

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probabilities: $P(x)$
 Probability: $P(x)$
 Quantifying uncertainty
 Not a trade-off in classical agent systems
 Aids: Complexity representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 In: $P(x)$
 Out: $P(x)$

Graphical models 101

Classes of graphical models

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data

Graph theory

Maximal cliques and minimal separators

What are decomposable models
 Decomposable models are a Markov Random Fields for which the graph is chordal or triangulated

Properties of decomposable models

1. Closed form for $P(x)$
2. No. of log-likelihood
3. Junction tree algorithm
4. No. of cliques $O(n^2)$
5. Linear-time separable property [3, 4]
6. Interaction between ICI and MRF [2]

Useful algorithms

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:

1. scalable scoring
2. efficient search
3. scalable belief propagation

= all the results we will show here

Bottom line
 A score decomposable models is equivalent to:
 - AIC for MRFs
 - A set of separable Bayesian networks

Most scores are scalable
 Entropy [1] ✓
 Bayesian Ladder [2] ✓ Because it is unrolled when entropy is used
 Gradient [3] ✓
 Max. FMS [4] ✓

Break

Efficient search

Scoring in greedy search

Clique graph (CG)

Clique graph and greedy search

Search and statistical paradigm

The nitty-gritty

Counting efficiently
 Scoring for example with KL minimized when...
 What does it mean to compute $P(x)$?
 Efficient counting built upon efficient counting

Counting efficiently (2)
 Many different counting techniques require making use of the CG

Memorization
 From the high multiplicity of the CG to the low multiplicity of the CG

Addition of the same edge to different reference models
 What we have seen so far
 Curvature
 How often does that happen?
 How can we use this information?

How fast can we get?

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell

1. Graphical models are everywhere
2. We can learn graphical models from data with 1,000+ variables
3. It is crucial to be able to use the library that we are providing on your tablet device
4. There is still so much work to be done

Open problems

1. Efficient constrained search
2. Better scores (eg on Directed scoring on MRF)
3. Efficient scoring of marginal "bits"
4. Efficient data structures for counting
5. Learning out of core
6. Latent variables

Open problems (2)

1. How to handle numerical variables
2. How to handle missing values?
3. Learning accurate parameters in large tables

Scalable learning of graphical models
 From the lecture and your study

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probabilities: $\sum_{\mathbf{x}} p(\mathbf{x}) = 1$
 Probability: $p(\mathbf{x}) = \prod_{i,j} \theta_{ij}^{x_{ij}}$
 Quantifying uncertainty: θ_{ij}
 Not a trade-off in classical agent systems
 Aka: Compactly representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 In: θ_{ij}
 Out: θ_{ij}

Graphical models 101

Classes of graphical models

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data

Graph theory

Maximal cliques and minimal separators

What are decomposable models
 Decomposable models are a Markov Random Fields for which the graph is chordal or triangulated

Properties of decomposable models

1. Closed form for $p(\mathbf{x}) = \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c)$
2. No. of log-likelihood: Each component that can be reduced to a graphical model can be reduced to a sum of log-likelihoods
3. Junction-tree algorithm: Exact inference using junction tree
4. No. of cliques: $O(n^2)$
5. Linear-time: decomposable property [1, 4]
6. Interaction between IJ and MRF: [2]

Useful algorithms

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:
 1. scalable scoring
 2. efficient search
 3. scalable belief propagation
 = all the results we will show here

Bottom line
 A score decomposable models is equivalent to:
 - AIC for MRFs
 - A set of log-likelihoods by linear iteration

Most scores are scalable
 Entropy [1] ✓
 Softmax Ladder [2] ✓ Because it is unrolled when entropy is used
 Gradient [3] ✓
 Max. FIM [4] [5] ✓

Break

Efficient search

Scoring in greedy search

Clique graph (CG)

Clique graph and greedy search

Search and statistical paradigm

The nitty-gritty

Counting efficiently
 Scoring: for example with KL, minimized when...
 What does it mean to compute $\prod_{i,j} \theta_{ij}^{x_{ij}}$?
 = efficient search built upon efficient counting

Counting efficiently (2)
 Many different counting techniques require making use of the CG

Memorization
 From the high multiplicity of the CG, we can derive a more compact representation of the CG

Addition of the same edge to different reference models
 What we have seen so far: Counting the addition of an edge into a graph

How fast can we get?

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell

1. Graphical models are everywhere
2. We can learn graphical models from data with 1,000+ variables
3. It is crucial to be able to use the library that we are providing on your tablet device
4. There is still so much work to be done

Open problems

1. Efficient constrained search
2. Better scores (eg on Directed scoring on IJ)
3. Efficient scoring of marginal "bits"
4. Efficient data structures for counting
5. Learning out of core
6. Latent variables

Open problems (2)

1. How to handle numerical variables
2. How to handle missing values?
3. Learning accurate parameters in large tables

We hope that you enjoyed our tutorial on...
 Scalable learning of graphical models
 François Fleuret and Geoff Gordon

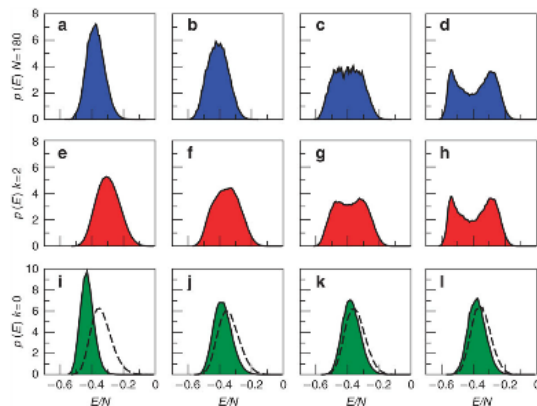
What is a probabilistic graphical model?

