“Canonical” Probability Distributions

Marek J. Drużdżel

University of Pittsburgh

School of Information Sciences and Intelligent Systems Program

marek@sis.pitt.edu
http://www.pitt.edu/~druzdzel

Politechnika Białostocka

Wydział Informatyki

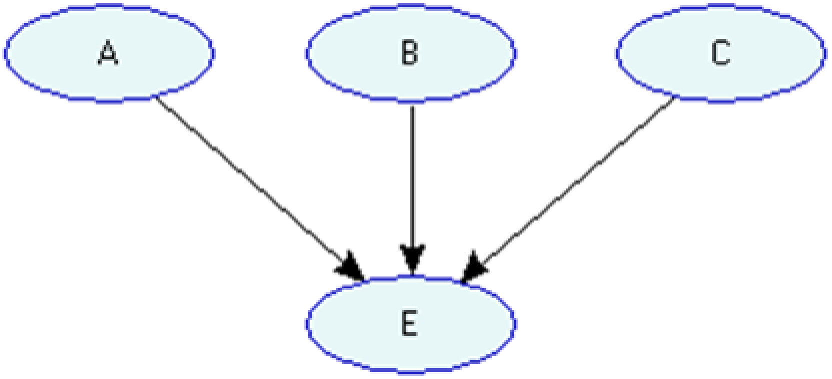
m.druzdzel@pb.edu.pl
http://www.wi.pb.edu.pl/~druzdzel/

Outline

- Motivation
- Noisy-OR
- Leaky Noisy-OR
- Noisy-MAX
- Examples
- Concluding remarks

Why do we need them?

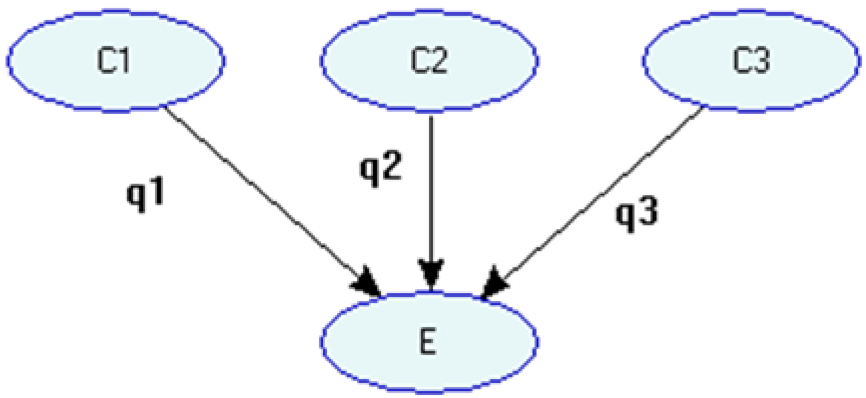
- A major problems with Bayesian Networks is the exponential growth of conditional probability tables (CPTs) in the number of parents



- This is a serious practical limitation

Solution: Noisy-OR

- Various solutions were proposed, but one of them seems to be most popular and useful: Noisy-OR
- We assume nodes are binary {present, absent}
- One numerical parameter per parent q_i



