



Exploiting Structure in Probability Distributions

Irit Gat-Viks

Based on presentation and lecture notes of
Nir Friedman, Hebrew University

Basic Probability Definitions

- ◆ **Product Rule:** $P(A,B)=P(A | B)*P(B)= P(B | A)*P(A)$
- ◆ **Independence** between A and B: $P(A,B)=P(A)*P(B)$,
or alternatively: $P(A|B)=P(A)$, $P(B|A)=P(B)$.
- ◆ **Total probability theorem:** $\bigcup_{i=1}^n B_i = \Omega$, $\forall i \neq j \ B_i \cap B_j = \phi$

$$P(A) = \sum_{i=1}^n P(A, B_i) = \sum_{i=1}^n P(B_i) * P(A | B_i)$$

Basic Probability Definitions

◆ Bayes Rule:

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

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

◆ Chain Rule:

$$P(X_1, \dots, X_n) =$$

$$P(X_1 | X_2, \dots, X_n) \cdot P(X_2 | X_3, \dots, X_n) \cdot P(X_3 | X_4, \dots, X_n) \cdot \dots \cdot P(X_{n-1} | X_n) \cdot P(X_n)$$

Exploiting Independence Property

- ◆G: whether the woman is pregnant
- ◆D: whether the doctor's test is positive

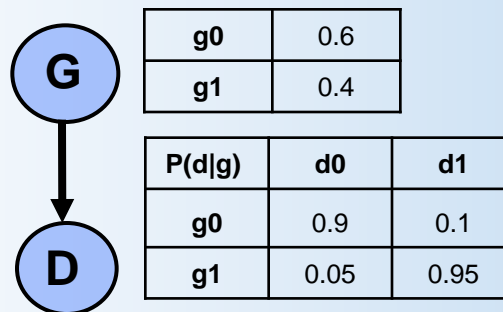
The joint distribution representation $P(g,d)$:

G	D	P(G,D)
0	0	0.54
0	1	0.06
1	0	0.02
1	1	0.38

Factorial representation

Using conditional probability: $P(g,d)=P(g)*P(d|g)$.

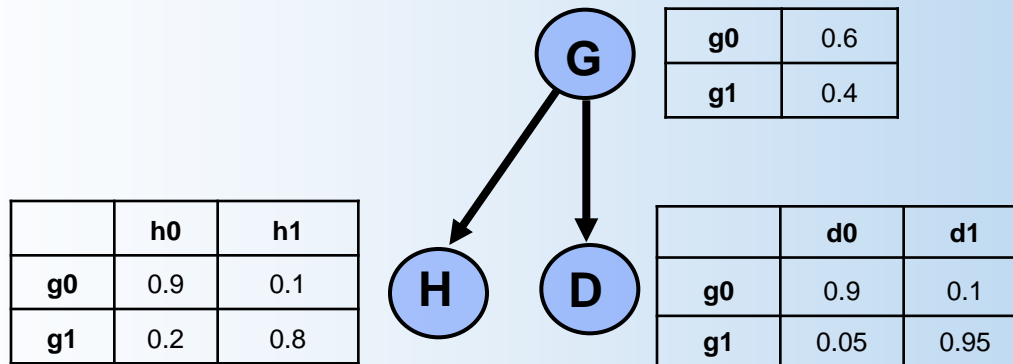
The distribution of $P(g)$, $P(d|g)$:



Example: $P(g_0,d_1)=0.06$ vs. $P(g_0)*P(d_1|g_0)=0.6*0.1=0.06$

Exploiting Independence Property

- ◆ H: home test
- ◆ Independence assumption: $\text{Ind}(H;D|G)$ (i.e., given G, H is independent of D).



Joint distribution

Factorial representation

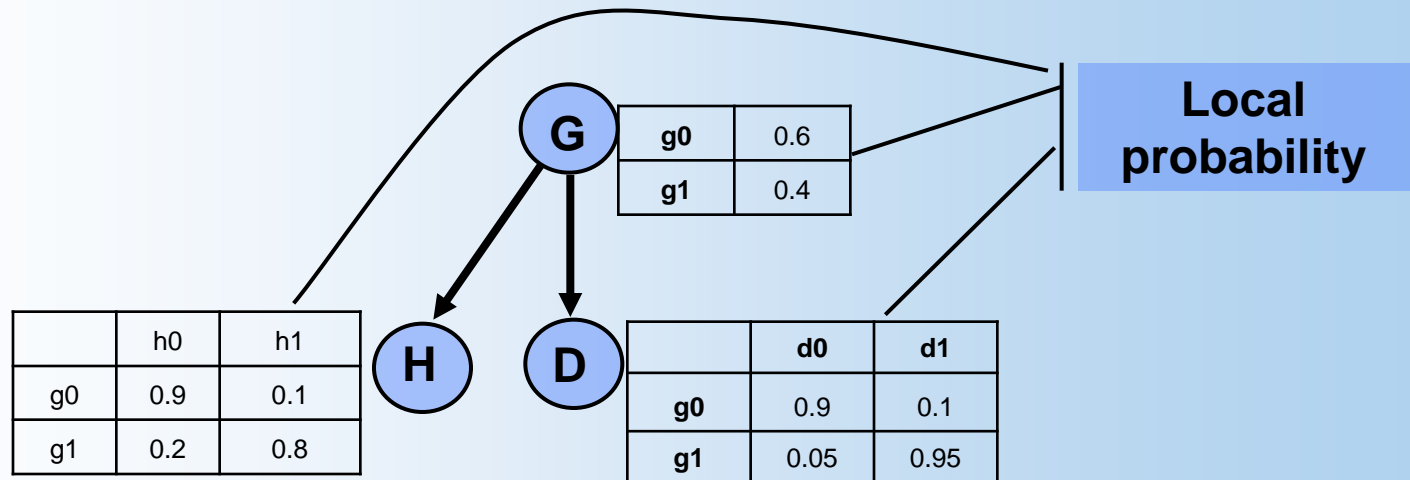
$$P(d,h,g) = P(d,h|g) * P(g) = P(d|g) * P(h|g) * P(g)$$

Product rule

$\text{Ind}(H;D|G)$

Exploiting Independence Property

	representation of $P(d,g,h)$	
	joint distribution	factored distribution
No. of parameters	7	5
Adding new variable H	changing the distribution entirely	Modularity: reuse the local probability model. (Only new local probability model for H.)