Probability theory

analysis called classical continuous convergence
defined discrete distribution Edit event
example function independent large law list mathematical
measure modern number occur
probability random sample
space statistics theorem theory value
variables



Quantifying uncertainty

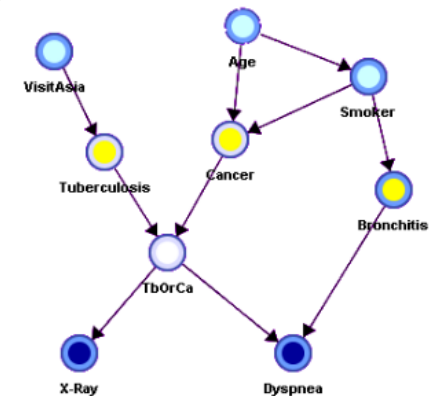


Graph Theory

algebra algorithm analysis applications called class coloring
computer connected data drawing edges example
finding generalized given **graph** list mathematics
matrix model networks number Press problem properties
related represent software structure study subgraphs
systems theory topology trees type used vertex
vertices



Not a black box +
Efficient algorithms



Aim: Compactly representing probability distributions

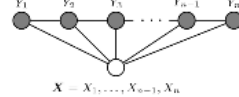
What are graphical models useful for?

Studying correlations & independencies



Simultaneously predicting multiple variables

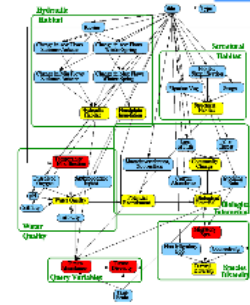
Hidden Markov Models (HMM),
Maximum Entropy Markov Models (MEMM),
Conditional Random Fields (CRF),
Dense Random Fields (DRF),
...



of...

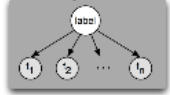
... the next sequence of words
... the class of sets of pixels
...

Causal Discovery & Inference



Classification

Naive Bayes

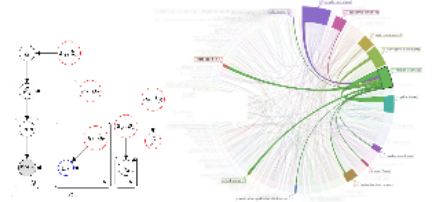


TAN



KDB, AODE, AnDE, ...

Topic Modelling

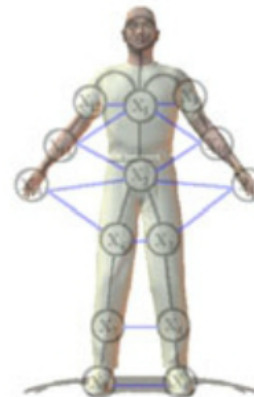
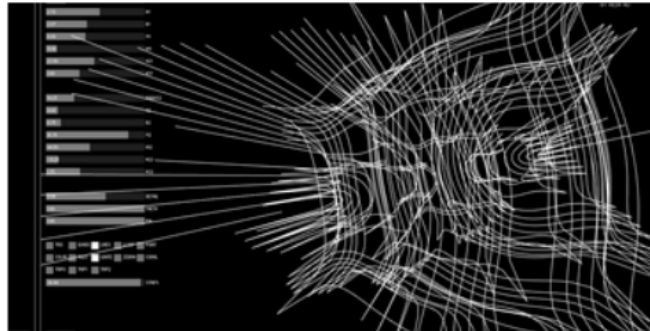
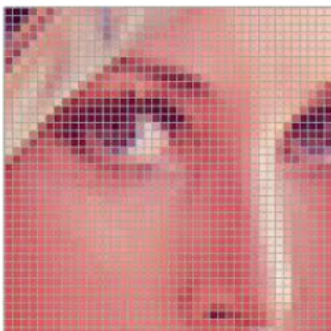