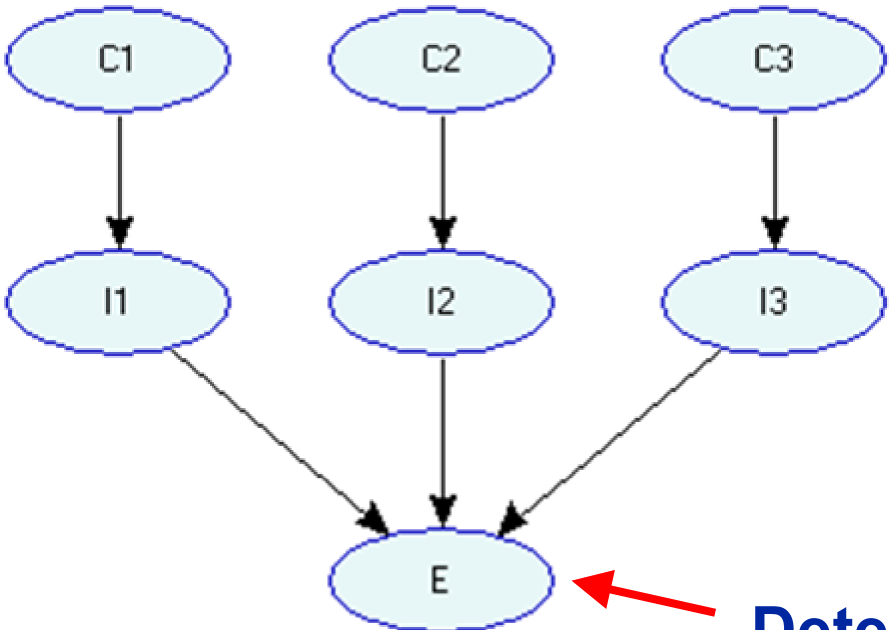
Noisy-OR: The meaning of q_i?

- q_i the meaning of probability, so it takes values between 0 and 1
- q_i is the probability that E=present given C_i=present and all other parents C=absent

$$q_i=P(E=\mathit{present} \mid C_1=\mathit{absent}, \ldots, C_i=\mathit{present}, \ldots, C_n=\mathit{absent})$$

Why is it called Noisy-OR?

- If all parameters $q_i = 1$, noisy-OR becomes logical OR.
- An alternative representation of Noisy-OR



Node0	present	absent
present	q	0
absent	1-q	1

Deterministic OR

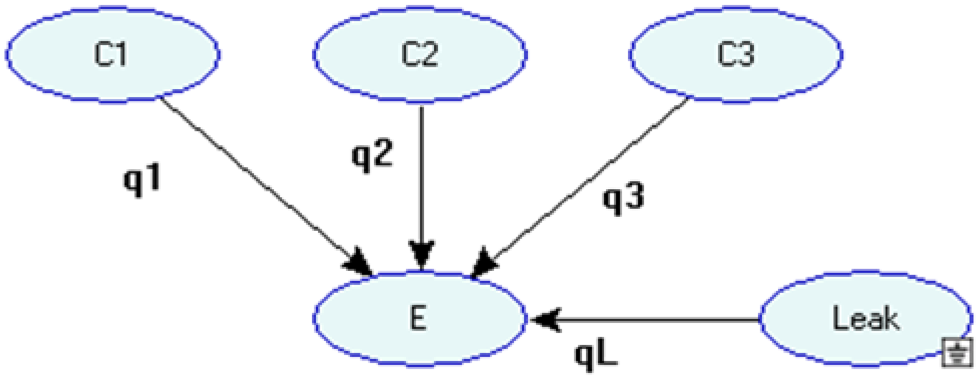
Noisy-OR vs. CPT

- Noisy-OR always defines a unique CPT (i.e., you can always calculate the CPT that is defined by a noisy-OR gate)

$$P(E = absent \mid C1,\ldots,Cn) = \prod_{C_i = present} (1 - q_i)$$

Leaky Noisy-OR

- Noisy-OR assumes that the effect will be absent with probability 1 if all the causes are absent. This is not very realistic.
- Leak – special dummy node, that represents influence of all unmodeled causes on the effect node.
- Leak is always present



Leaky Noisy-OR: Parameters

- An extension to the Noisy-OR
- Two parameterizations of leaky Noisy-OR: due to Henrion and Diez
- They are mathematically equivalent, however imply different questions in knowledge elicitation

Leaky Noisy-OR: Diez

- **Leak probability q_L:**

$$q_L = P(E = present \mid C1 = absent,..,CN = absent)$$

- **Link probability q_i:**

$$q_i = P(E = present \mid C1 = absent,...,Ci = present, CN = absent, L = absent)$$

