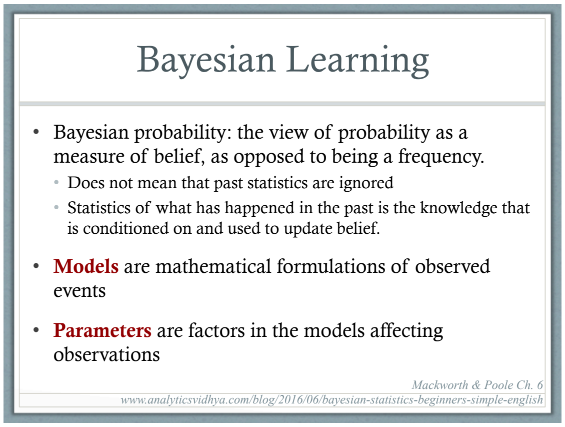
(Ch. 20.1–20.2)

Material from Dr. Marie desJardin,



Quick Bookkeeping

- Today:
  - Tail end of machine learning (for now)
  - Knowledge-based agents and knowledge representation
- Next time:
  - Propositional logic
  - Logical inference
- After that: planning, planning, more planning

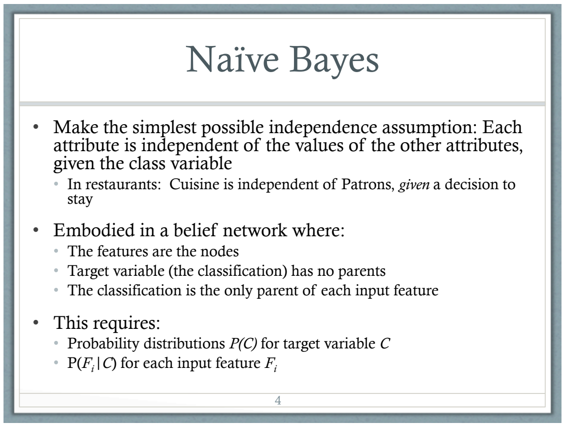


Bayesian Learning

- Bayesian probability: the view of probability as a measure of belief, as opposed to being a frequency.
  - Does not mean that past statistics are ignored
  - Statistics of what has happened in the past is the knowledge that is conditioned on and used to update belief.
- **Models** are mathematical formulations of observed events
- **Parameters** are factors in the models affecting observations

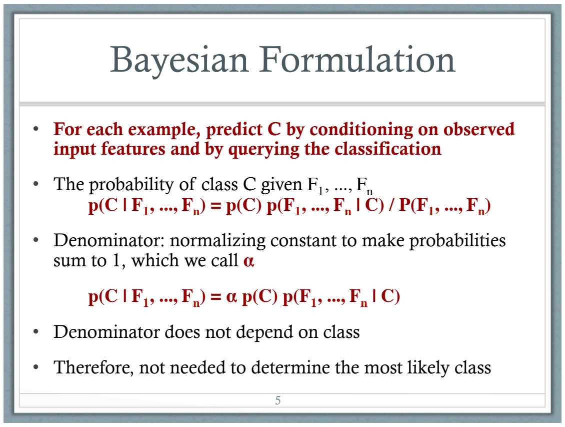
Mackworth & Poole Ch. 6

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english



Naïve Bayes

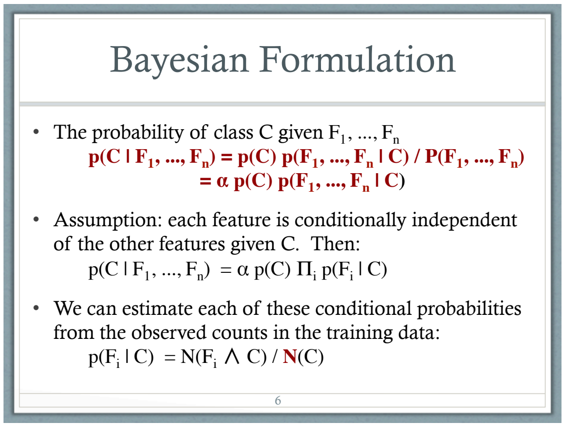
- Make the simplest possible independence assumption: Each attribute is independent of the values of the other attributes, given the class variable
  - In restaurants: Cuisine is independent of Patrons, *given* a decision to stay
- Embodied in a belief network where:
  - The features are the nodes
  - Target variable (the classification) has no parents
  - The classification is the only parent of each input feature
- This requires:
  - Probability distributions $P(C)$ for target variable C
  - $P(F_i \mid C)$ for each input feature F_i