(c) Buntine and Mishra @KDD'14

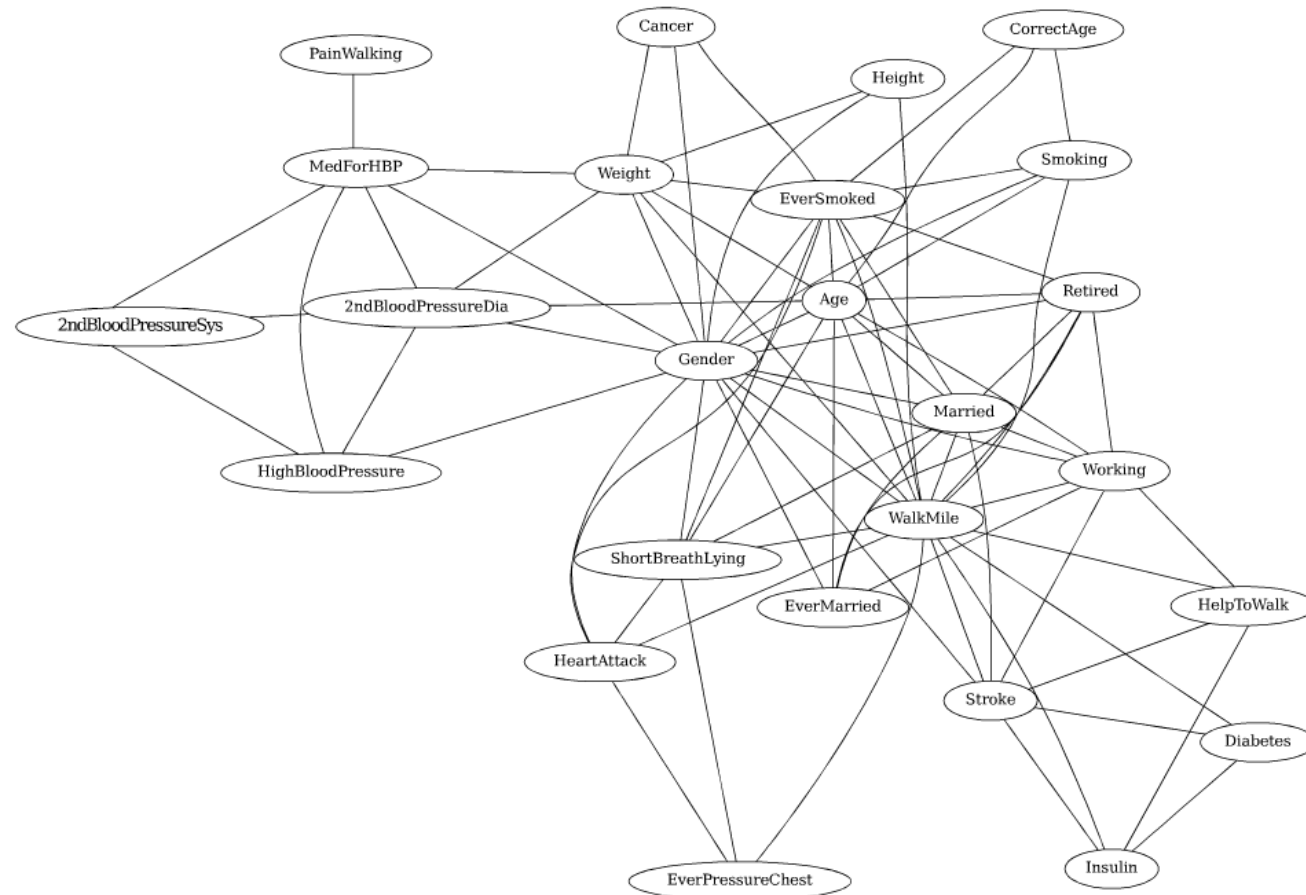


...

+ the thousands of applications of these methods...

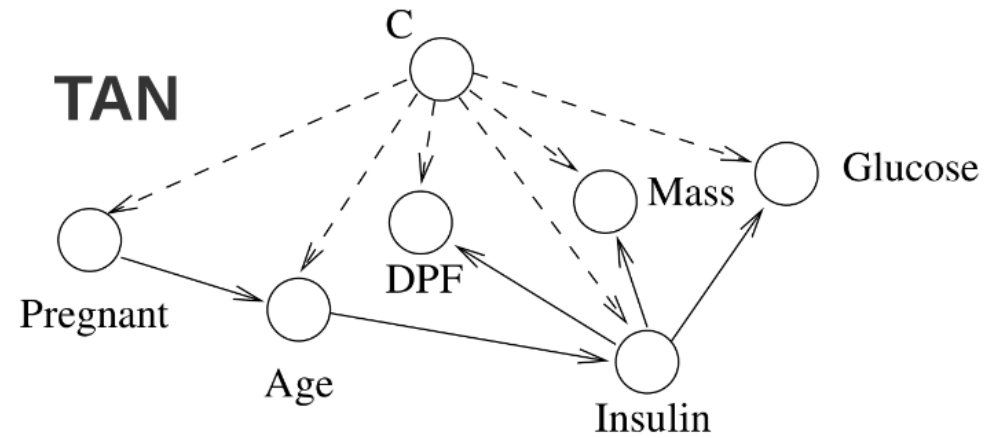
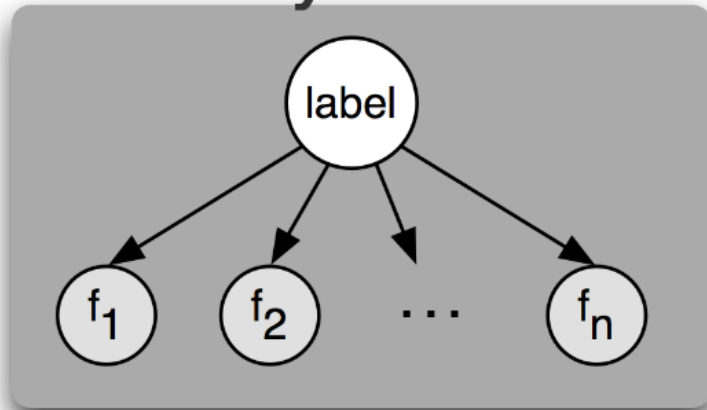


Studying correlations & independencies



Classification

Naive Bayes

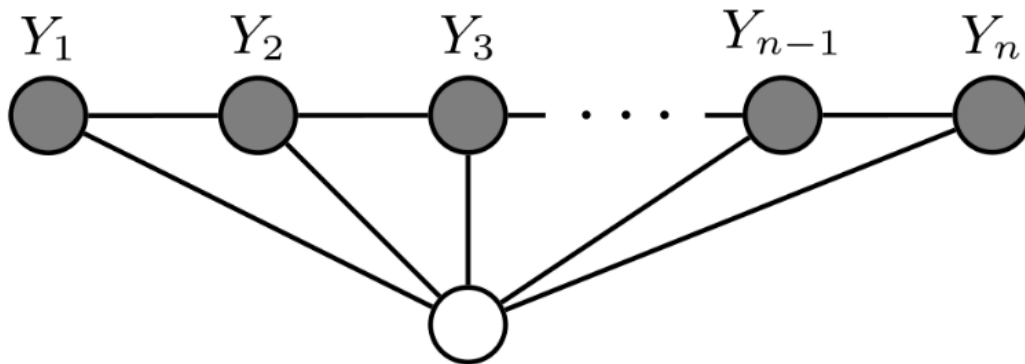


KDB, AODE, AnDE, ...

Simultaneously predicting multiple variables

Hidden Markov Models (HMM),
Maximum Entropy Markov Models (MEMM),
Conditional Random Fields (CRF),
Dense Random Fields (DRF),
...