- **How to calculate the CPT:**

$$P(E = absent \mid C1,...,Cn) = (1 - q_L) \prod_{C_i = present} (1 - q_i)$$

Leaky Noisy-OR: Henrion

- **Leak probability p_L: (same as Diez)**

$$p_L = P(E = present \mid C1 = absent, .., CN = absent)$$

- **Link probability p_i: (no leak term)**

$$p_i = P(E = present \mid C1 = absent, ..., Ci = present, CN = absent)$$

- **How to calculate CPT:**

$$P(E = absent \mid C1, ..., Cn) = (1 - p_L) \prod_{C_i = present} \frac{1 - p_i}{1 - p_L}$$

Henrion vs. Diez

- **They imply different questions to ask of experts:**
- **Henrion:**

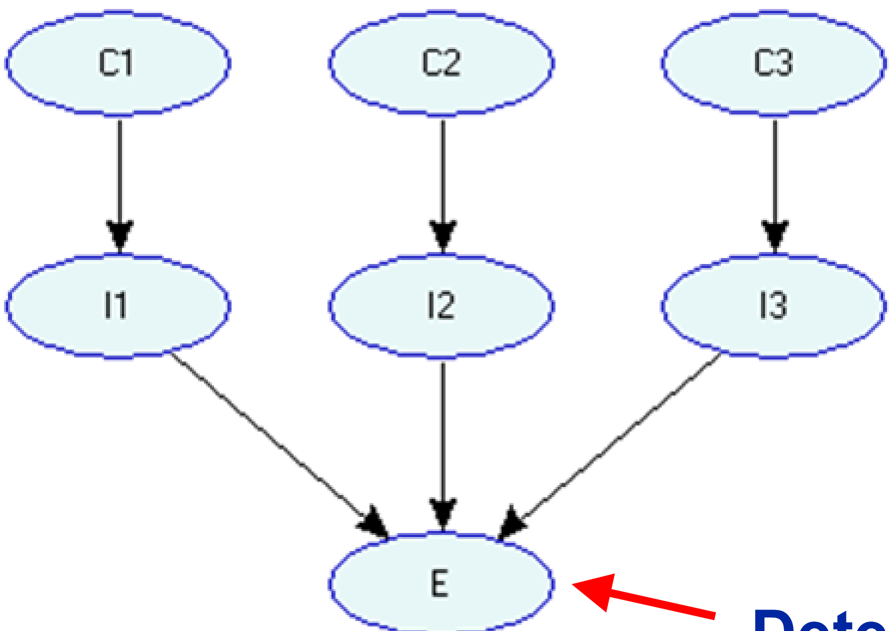
"What is the probability that E is present given that C_i is present and all other modeled causes are absent?"

- **Diez:**

"What is the probability that E is present given that C_i is present and all other modeled and unmodeled causes are absent?"