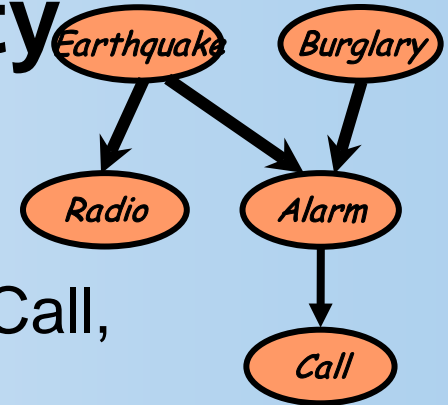


=> **Bayesian networks:** Exploiting independence properties of the distribution in order to allow a compact and natural representation.

Outline

- ◆ Introduction
- ◆ Bayesian Networks
 - » Representation & Semantics
 - Inference in Bayesian networks
 - Learning Bayesian networks

Representing the Uncertainty



- ◆ A story with five random variables:
 - Burglary, Earthquake, Alarm, Neighbor Call, Radio Announcement
 - Specify joint distribution with $2^5=32$ parameters

maybe...

- ◆ An expert system for monitoring intensive care patients
 - Specify joint distribution over 37 variables with (at least) 2^{37} parameters

no way!!!

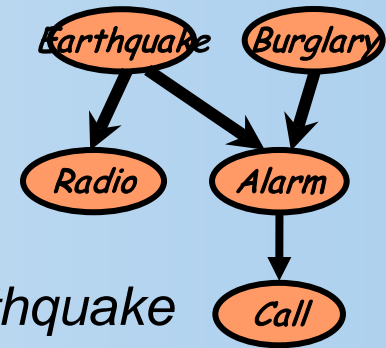
Probabilistic Independence: a Key for Representation and Reasoning

- ◆ Recall that if X and Y are **independent** given Z then

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

- ◆ In our story...if

- *burglary* and *earthquake* are **independent**
- *alarm sound* and *radio* are **independent** given *earthquake*



- ◆ then instead of 15 parameters we need 8

$$P(A, R, E, B) = P(A | R, E, B) \cdot P(R | E, B) \cdot P(E | B) \cdot P(B)$$

versus

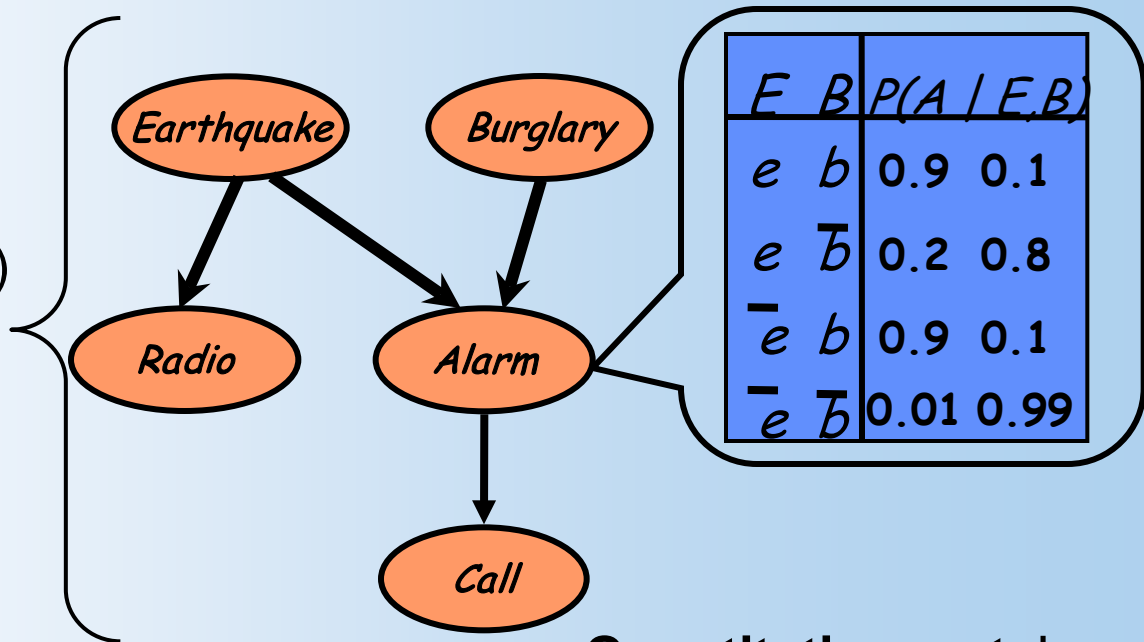
$$P(A, R, E, B) = P(A | E, B) \cdot P(R | E) \cdot P(E) \cdot P(B)$$

Need a language to represent independence statements

Bayesian networks

Efficient representation of probability distributions via conditional independence

- Qualitative part:** statistical independence statements
- Directed acyclic graph (DAG)
- ◆ Nodes - random variables of interest (exhaustive and mutually exclusive states)
 - ◆ Edges - direct influence



- ◆ **Quantitative part:** Local probability models. Set of conditional probability distributions.