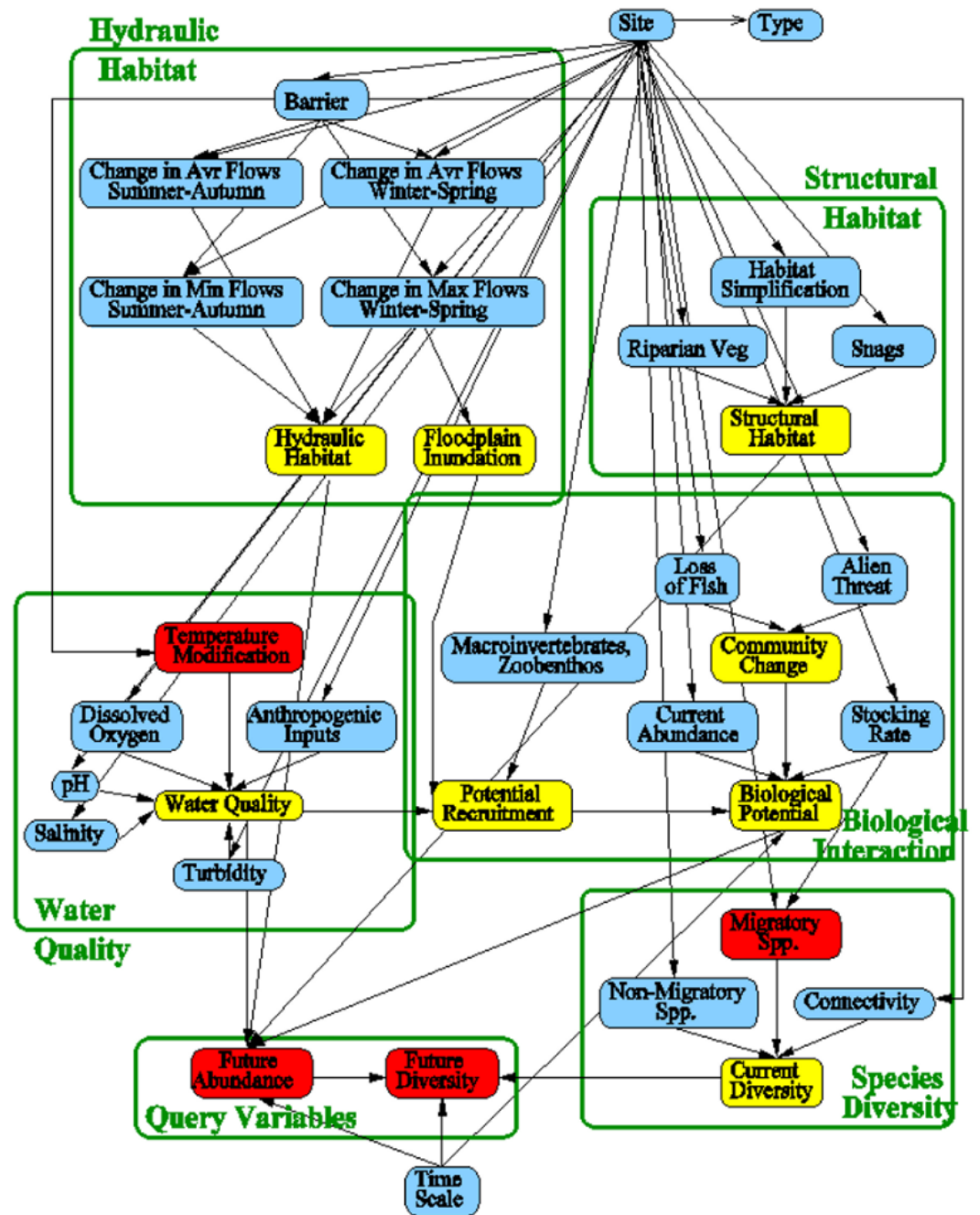
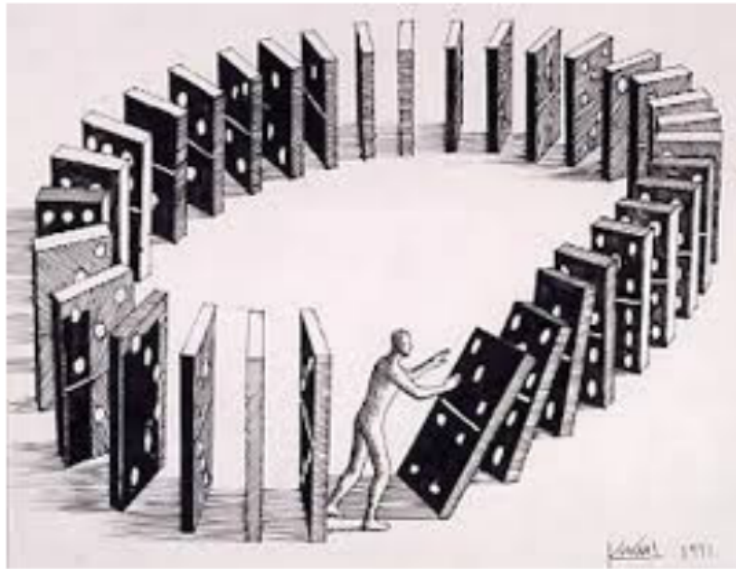
of...