Bayesian Formulation

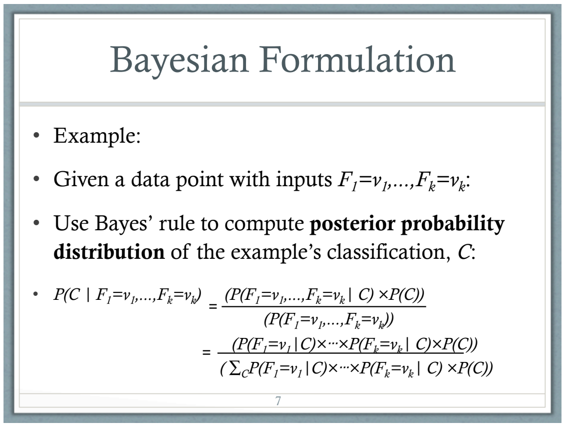
- **For each example, predict C by conditioning on observed input features and by querying the classification**
- The probability of class C given $F_1, \ldots, F_n$
  $\mathbf{p(C \mid F_1, \ldots, F_n) = p(C)\, p(F_1, \ldots, F_n \mid C) / P(F_1, \ldots, F_n)}$
- Denominator: normalizing constant to make probabilities sum to 1, which we call $\boldsymbol{\alpha}$

  $\mathbf{p(C \mid F_1, \ldots, F_n) = \alpha\, p(C)\, p(F_1, \ldots, F_n \mid C)}$
- Denominator does not depend on class
- Therefore, not needed to determine the most likely class



Bayesian Formulation

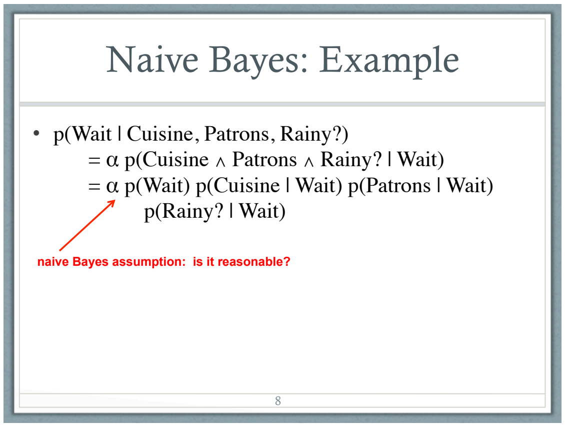
- The probability of class C given $F_1, \ldots, F_n$
  $\mathbf{p(C \mid F_1, \ldots, F_n) = p(C)\, p(F_1, \ldots, F_n \mid C) / P(F_1, \ldots, F_n)}$
  $\mathbf{= \alpha\, p(C)\, p(F_1, \ldots, F_n \mid C)}$
- Assumption: each feature is conditionally independent of the other features given C. Then:
  $p(C \mid F_1, \ldots, F_n) = \alpha\, p(C)\, \Pi_i\, p(F_i \mid C)$
- We can estimate each of these conditional probabilities from the observed counts in the training data:
  $p(F_i \mid C) = N(F_i \land C) / \mathbf{N}(C)$



Bayesian Formulation

- Example:
- Given a data point with inputs $F_1=v_1,\ldots,F_k=v_k$:
- Use Bayes' rule to compute **posterior probability distribution** of the example's classification, C:
- $P(C \mid F_1=v_1,\ldots,F_k=v_k) = \dfrac{(P(F_1=v_1,\ldots,F_k=v_k \mid C) \times P(C))}{(P(F_1=v_1,\ldots,F_k=v_k))}$

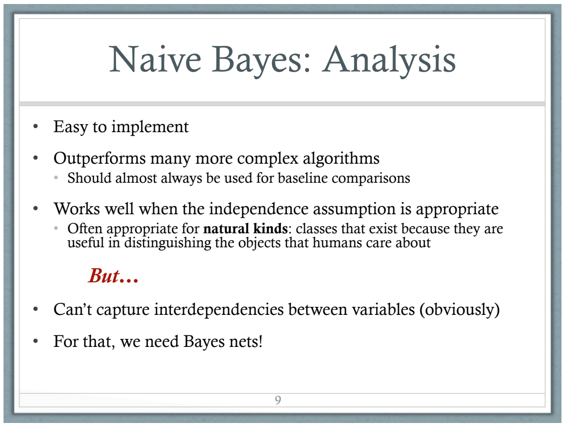
$$= \frac{(P(F_1=v_1 \mid C)\times\cdots\times P(F_k=v_k \mid C)\times P(C))}{(\sum_C P(F_1=v_1 \mid C)\times\cdots\times P(F_k=v_k \mid C) \times P(C))}$$