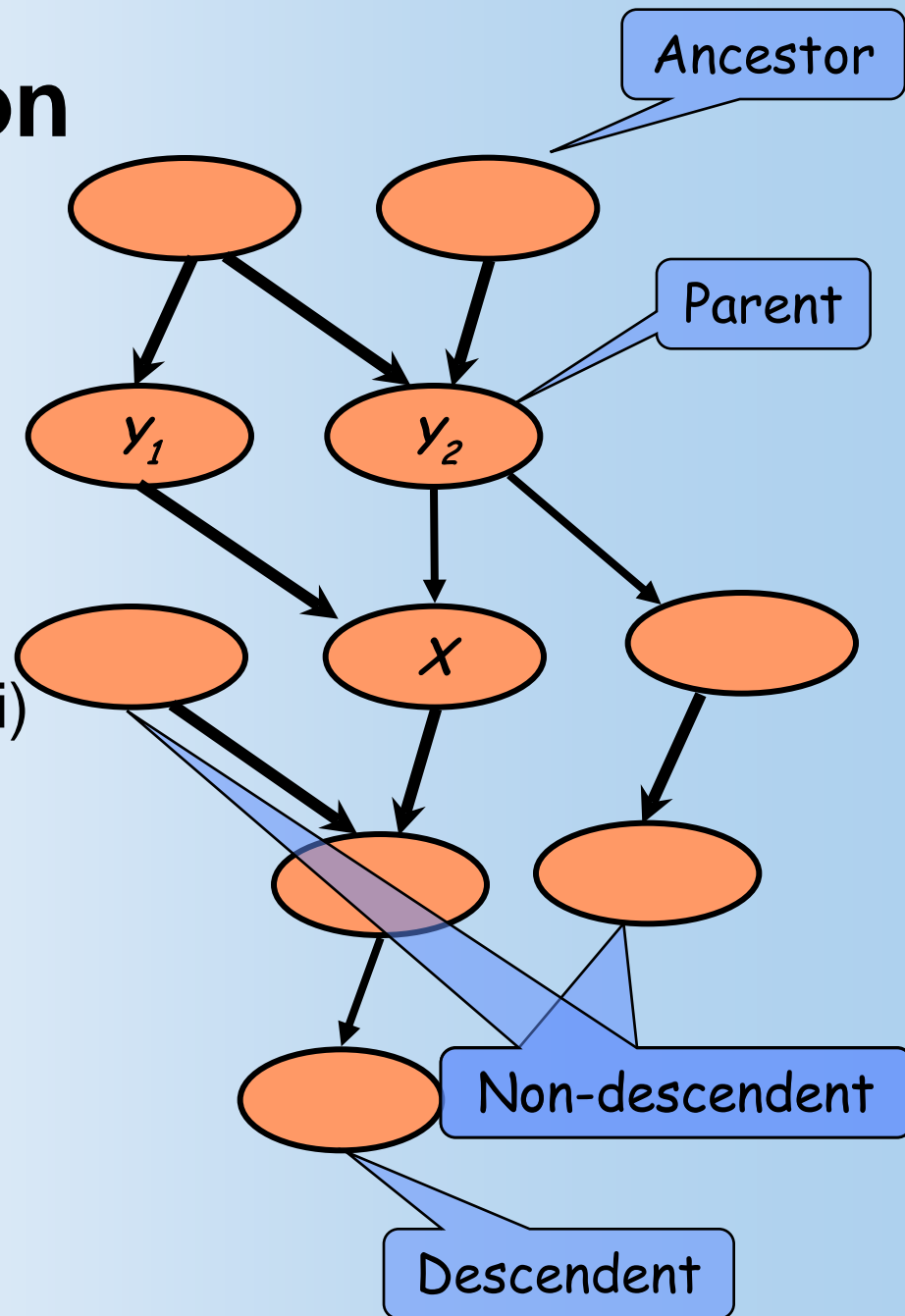
Markov Assumption

Generalizing:

- ◆ A child is **conditionally independent** from its non-descendants, given the value of its parents.

$\text{Ind}(X_i ; \text{NonDescendant}_{X_i} \mid \text{Pa}_{X_i})$

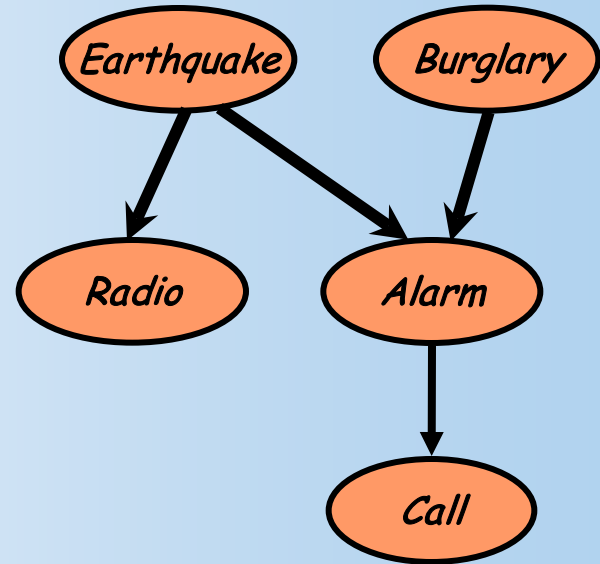
- ◆ It is a natural assumption for many **causal** processes



Markov Assumption (cont.)

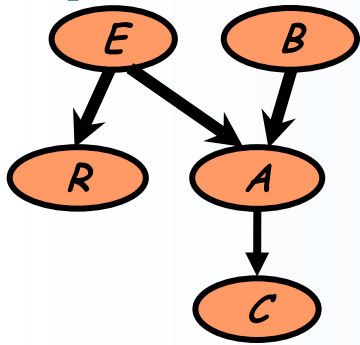
◆ In this example:

- R is independent of A, B, C , given E
- A is independent of R , given B and E
- C is independent of B, E, R , given A



- ◆ Are other independencies implied by these ones?
- ◆ A graph-theoretical criteria identifies all such independencies

Bayesian Network Semantics



Qualitative part
conditional
independence
statements
in BN structure

Quantitative part
local
probability
Models
+ (e.g., multinomial,
linear Gaussian)

= Unique joint
distribution
over domain

◆ Compact & efficient representation:

- nodes have $\leq k$ parents $\Rightarrow O(2^k n)$ vs. $O(2^n)$ params
- parameters pertain to local interactions

$$P(C, A, R, E, B) = P(B) * P(E|B) * P(R|E, B) * P(A|R, B, E) * P(C|A, R, B, E)$$

versus

$$P(C, A, R, E, B) = P(B) * P(E) * P(R|E) * P(A|B, E) * P(C|A)$$

→ In general:
$$P(x_1, \dots, x_n) = \prod_{i=1, \dots, n} P(x_i | Pa_{x_i})$$

Outline

- ◆ Introduction
- ◆ Bayesian Networks
 - Representation & Semantics
 - » Inference in Bayesian networks
 - Learning Bayesian networks

Inference in Bayesian networks

- ◆ A Bayesian network represents a probability distribution.
- ◆ Can we answer queries about this distribution?