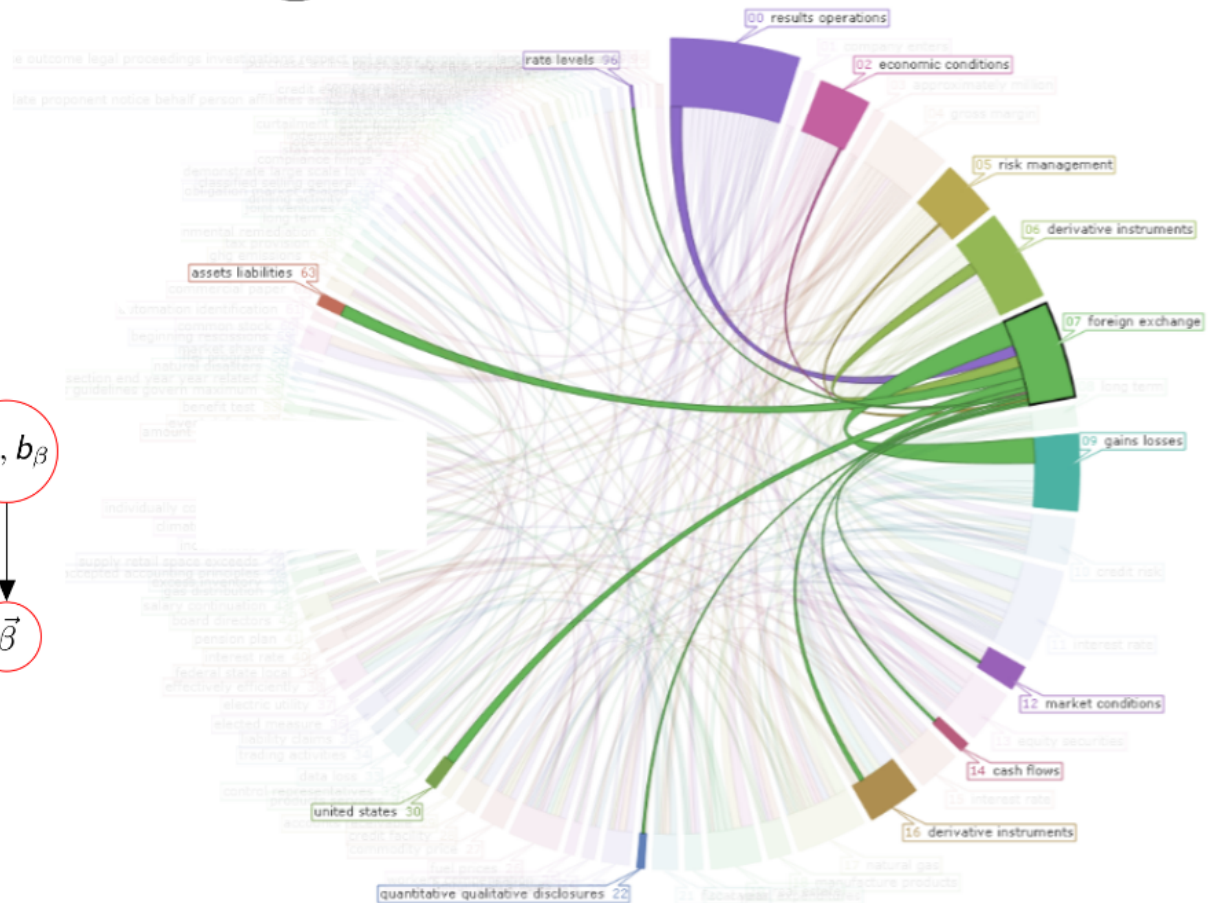
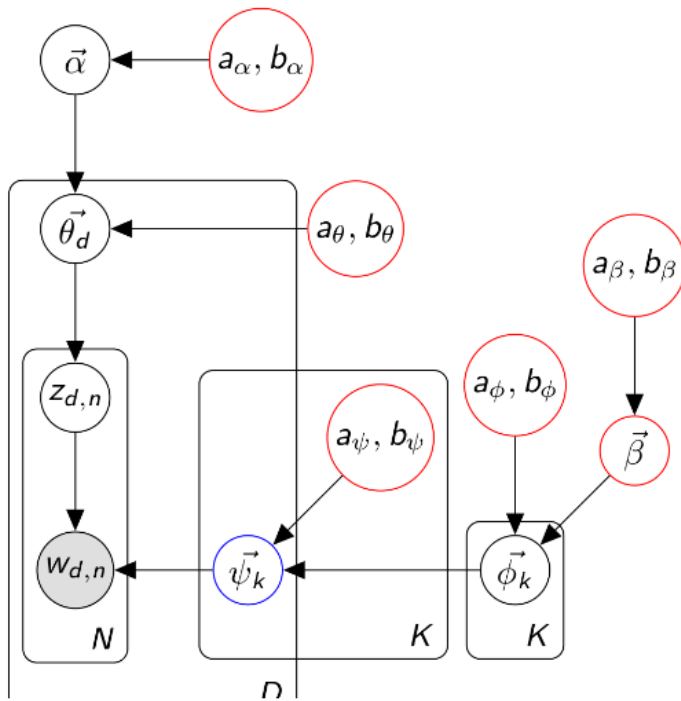
$$\mathbf{X} = X_1, \dots, X_{n-1}, X_n$$

... the next sequence
of words
... the class of sets of
pixels
...

Causal Discovery & Inference



Topic Modelling



(c) Buntine and Mishra @KDD'14

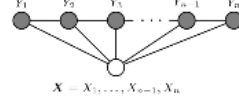
What are graphical models useful for?

Studying correlations & independencies



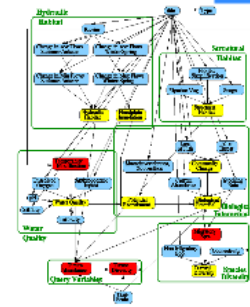
Simultaneously predicting multiple variables

Hidden Markov Models (HMM),
Maximum Entropy Markov Models (MEMM),
Conditional Random Fields (CRF),
Dense Random Fields (DRF),
...



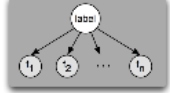
of...
... the next sequence of words
... the class of sets of pixels
...

Causal Discovery & Inference



Classification

Naive Bayes

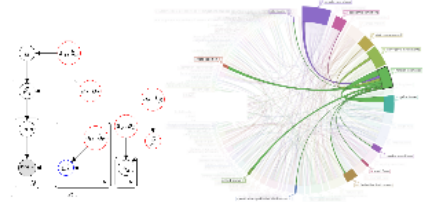


TAN



KDB, AODE, AnDE, ...

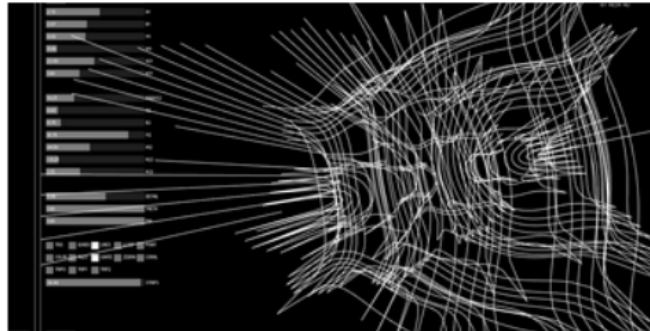
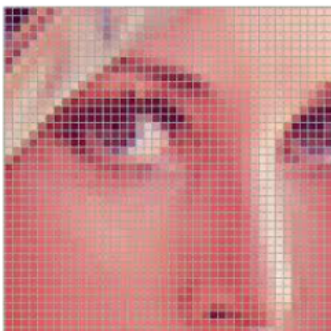
Topic Modelling



(c) Buntine and Mishra @KDD'14



+ the thousands of applications of these methods...