Naive Bayes: Example

- p(Wait | Cuisine, Patrons, Rainy?)
  = α p(Cuisine ∧ Patrons ∧ Rainy? | Wait)
  = α p(Wait) p(Cuisine | Wait) p(Patrons | Wait)
  p(Rainy? | Wait)

naive Bayes assumption: is it reasonable?

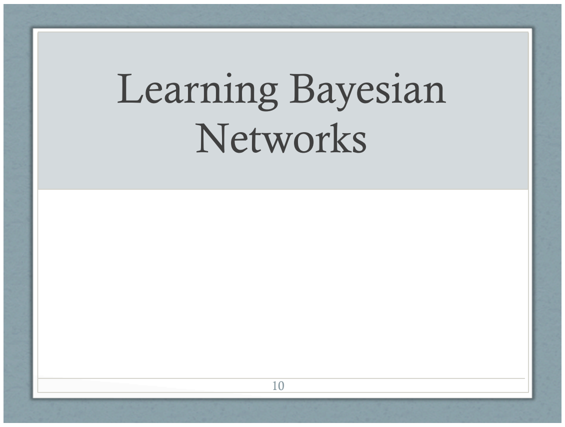


Naive Bayes: Analysis

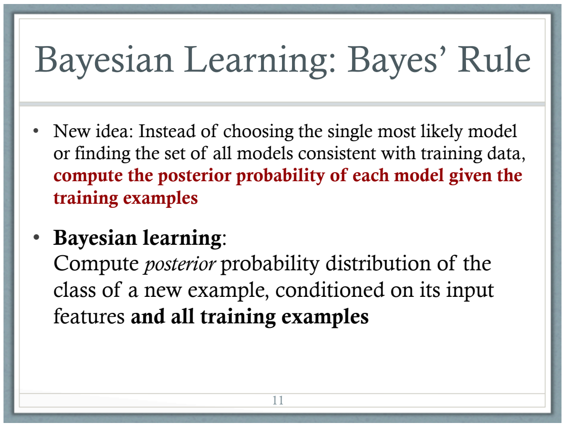
- Easy to implement
- Outperforms many more complex algorithms
  - Should almost always be used for baseline comparisons
- Works well when the independence assumption is appropriate
  - Often appropriate for **natural kinds**: classes that exist because they are useful in distinguishing the objects that humans care about

But…

- Can't capture interdependencies between variables (obviously)
- For that, we need Bayes nets!

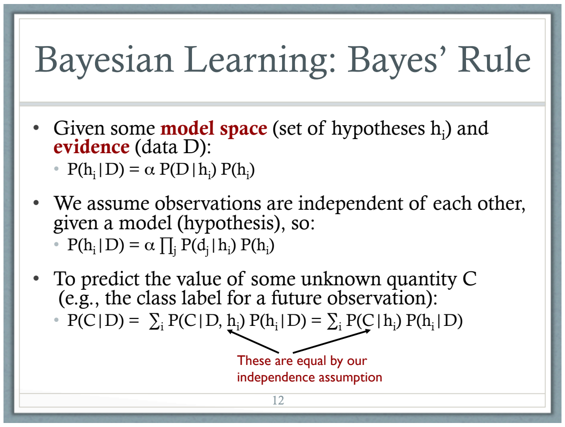


Learning Bayesian Networks



Bayesian Learning: Bayes' Rule

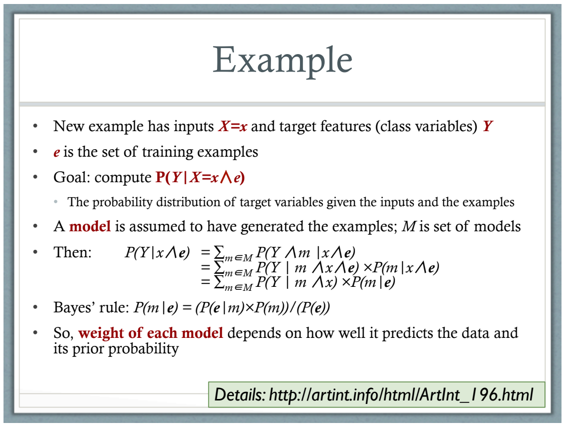
- New idea: Instead of choosing the single most likely model or finding the set of all models consistent with training data, **compute the posterior probability of each model given the training examples**
- **Bayesian learning**:
  Compute *posterior* probability distribution of the class of a new example, conditioned on its input features **and all training examples**