Examples:

- ◆ $P(Y|Z=z)$
- ◆ Most probable estimation $MPE(W | Z = z) = \arg \max_w P(w, z)$
- ◆ Maximum a posteriori $MAP(Y | Z = z) = \arg \max_y P(y | z)$

Inference in Bayesian networks

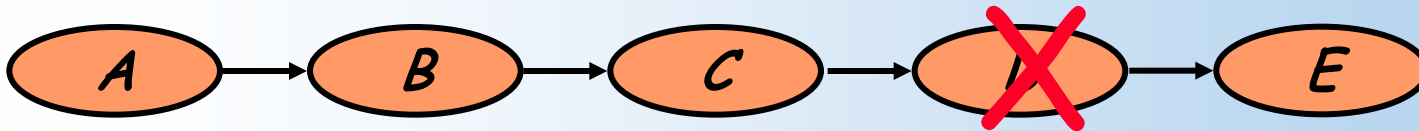
- ◆ Goal: compute $P(E=e, A=a)$ in the following Bayesian network:



- ◆ Using definition of probability, we have

$$\begin{aligned} P(a, e) &= \sum_b \sum_c \sum_d P(a, b, c, d, e) \\ &= \sum_b \sum_c \sum_d P(a)P(b | a)P(c | b)P(d | c)P(e | d) \end{aligned}$$

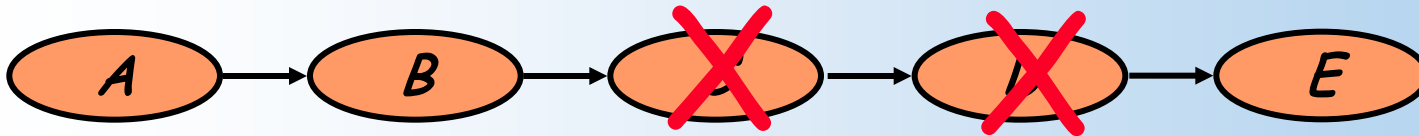
Inference in Bayesian networks



◆ Eliminating d , we get

$$\begin{aligned} P(a, e) &= \sum_b \sum_c \sum_d P(a)P(b|a)P(c|b)P(d|c)P(e|d) \\ &= \sum_b \sum_c P(a)P(b|a)P(c|b) \underbrace{\sum_d P(d|c)P(e|d)}_{P(e|c)} \\ &= \sum_b \sum_c P(a)P(b|a)P(c|b)P(e|c) \end{aligned}$$

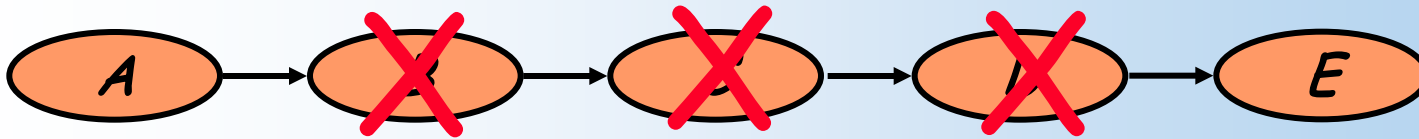
Inference in Bayesian networks



◆ Eliminating c , we get

$$\begin{aligned} P(a, e) &= \sum_b \sum_c P(a)P(b|a)P(c|b)P(e|c) \\ &= \sum_b P(a)P(b|a) \sum_c P(c|b)P(e|c) \\ &= \sum_b P(a)P(b|a)p(e|b) \end{aligned}$$

Inference in Bayesian networks



◆ Finally, we eliminate b

$$\begin{aligned} P(a, e) &= \sum_b P(a)P(b | a)p(e | b) \\ &= P(a) \underbrace{\sum_b P(b | a)p(e | b)}_{P(e | a)} \\ &= P(a)P(e | a) \end{aligned}$$

Variable Elimination Algorithm

General idea:

- ◆ Write query in the form

$$P(x_1) = \sum_{x_k} \cdots \sum_{x_3} \sum_{x_2} \prod_i P(x_i | p \ \varphi)$$

- ◆ Iteratively
 - Move all irrelevant terms outside of innermost sum
 - Perform innermost sum, getting a new term
 - Insert the new term into the product
- ◆ In case of evidence $P(x_1 | \text{evidence } x_j)$, use: $P(x_i | x_j) = P(x_i, x_j) / P(x_j)$

Complexity of inference

Lower bound:

General inference in BNs is **NP-hard**.

Upper bound:

Naïve exact inference

- ◆ **exponential** in the number of variables in the network

Variable elimination complexity

- ◆ **exponential** in the size of largest factor
- ◆ **polynomial** in the number of variables in the network
- ◆ Variable elimination computation depend on order of elimination (many heuristics, e.g., clique tree algorithm).

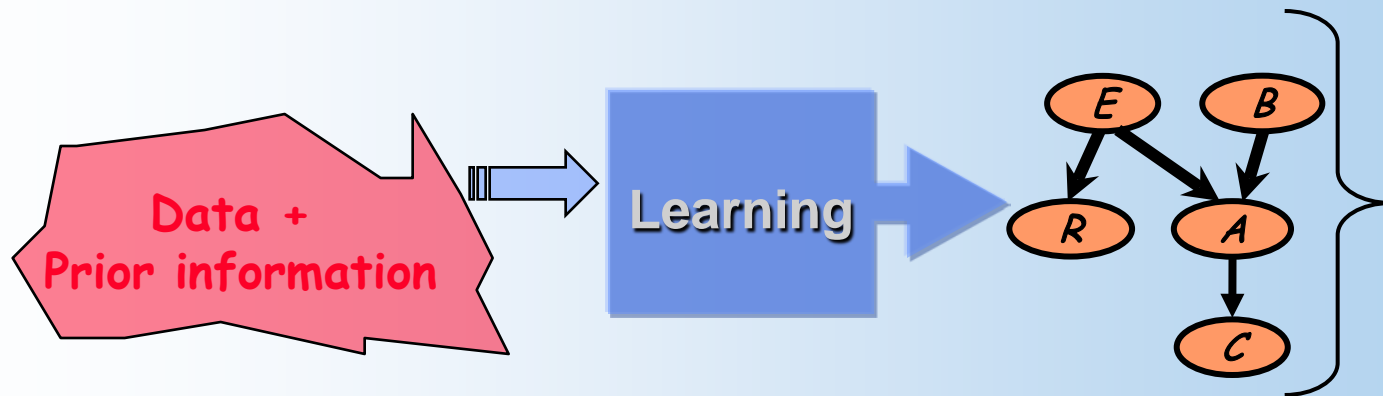
Outline

- ◆ Introduction
- ◆ Bayesian Networks
 - Representation & Semantics
 - Inference in Bayesian networks
 - » Learning Bayesian networks
 - ◆ Parameter Learning
 - ◆ Structure Learning

Learning

◆ Process

- **Input:** dataset and prior information
- **Output:** Bayesian network



The Learning Problem

	Known Structure	Unknown Structure
Complete Data	Statistical parametric estimation (closed-form eq.)	Discrete optimization over structures (discrete search)
Incomplete Data	Parametric optimization (EM, gradient descent...)	Combined (Structural EM, mixture models...)