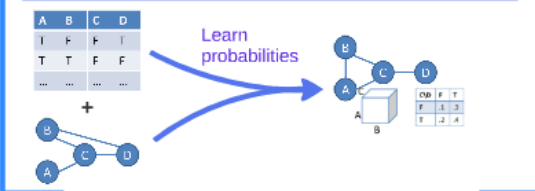
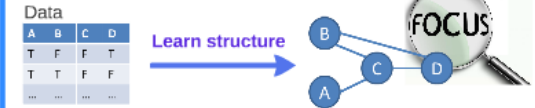
What we will and will not cover

Discrete case

- Neat problem definitions
- Already very challenging
- Handling numerical variables with discretisation



Structure and parameters



What's
IN

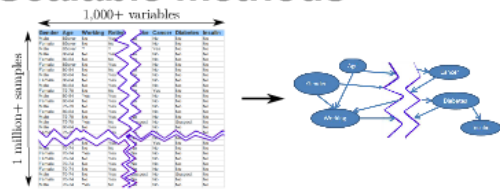


What's



OUT

Scalable methods



if #variables < 30 then
use [1]
end if

if #samples < 500 then
use model averaging [2,3]
// model selection not really relevant
end if

[1] T. Silander, A simple approach for finding the globally optimal Bayesian network structure
UAI 2006 - <http://arxiv.org/abs/2006.0875>
[2] J. Hoeting, Bayesian Model Averaging: A Tutorial, Statistical Science 1999
[3] B.M. Broom, Model averaging strategies for structure learning. BMC Bioinformatics 2012

Modelling the joint distribution

Joint discrete distribution

$$\rightarrow P(X_1 = x_1, \dots, X_n = x_n)$$

Modelling conditional distribution

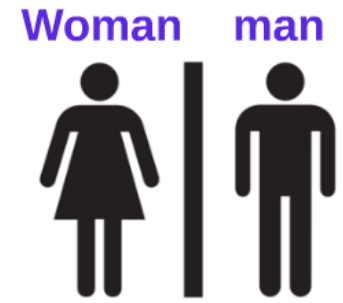
- **Open problem** with intense research effort
- Possible to use the joint to **approximate** a structure that models the conditional (eg TAN [1])



[1] N. Friedman, Bayesian Network Classifiers, Machine Learning, 1997.

Discrete case

- Neat problem definitions
- Already very challenging
- Handling numerical variables with discretisation

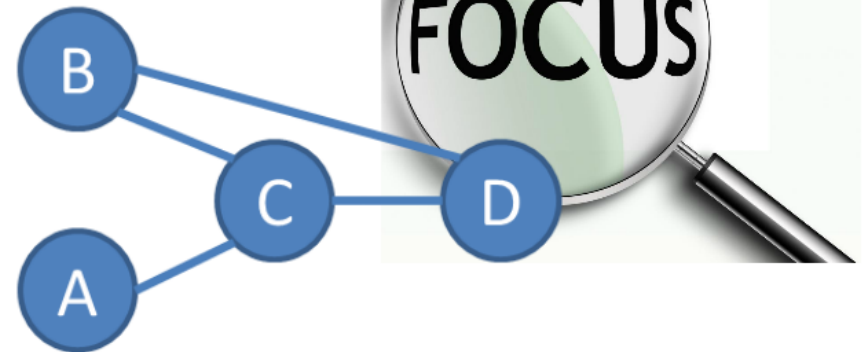


Structure and parameters

Data

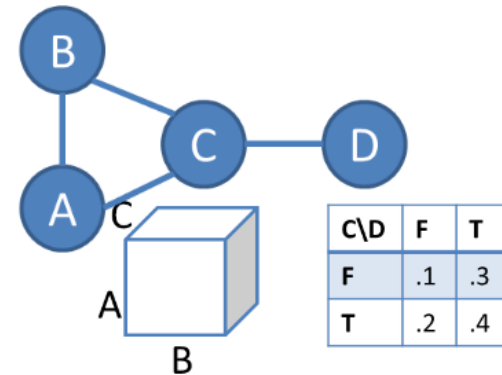
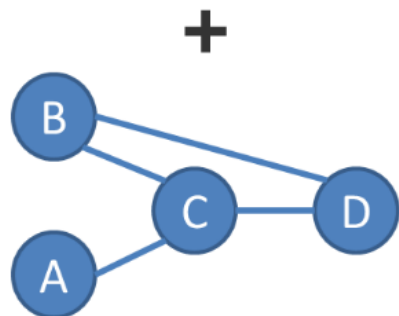
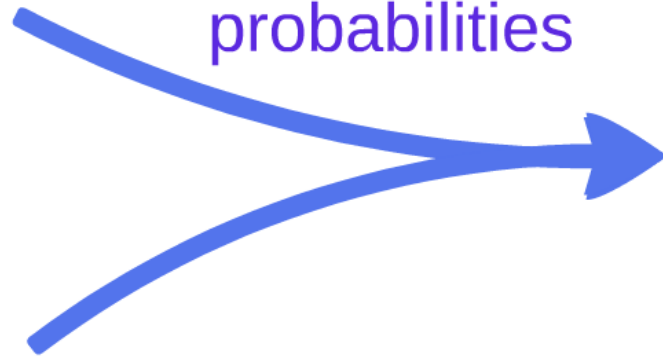
A	B	C	D
T	F	F	T
T	T	F	F
...

Learn structure



A	B	C	D
T	F	F	T
T	T	F	F
...

