# BAYESIAN NETWORKS

Sebastian Ordyniak

ALGORITHMS AND COMPLEXITY GROUP

# Bayesian Networks

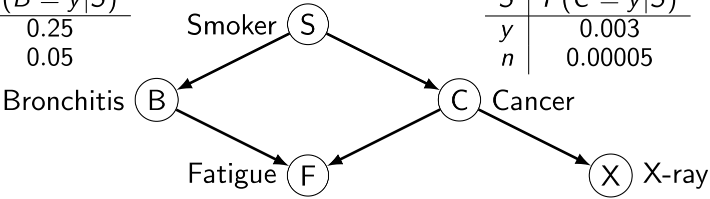
- introduced by Judea Perl in 1985 (2011 Turing Award Winner)
- compact representation of probability distribution via a DAG plus tables associated with the nodes of the network
- other and related probabilistic networks include: Markov Random Fields, Factor Graphs . . .

# Example

| $P(S=y)$ |
|---|
| 0.2 |

| $S$ | $P(B=y\|S)$ |
|---|---|
| $y$ | 0.25 |
| $n$ | 0.05 |

| $S$ | $P(C=y\|S)$ |
|---|---|
| $y$ | 0.003 |
| $n$ | 0.00005 |

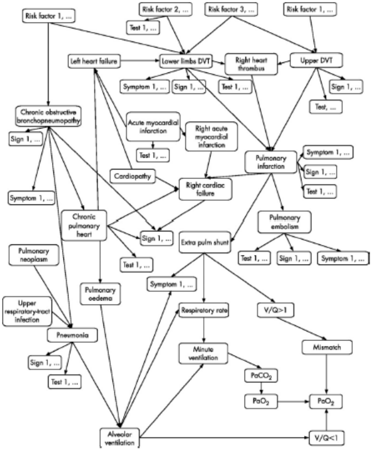



| $B$ | $C$ | $P(F=y\|B,C)$ |
|---|---|---|
| $y$ | $y$ | 0.75 |
| $y$ | $n$ | 0.1 |
| $n$ | $y$ | 0.5 |
| $n$ | $n$ | 0.05 |

| $C$ | $P(X=y\|C)$ |
|---|---|
| $y$ | 0.6 |
| $n$ | 0.02 |

# Applications

- diagnosis
- computational biology
- document classification
- information retrieval
- image processing
- decision support
- etc.



# Computational Problems

## BN Reasoning

Given a BN, compute the probability of a variable taking a specific value (possibly conditioned on values of other variables).

## BN Learning

Given a set of sample data, find a BN that "best fits" the data.

- BN Structure Learning: Given sample data find the best DAG
- BN Parameter Learning: Given sample data and a DAG, find the best probability tables

# BN Reasoning

### Problem:

Given a BN, compute the probability of a variable taking a specific value (possibly conditioned on values of other variables).

### Complexity

- #P-complete, usually solved via weighted model counting
- fixed-parameter tractable w.r.t. the "treewidth of the BN".

# BN Learning

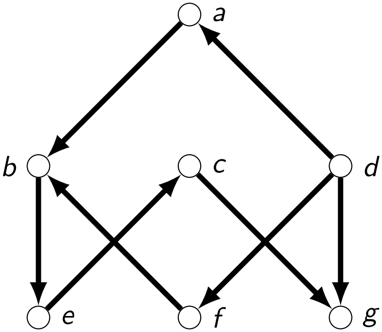
## Problem:

Given sample data, find a BN that “best fits” the data.

| $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ | $\#D$ |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 0 | 1 | 1 | 3 |
| 1 | 1 | 0 | 1 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 5 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | 6 |
| 1 | 0 | 1 | 0 | 0 | 0 | 0 | 3 |
| 1 | 0 | 1 | 0 | 1 | 1 | 0 | 4 |

$D = \{D_1, \ldots, D_m\}$

$\mapsto$




An optimal BN for $D$.