- ◆ We will focus on complete data for the rest of the talk
 - The situation with incomplete data is more involved

Outline

- ◆ Introduction
- ◆ Bayesian Networks
 - Representation & Semantics
 - Inference in Bayesian networks
 - Learning Bayesian networks
 - » Parameter Learning
 - ◆ Structure Learning

Learning Parameters

- ◆ Key concept: the **likelihood function**

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

- measures how the probability of the data changes when we change parameters
-
- ◆ Estimation:
 - MLE: choose parameters that maximize likelihood
 - Bayesian: treat parameters as an unknown quantity, and marginalize over it

MLE principle for Binomial Data

◆ Data: H, H, T, H, H . Θ is the unknown probability $P(H)$.

◆ Likelihood function: $L(\Theta : D) = \prod_{k=0,1} \theta_k^{N_k}$

$$L(\theta : D) = \theta \cdot \theta \cdot (1 - \theta) \cdot \theta \cdot \theta$$

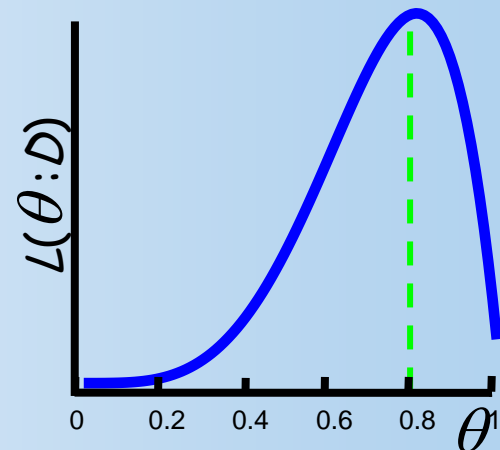
◆ Estimation task: Given a sequence of samples $x[1], x[2] \dots x[M]$, we want to estimate the probability $P(H) = \theta$ and $P(T) = 1 - \theta$.

◆ MLE principle: choose parameter that maximize the likelihood function.

◆ Applying the MLE principle we get

$$\hat{\theta} = \frac{N_H}{N_H + N_T}$$

◆ MLE for $P(X = H)$ is $4/5 = 0.8$



MLE principle for Multinomial Data

- ◆ Suppose X can have the values $1, 2, \dots, k$.
- ◆ We want to learn the parameters $\theta_1, \dots, \theta_k$.
- ◆ N_1, \dots, N_k - The number of times each outcome is observed.

- ◆ Likelihood function:

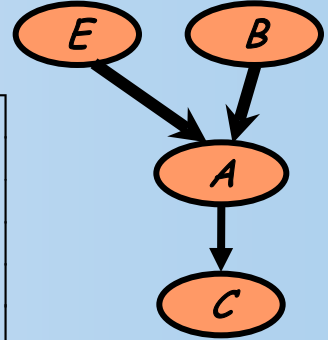
$$\mathcal{L}(\Theta : \mathcal{D}) = \prod_{k=1}^K \theta_k^{N_k}$$

Count of k^{th} outcome in \mathcal{D}

Probability of k^{th} outcome

- ◆ The MLE is: $\hat{\theta}_i = \frac{N_i}{\sum_{l=1, \dots, k} N_l}$

MLE principle for Bayesian networks



- ◆ Training data has the form:

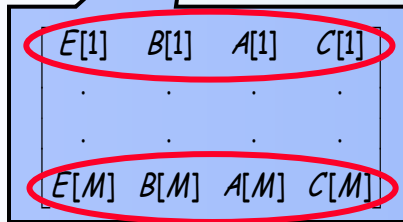
$$D = \begin{bmatrix} E[1] & B[1] & A[1] & C[1] \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ E[M] & B[M] & A[M] & C[M] \end{bmatrix}$$

- ◆ Assume i.i.d. samples

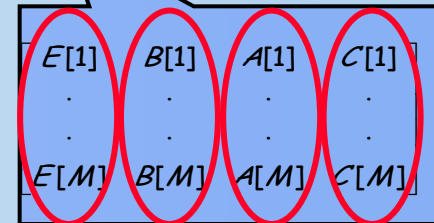
$$L(\Theta : D) = \prod_m P(E[m], B[m], A[m], C[m] : \Theta)$$

$$= \prod_m \left(\begin{array}{l} P(E[m] : \Theta) \\ P(B[m] : \Theta) \\ P(A[m] | B[m], E[m] : \Theta) \\ P(C[m] | A[m] : \Theta) \end{array} \right)$$

By definition of network



$$= \prod_m P(E[m] : \Theta) \prod_m P(B[m] : \Theta) \prod_m P(A[m] | B[m], E[m] : \Theta) \prod_m P(C[m] | A[m] : \Theta)$$



MLE principle for Bayesian networks

- ◆ Generalizing for any Bayesian network:

$$L(\Theta : D) = \prod_i \prod_m P(x_i[m] | Pa_i[m] : \Theta_i) = \prod_i L_i(\Theta_i : D)$$

$$\begin{aligned} L_i(\theta_i : D) &= \prod_m P(x_i[m] | Pa_i[m] : \theta_i) \\ &= \prod_{pa_i} \prod_{x_i} P(x_i | pa_i : \theta_i)^{N(x_i, pa_i)} = \prod_{pa_i} \prod_{x_i} \theta_{x_i | pa_i}^{N(x_i, pa_i)} \end{aligned}$$

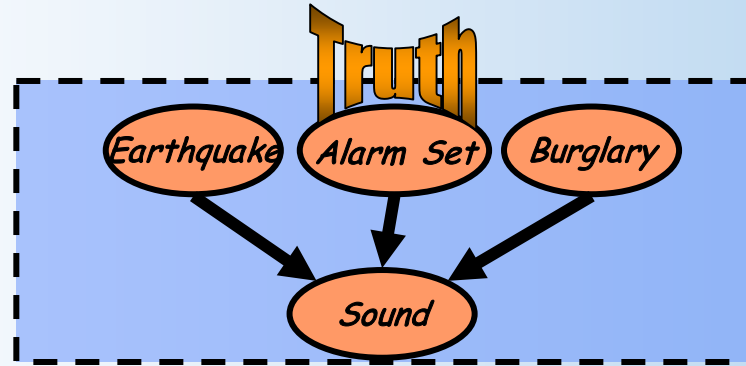
- The likelihood decomposes according to the network structure.
- **Decomposition \Rightarrow Independent estimation problems**
(If the parameters for each family are not related)
- For each value pa_i of the parent of X_i we get independent multinomial problem.