Noisy-MAX

- **Noisy-MAX is a version of Noisy-OR for multi-valued nodes.**

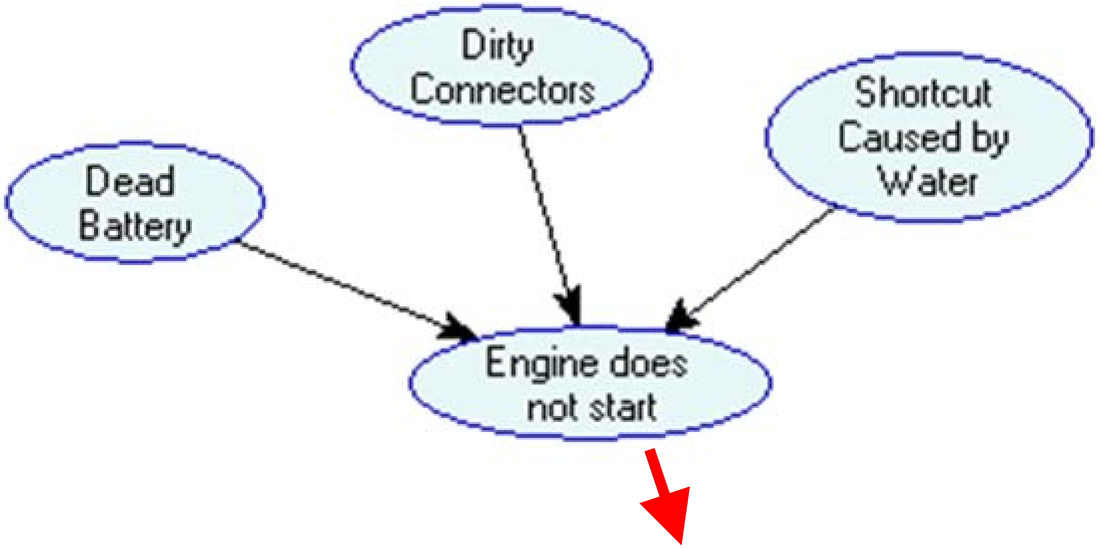


Node2	high	med	low
high	0.7	0.5	0
medium	0.2	0.3	0
low	0.1	0.2	1

Deterministic MAX

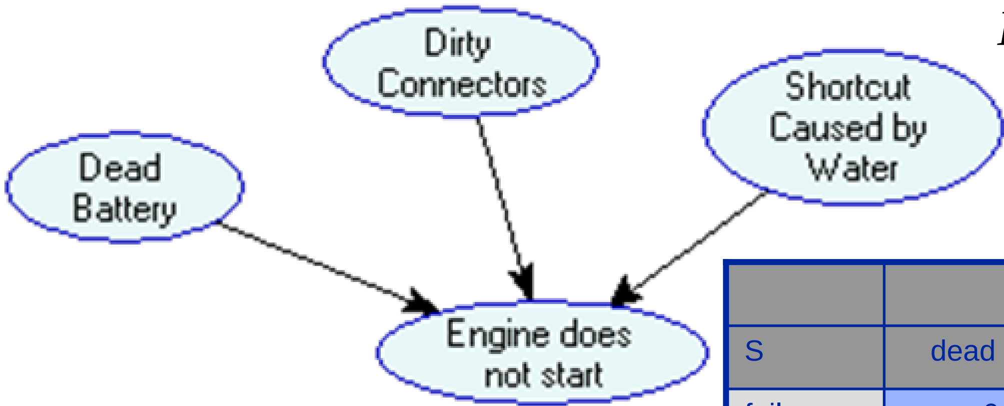
Examples

Deterministic OR



DB	ok				dead			
DC	clean		dirty		clean		dirty	
S	ok	short	ok	short	ok	short	ok	short
fail	0	1	1	1	1	1	1	1
start	1	0	0	0	0	0	0	0

Noisy-OR



$$P(E = absent \mid C1,\ldots,Cn) = \prod_{C_i = present} (1 - q_i)$$

	DB		DC		S	
S	dead	ok	dirty	clean	short	ok
fail	0.9	0	0.8	0	0.5	0
stat	0.1	1	0.2	1	0.5	1

DB	ok				dead			
DC	clean		dirty		clean		dirty	
S	ok	short	ok	short	ok	short	ok	short
fail	0	0.5	0.8	0.9	0.9	0.95	0.98	0.99
stat	1	0.5	0.2	0.1	0.1	0.05	0.02	0.01

Leaky Noisy-OR

"Leak" or "background" probability to model all unmodeled causes is often used

	DB		DC		S		leak
S	dead	ok	dirty	clean	short	ok	
fail	0.9	0	0.8	0	0.5	0	0.1
stat	0.1	1	0.2	1	0.5	1	0.9

DB	ok				dead			
DC	clean		dirty		clean		dirty	
S	ok	short	ok	short	ok	short	ok	short
fail	0.1	0.5	0.8	0.888	0.9	0.944	0.977	0.987
stat	0.9	0.5	0.2	0.112	0.1	0.056	0.023	0.013

$$P(E = absent \mid C1,\ldots,Cn) = (1 - q_L) \prod_{C_i = present} \frac{1 - q_i}{1 - q_L}$$

Concluding remarks

- In practical models, canonical gates are the only way to go
- There are significant computational advantages that stem from canonical gates