Bayesian Learning: Bayes' Rule

- Given some **model space** (set of hypotheses h_i) and **evidence** (data D):
  - $P(h_i \mid D) = \alpha\, P(D \mid h_i)\, P(h_i)$
- We assume observations are independent of each other, given a model (hypothesis), so:
  - $P(h_i \mid D) = \alpha \prod_j P(d_j \mid h_i)\, P(h_i)$
- To predict the value of some unknown quantity C (e.g., the class label for a future observation):
  - $P(C \mid D) = \sum_i P(C \mid D, h_i)\, P(h_i \mid D) = \sum_i P(C \mid h_i)\, P(h_i \mid D)$

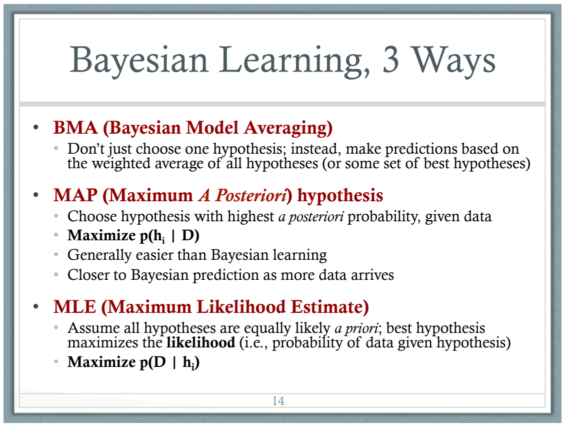
These are equal by our independence assumption



Example

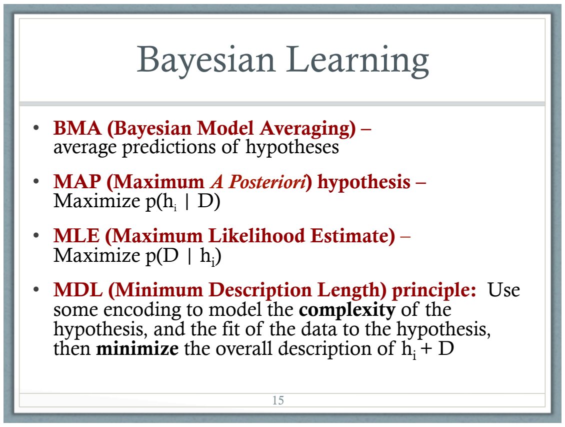
- New example has inputs ***X*=*x*** and target features (class variables) ***Y***
- ***e*** is the set of training examples
- Goal: compute **P(*Y*|*X*=*x*∧*e*)**
  - The probability distribution of target variables given the inputs and the examples
- A **model** is assumed to have generated the examples; *M* is set of models
- Then: $$\begin{aligned} P(Y \mid x \wedge \mathbf{e}) &= \textstyle\sum_{m \in M} P(Y \wedge m \mid x \wedge \mathbf{e}) \\ &= \textstyle\sum_{m \in M} P(Y \mid m \wedge x \wedge \mathbf{e}) \times P(m \mid x \wedge \mathbf{e}) \\ &= \textstyle\sum_{m \in M} P(Y \mid m \wedge x) \times P(m \mid \mathbf{e}) \end{aligned}$$
- Bayes' rule: $P(m \mid \mathbf{e}) = (P(\mathbf{e} \mid m) \times P(m)) / (P(\mathbf{e}))$
- So, **weight of each model** depends on how well it predicts the data and its prior probability