- The **MLE** is $\hat{\theta}_{x_i | pa_i} = \frac{N(x_i, pa_i)}{N(pa_i)}$

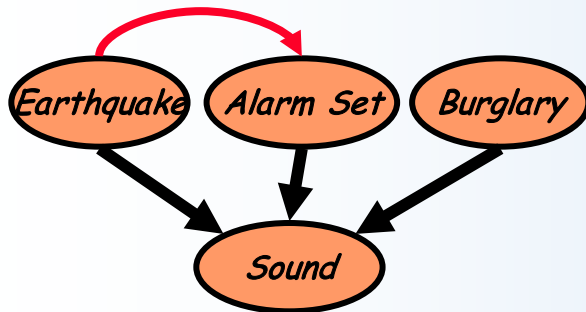
Outline

- ◆ Introduction
- ◆ Bayesian Networks
 - Representation & Semantics
 - Inference in Bayesian networks
 - Learning Bayesian networks
 - ◆ Parameter Learning
 - » Structure Learning

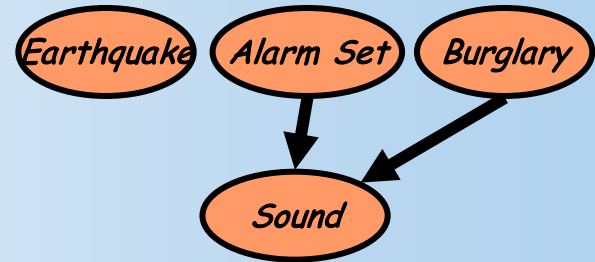
Learning Structure: Motivation



Adding an arc



Missing an arc



Optimization Problem

Input:

- Training data
- Scoring function (including priors)
- Set of possible structures

Output:

- A network (or networks) that maximize the score

Key Property:

- **Decomposability:** the score of a network is a sum of terms.

Scores

For example. The BDE score:

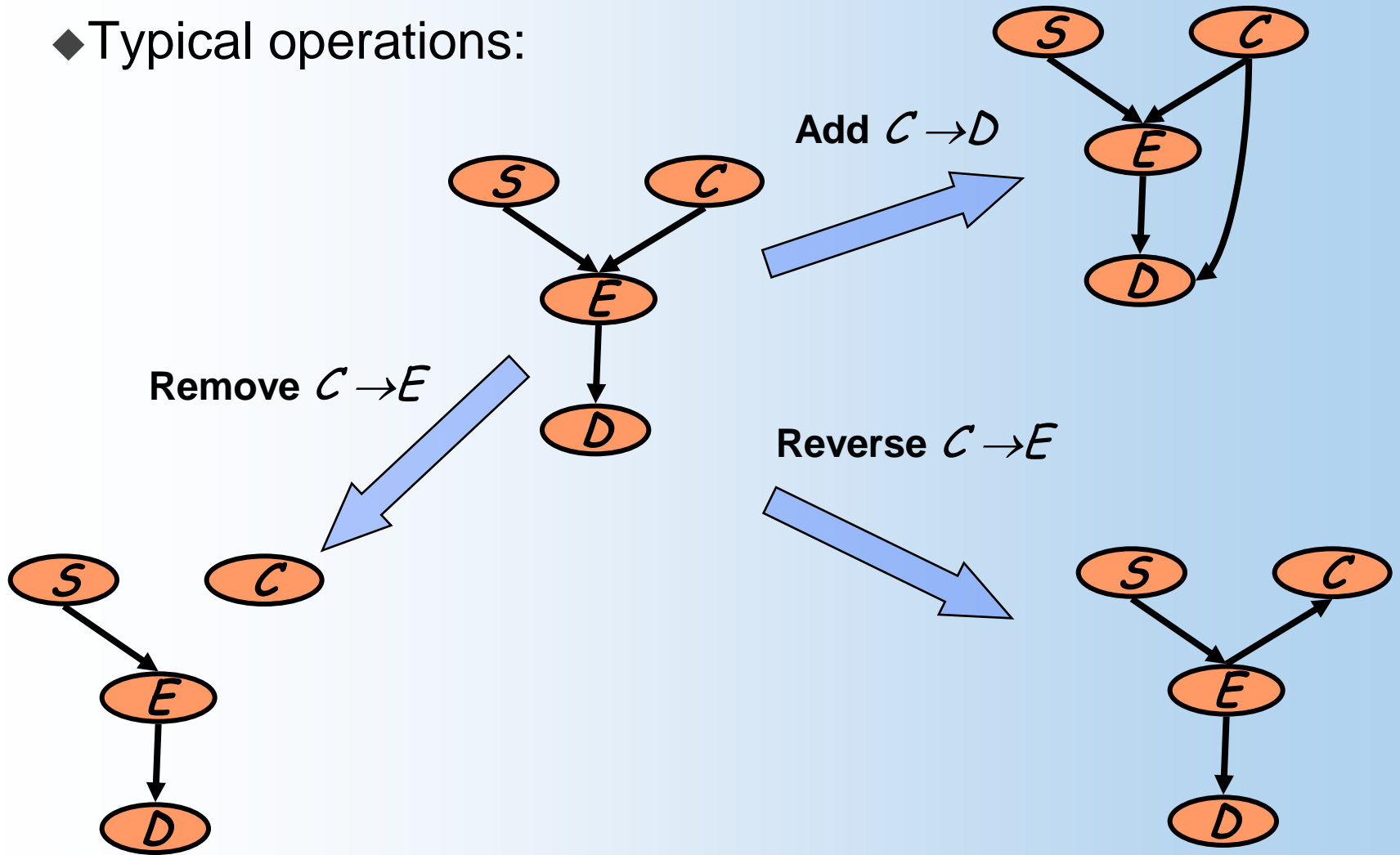
$$\begin{aligned} \text{Score}(G : D) &= P(G | D) \propto P(D | G)P(G) \\ &= \int P(D | G, \theta)P(\theta | G)d\theta P(G) \end{aligned}$$

When the data is complete, the score is **decomposable**:

$$\text{Score}(G : D) = \sum_i \text{Score}(X_i | Pa_i^G : D)$$

Heuristic Search (cont.)