# BN Learning: Combinatorial Model

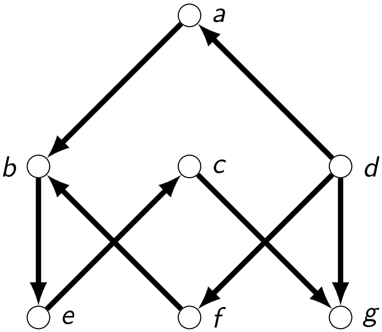
## Problem:

Given a local score function $f$ that assigns scores to every parent set of a node, find a BN (DAG) whose score is maximum with respect to $f$.

| node | set of parents | score |
|---|---|---|
| $a$ | $\{d\}$ | 1 |
| $a$ | $\{b, c, d\}$ | 0.5 |
| $b$ | $\{a, f\}$ | 1 |
| $c$ | $\{e\}$ | 1 |
| $d$ | $\emptyset$ | 1 |
| $e$ | $\{b\}$ | 1 |
| $f$ | $\{d\}$ | 1 |
| $g$ | $\{c, d\}$ | 1 |

A local score function $f$

$\mapsto$




An optimal BN for $f$

# BN Learning: Combinatorial Model

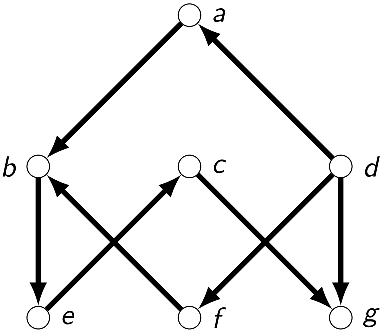
## Problem:

Given a local score function $f$ that assigns scores to every parent set of a node, find a BN (DAG) whose score is maximum with respect to $f$.

| node | set of parents | score |
|---|---|---|
| $a$ | $\{d\}$ | 1 |
| $a$ | $\{b, c, d\}$ | 0.5 |
| $b$ | $\{a, f\}$ | 1 |
| $c$ | $\{e\}$ | 1 |
| $d$ | $\emptyset$ | 1 |
| $e$ | $\{b\}$ | 1 |
| $f$ | $\{d\}$ | 1 |
| $g$ | $\{c, d\}$ | 1 |

A local score function $f$

$\mapsto$




An optimal BN for $f$

NP-complete!!

# BN Learning

We focused on learning BN that are "good" for BN Reasoning.

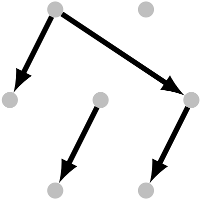
- BN that are almost "acyclic" ($k$-**Branchings**),
- BN contained in a "super-structure" of bounded treewidth

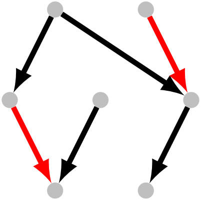
# BN Learning: $k$-Branchings

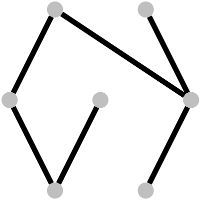
Branching: BN where every node has at most one parent.

$k$-Branching: BN that is a polytree and becomes a branching after deleting $k$ arcs.

Polytree: BN whose underlying undirected graph is acyclic.







A maximum branching can be found in polynomial time (Chu and Liu, 1965).

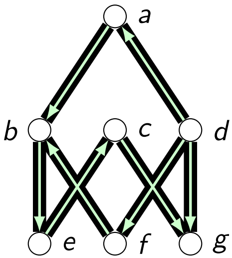
Our Result: For every $k$, a maximum $k$-branching can be found in polynomial time.

Finding a maximum polytree is NP-hard (Dasgupta, 1999).

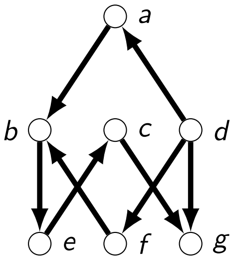
# BN Learning: Superstructure Approach

| node | set of parents | score |
|---|---|---|
| $a$ | $\{d\}$ | 1 |
| $a$ | $\{b, c, d\}$ | 0.5 |
| $b$ | $\{a, f\}$ | 1 |
| $c$ | $\{e\}$ | 1 |
| $d$ | $\emptyset$ | 1 |
| $e$ | $\{b\}$ | 1 |
| $f$ | $\{d\}$ | 1 |
| $g$ | $\{c, d\}$ | 1 |

$\mapsto$




$\mapsto$




**Local Score function**

**Superstructure $S$**

**An optimal BN contained in $S$**

# BN Learning: Superstructure Approach

## Main Results

BN Learning parameterized by the treewidth of the superstructure is:

- in XP and W[1]-hard
- in FPT if addionally the superstructure has bounded degree

Implementation of the FPT algorithm for treewidth and bounded degree is available.

# Thank You!