Details: http://artint.info/html/ArtInt_196.html



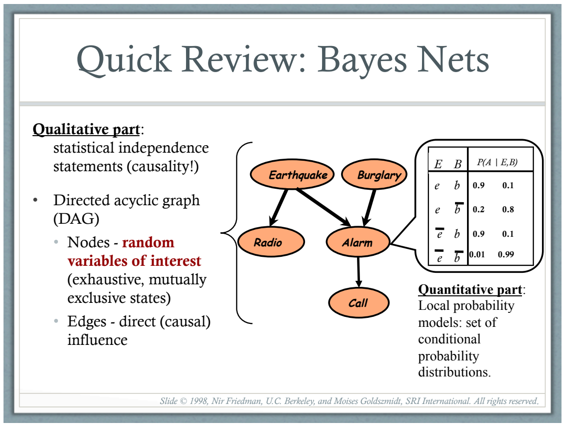
Bayesian Learning, 3 Ways

- **BMA (Bayesian Model Averaging)**
  - Don't just choose one hypothesis; instead, make predictions based on the weighted average of all hypotheses (or some set of best hypotheses)
- **MAP (Maximum *A Posteriori*) hypothesis**
  - Choose hypothesis with highest *a posteriori* probability, given data
  - **Maximize p(h_i | D)**
  - Generally easier than Bayesian learning
  - Closer to Bayesian prediction as more data arrives
- **MLE (Maximum Likelihood Estimate)**
  - Assume all hypotheses are equally likely *a priori*; best hypothesis maximizes the **likelihood** (i.e., probability of data given hypothesis)
  - **Maximize p(D | h_i)**



Bayesian Learning

- **BMA (Bayesian Model Averaging) –** average predictions of hypotheses
- **MAP (Maximum *A Posteriori*) hypothesis –** Maximize p(h_i | D)
- **MLE (Maximum Likelihood Estimate) –** Maximize p(D | h_i)
- **MDL (Minimum Description Length) principle:** Use some encoding to model the **complexity** of the hypothesis, and the fit of the data to the hypothesis, then **minimize** the overall description of h_i + D



Quick Review: Bayes Nets

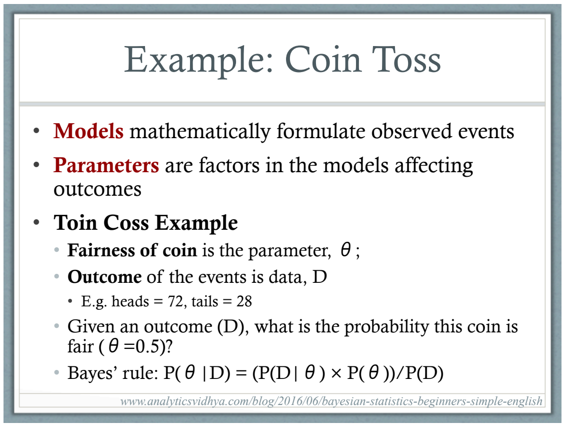
Qualitative part:
statistical independence statements (causality!)

- Directed acyclic graph (DAG)
  - Nodes - **random variables of interest** (exhaustive, mutually exclusive states)
  - Edges - direct (causal) influence

E	*B*	P(A | E,B)	
e	*b*	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	*b*	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

Quantitative part:
Local probability models: set of conditional probability distributions.

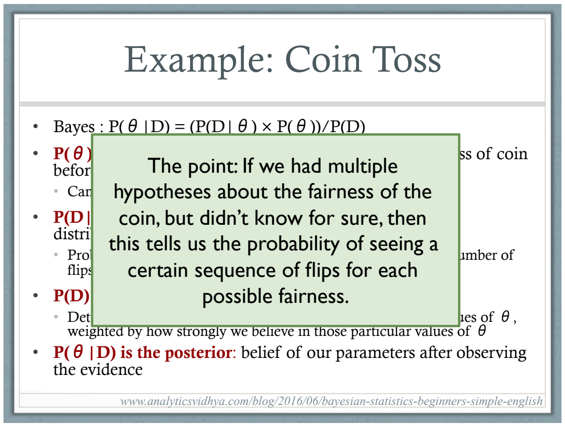
Slide © 1998, Nir Friedman, U.C. Berkeley, and Moises Goldszmidt, SRI International. All rights reserved.