Learn probabilities

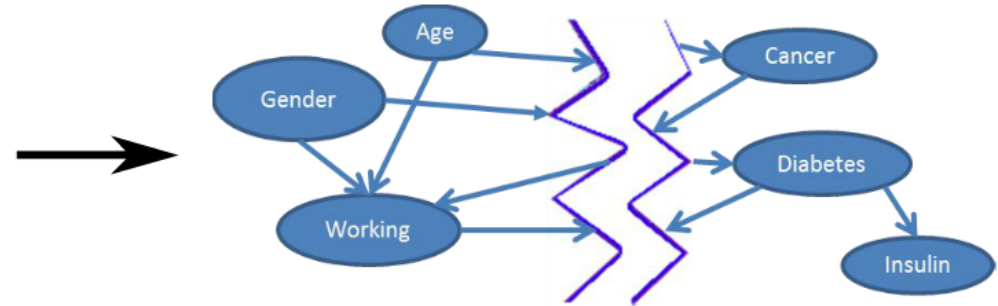


Scalable methods

1,000+ variables

1 million+ samples

Gender	Age	Working	Retire	Smoke	Cancer	Diabetes	Insulin
Male	85over	No	Yes	No	No	No	No
Female	85over	No	No	No	No	No	No
Male	85over	?	?	No	Yes	No	No
Male	80-84	No	Yes	No	No	No	No
Female	80-84	No	No	No	No	No	No
Female	85over	No	Yes	No	No	No	No
Female	80-84	No	No	No	No	No	No
Male	80-84	No	Yes	No	No	No	No
Female	80-84	No	Yes	No	No	No	No
Male	80-84	No	Yes	No	No	No	No
Female	75-79	No	No	No	Yes	No	No
Male	80-84	Yes	Yes	No	No	No	No
Female	80-84	No	Yes	No	No	No	No
Male	75-79	No	Yes	No	No	No	No
Female	80-84	No	Yes	No	No	No	No
Male	75-79	No	Yes	No	No	No	No
Male	75-79	Yes	No	Suspect	No	Suspect	No
Male	80-84	No	Yes	No	No	No	No
Female	75-79	No	Yes	No	No	No	No
Male	70-74	No	Yes	No	Yes	No	No
Male	70-74	No	No	No	No	No	No
Female	70-74	Yes	Yes	No	No	No	No
Female	70-74	No	Yes	No	No	No	No
Female	70-74	No	Yes	No	No	No	No
Female	70-74	No	Yes	No	No	No	No
Male	70-74	No	Yes	Suspect	No	Suspect	No
Female	70-74	No	Yes	No	No	No	No
Male	70-74	Yes	No	No	No	No	No



if #variables < 30 **then**
 use [1]
end if

if #samples < 500 **then**
 use model averaging [2,3]
 // model selection not really relevant
end if

[1] T.Silander, A simple approach for finding the globally optimal Bayesian network structure
UAI 2006 - <http://arxiv.org/abs/1206.6875>

[2] J. Hoeting, Bayesian Model Averaging: A Tutorial, Statistical Science 1999

[3] B.M. Broom, Model averaging strategies for structure learning. BMC Bioinformatics 2012

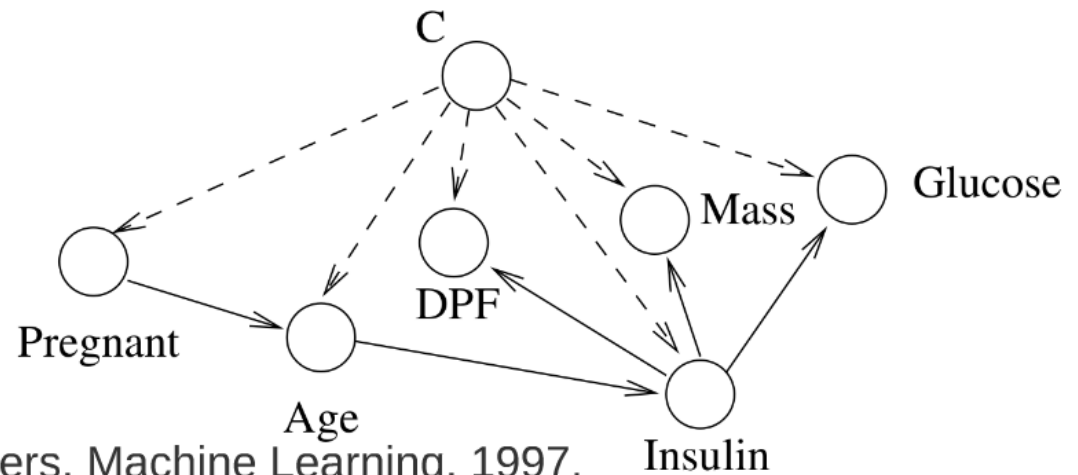
Modelling the joint distribution

Joint discrete distribution

$$\longrightarrow P(X_1 = x_1, \dots, X_n = x_n)$$

Modelling conditional distribution

- **Open problem** with intense research effort
- Possible to use the joint to **approximate** a structure that models the conditional (eg TAN [1])



[1] N. Friedman, Bayesian Network Classifiers, Machine Learning, 1997.

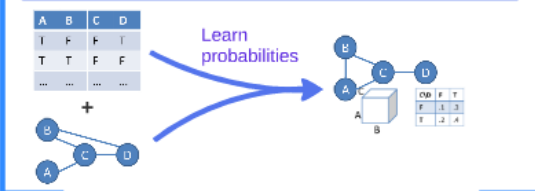
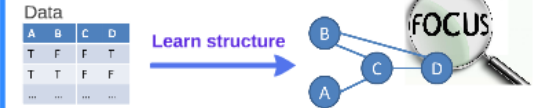
What we will and will not cover

Discrete case

- Neat problem definitions
- Already very challenging
- Handling numerical variables with discretisation



Structure and parameters



What's
IN

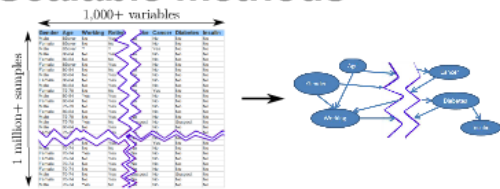


What's



OUT

Scalable methods



if #variables < 30 then
use [1]
end if

if #samples < 500 then
use model averaging [2,3]
// model selection not really relevant
end if