◆ Typical operations:



Heuristic Search

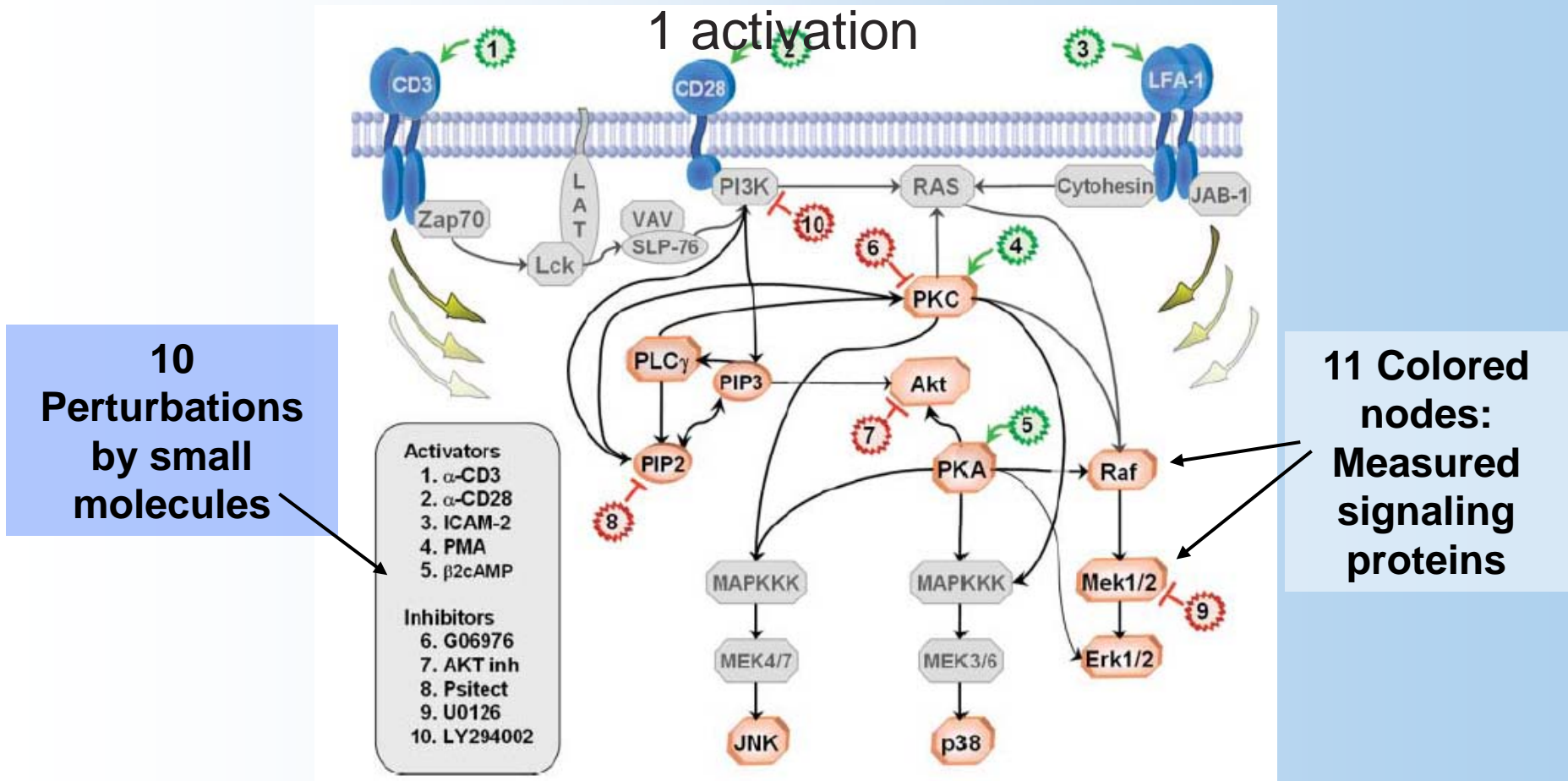
- ◆ We address the problem by using heuristic search
- ◆ Traverse the space of possible networks, looking for high-scoring structures
- ◆ Search techniques:
 - Greedy hill-climbing
 - Simulated Annealing
 - ...

Outline

- ◆ Introduction
- ◆ Bayesian Networks
 - Representation & Semantics
 - Inference in Bayesian networks
 - Learning Bayesian networks
 - Conclusion
- ◆ Applications