Example: Coin Toss

- **Models** mathematically formulate observed events
- **Parameters** are factors in the models affecting outcomes
- **Toin Coss Example**
  - **Fairness of coin** is the parameter, θ ;
  - **Outcome** of the events is data, D
    - E.g. heads = 72, tails = 28
  - Given an outcome (D), what is the probability this coin is fair (θ =0.5)?
  - Bayes' rule: P(θ |D) = (P(D| θ) × P(θ))/P(D)

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

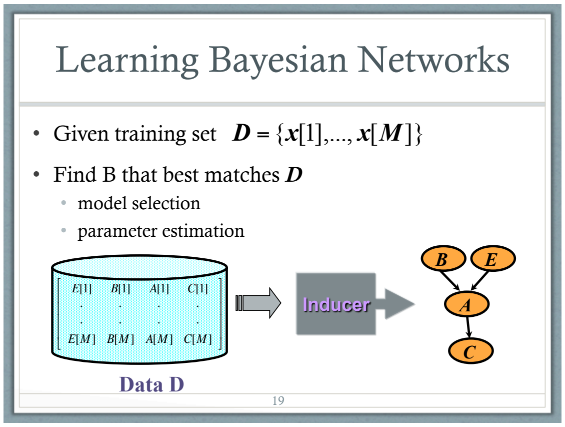


Example: Coin Toss

- Bayes : P(θ |D) = (P(D| θ) × P(θ))/P(D)
- **P(θ)** ss of coin befor
  - Can
- **P(D |** distri
  - Prol umber of flips
- **P(D)**
  - Det ues of θ , weighted by how strongly we believe in those particular values of θ
- **P(θ |D) is the posterior**: belief of our parameters after observing the evidence

The point: If we had multiple hypotheses about the fairness of the coin, but didn't know for sure, then this tells us the probability of seeing a certain sequence of flips for each possible fairness.

www.analyticsvidhya.com/blog/2016/06/bayesian-statistics-beginners-simple-english

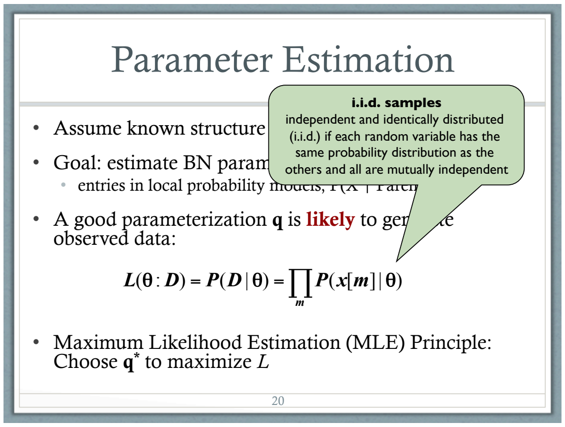


Learning Bayesian Networks

- Given training set $\boldsymbol{D} = \{\boldsymbol{x}[1], \ldots, \boldsymbol{x}[\boldsymbol{M}]\}$
- Find B that best matches $\boldsymbol{D}$
  - model selection
  - parameter estimation

$$\begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

Data D



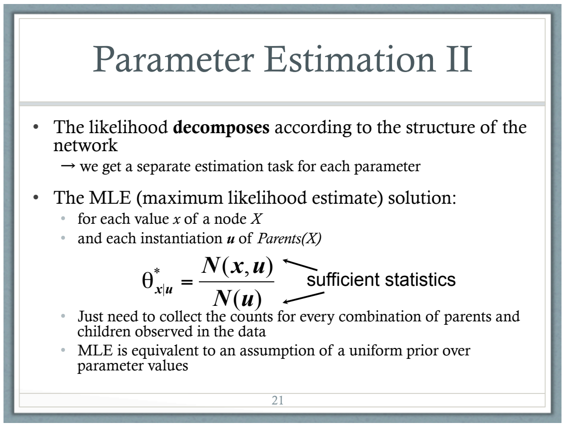
Parameter Estimation

- Assume known structure
- Goal: estimate BN param
  - entries in local probability models, P(X | Paren
- A good parameterization **q** is **likely** to ger te observed data:

$$L(\theta : D) = P(D \mid \theta) = \prod_m P(x[m] \mid \theta)$$

- Maximum Likelihood Estimation (MLE) Principle: Choose $\mathbf{q}^*$ to maximize L

i.i.d. samples
independent and identically distributed (i.i.d.) if each random variable has the same probability distribution as the others and all are mutually independent



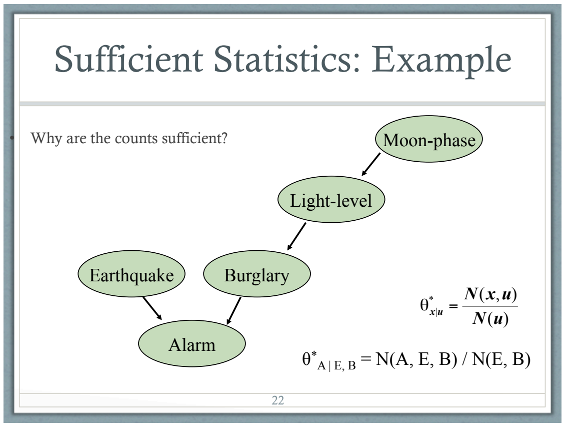
Parameter Estimation II

- The likelihood **decomposes** according to the structure of the network
  → we get a separate estimation task for each parameter
- The MLE (maximum likelihood estimate) solution:
  - for each value *x* of a node *X*
  - and each instantiation ***u*** of *Parents(X)*

$$\theta^*_{x|u} = \frac{N(x,u)}{N(u)}$$

sufficient statistics

  - Just need to collect the counts for every combination of parents and children observed in the data
  - MLE is equivalent to an assumption of a uniform prior over parameter values

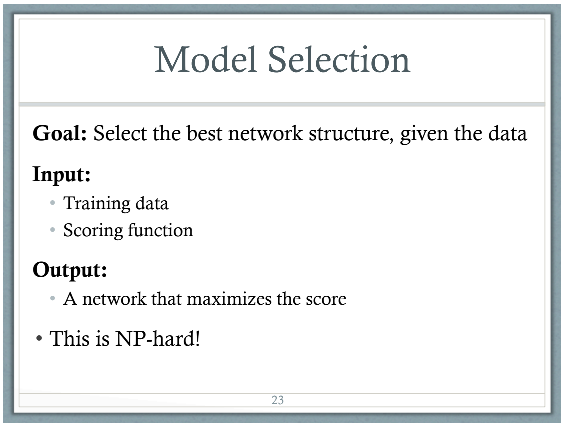


Sufficient Statistics: Example

Why are the counts sufficient?

$$\theta^*_{x|u} = \frac{N(x,u)}{N(u)}$$

$\theta^*_{A \mid E, B}$ = N(A, E, B) / N(E, B)



Model Selection

Goal: Select the best network structure, given the data

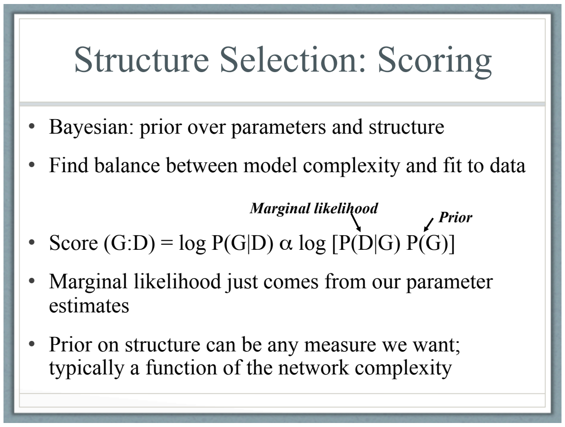
Input:

- Training data
- Scoring function

Output:

- A network that maximizes the score

- This is NP-hard!

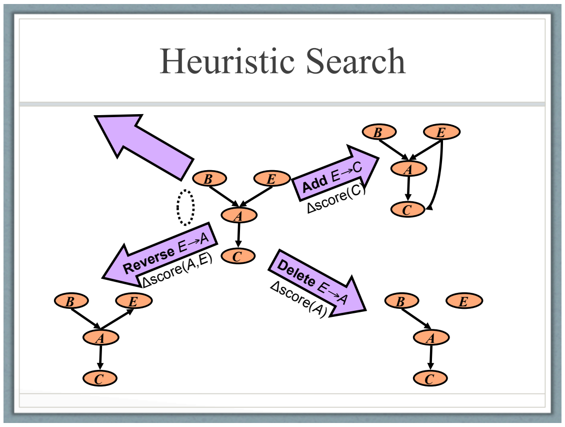


Structure Selection: Scoring

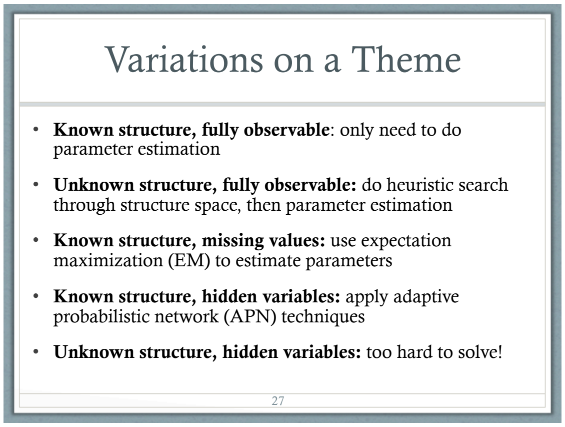
- Bayesian: prior over parameters and structure
- Find balance between model complexity and fit to data
- Score (G:D) = log P(G|D) α log [P(D|G) P(G)]