Example 1

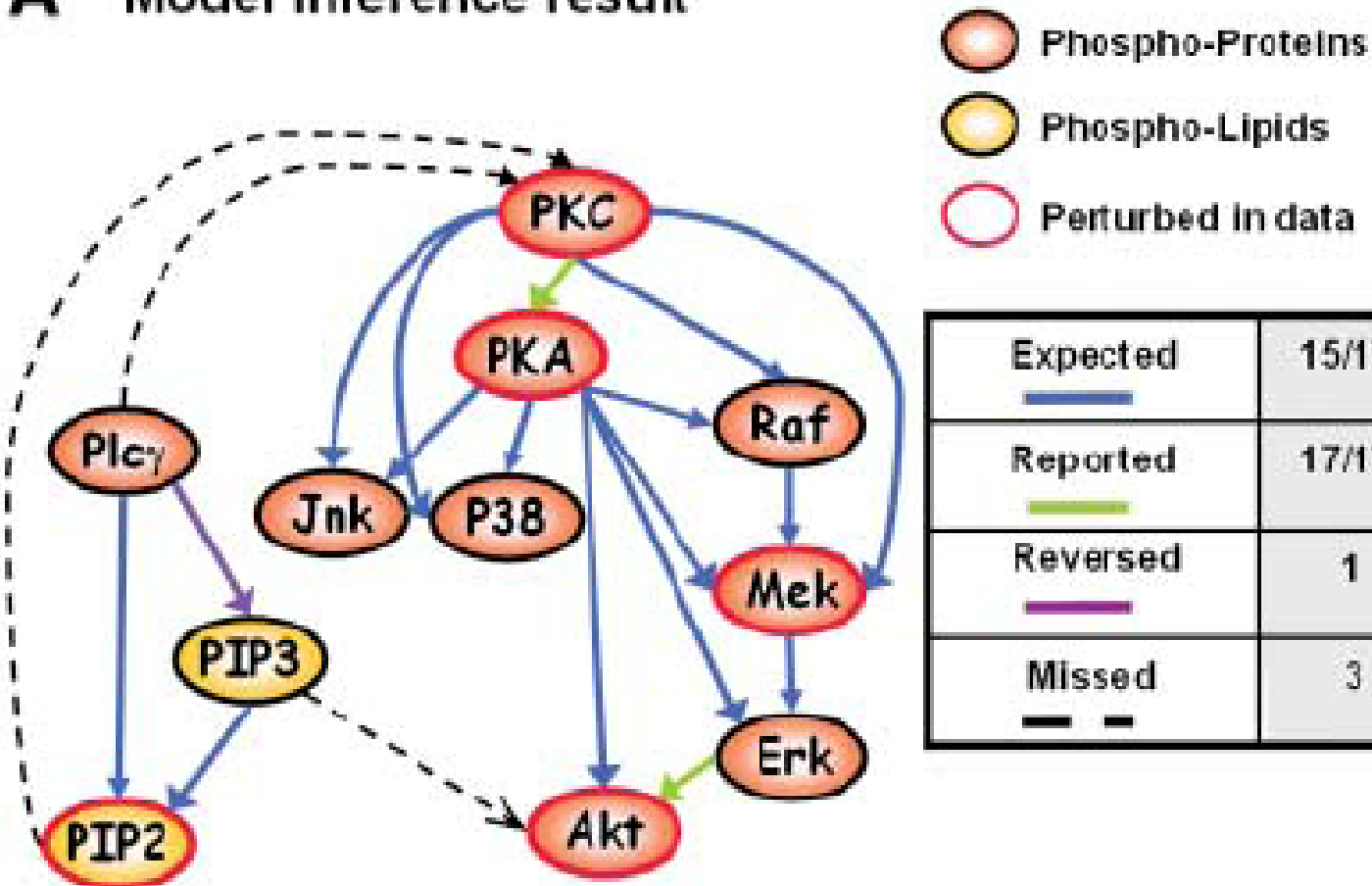
The currently accepted consensus network of human primary CD4 T cells, downstream of CD3, CD28, and LFA-1



Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. Karen Sachs, *et al.* 2005.

Bayesian network results

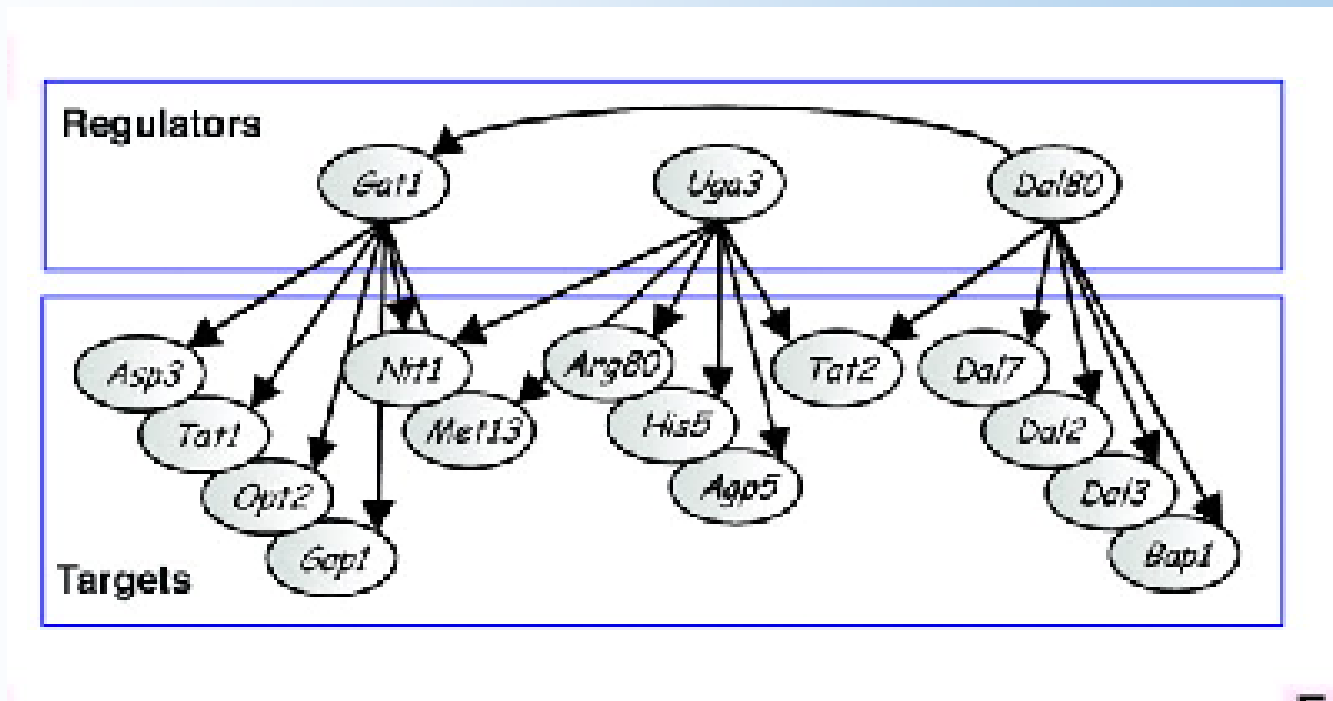
A Model inference result



- PKC \rightarrow PKA was validated experimentally.
- Akt was not affected by Erk in additional experiments

Example II: How to increase robustness

Two-level network



Challenges

- ◆ Correct handling of hidden variables (active proteins)
- ◆ Incorporating prior knowledge
 - Clearly, need to incorporate large mass of biological knowledge, and insight from sequence/structure databases
- ◆ Incorporating additional data sources
- ◆ Experimental design
 - How to learn causality from knockout experiments? How to plan such experiments?
- ◆ External Variables:
 - We want to relate regulation to external events: stimuli, temperature, nutrient levels, etc.
- ◆ Modeling Feedback loops

General References:

- ◆ D. Koller and N. Friedman, *probabilistic graphical models*
- ◆ Pearl, *Probabilistic Reasoning in Intelligent Systems*
- ◆ Jensen, *An Introduction to Bayesian Networks*
- ◆ Heckerman, *A tutorial on learning with Bayesian networks*