Marginal likelihood *Prior*

- Marginal likelihood just comes from our parameter estimates
- Prior on structure can be any measure we want; typically a function of the network complexity

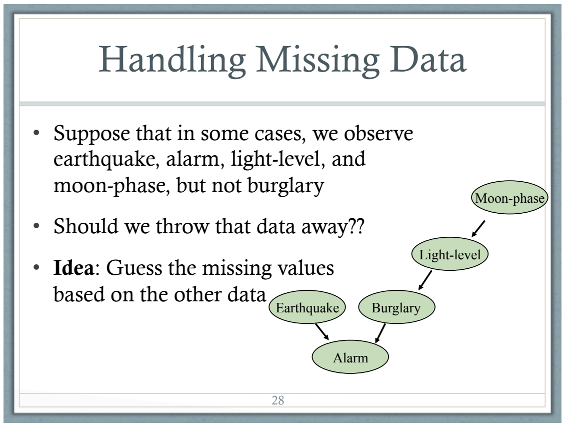


Heuristic Search



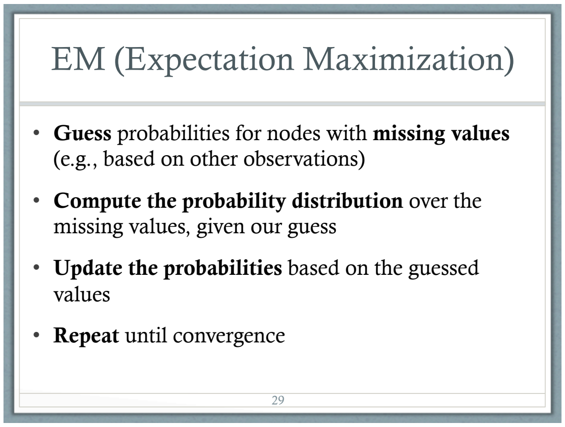
Variations on a Theme

- **Known structure, fully observable**: only need to do parameter estimation
- **Unknown structure, fully observable:** do heuristic search through structure space, then parameter estimation
- **Known structure, missing values:** use expectation maximization (EM) to estimate parameters
- **Known structure, hidden variables:** apply adaptive probabilistic network (APN) techniques
- **Unknown structure, hidden variables:** too hard to solve!



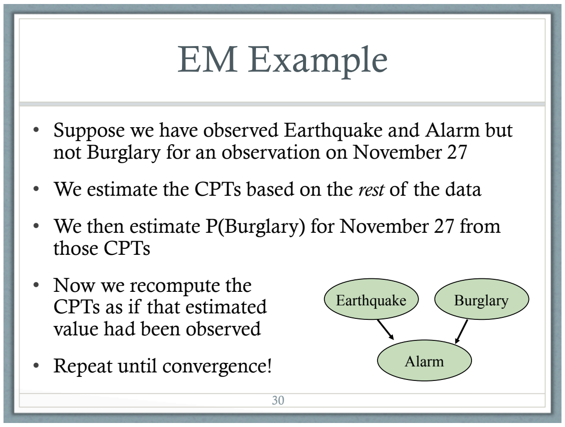
Handling Missing Data

- Suppose that in some cases, we observe earthquake, alarm, light-level, and moon-phase, but not burglary
- Should we throw that data away??
- **Idea**: Guess the missing values based on the other data



EM (Expectation Maximization)

- **Guess** probabilities for nodes with **missing values** (e.g., based on other observations)
- **Compute the probability distribution** over the missing values, given our guess
- **Update the probabilities** based on the guessed values
- **Repeat** until convergence



EM Example

- Suppose we have observed Earthquake and Alarm but not Burglary for an observation on November 27
- We estimate the CPTs based on the *rest* of the data
- We then estimate P(Burglary) for November 27 from those CPTs
- Now we recompute the CPTs as if that estimated value had been observed
- Repeat until convergence!