[1] T. Silander, A simple approach for finding the globally optimal Bayesian network structure
UAI 2006 - <http://arxiv.org/abs/2006.0875>
[2] J. Hoeting, Bayesian Model Averaging: A Tutorial, Statistical Science 1999
[3] B.M. Broom, Model averaging strategies for structure learning. BMC Bioinformatics 2012

Modelling the joint distribution

Joint discrete distribution

$$\rightarrow P(X_1 = x_1, \dots, X_n = x_n)$$

Modelling conditional distribution

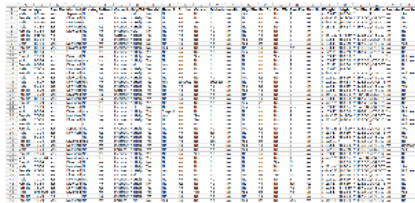
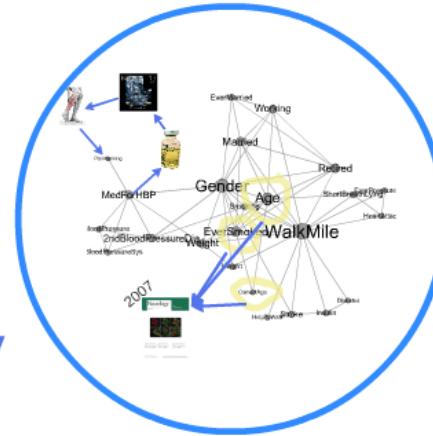
- **Open problem** with intense research effort
- Possible to use the joint to **approximate** a structure that models the conditional (eg TAN [1])



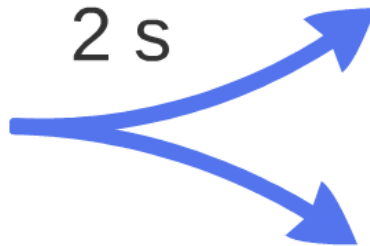
[1] N. Friedman, Bayesian Network Classifiers, Machine Learning, 1997.

Study of the elderly

- 25 variables
- 15,000 patients



2 s



Belief propagation

New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.

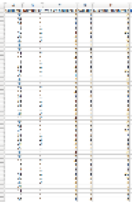


evidence	stroke	diabetes	heart attack
female under 70	5%	15%	10%
+ married	5%	15%	9%
+ smoking	7%	17%	12%
+ BP=17/10	8%	17%	13%
+ no help to walk	5%	16%	12%
+ quit smoking?	4%	14%	9%



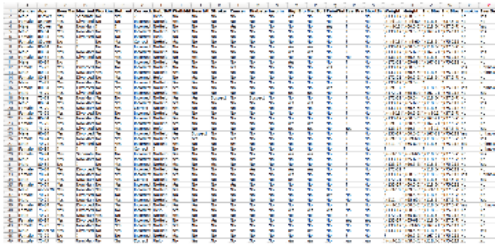
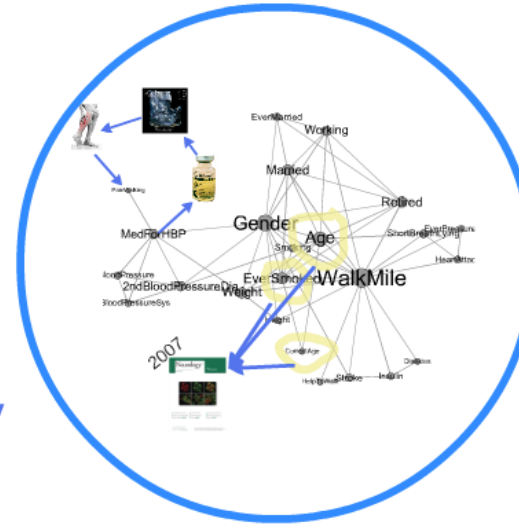
Insu

-
-

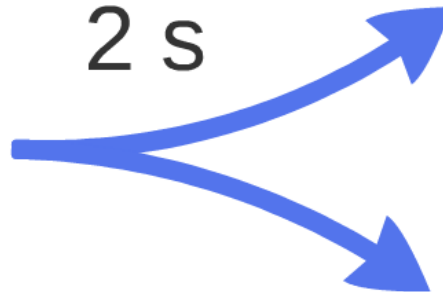


Study of the elderly

- 25 variables
- 15,000 patients



2 s



Belief propagation

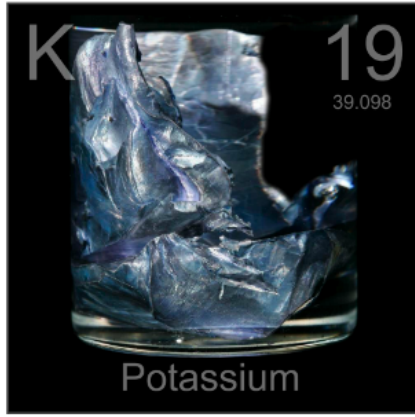
New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.



evidence	stroke	diabetes	heart attack
female under 70	5%	15%	10%
+ married	5%	15%	9%
+ smoking	7%	17%	12%
+ BP=17/10	8%	17%	13%
+ no help to walk	5%	16%	12%
+ quit smoking?	4%	14%	9%



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Gender	Age	EverMar	Married	Working	Retired	Correct	HelpToV	WalkMil	HeartAt	Stroke	Cancer	Diabete	Insulin	HighBlo	MedFor	PainWa	EverPre	ShortBr	Weight	Height	2ndBLoc	2ndBLoc	Smokin	EverSmo
2	Male	85over	Yes	Separate	No	Yes	?	Help	No	No	No	No	No	No	Yes	No	No	?	No	\(133-17	\(60.5-65	?	?	No	Yes
3	Female	85over	Yes	Divorced	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(37.5-75	No	No
4	Male	85over	Yes	NowMarr	?	?	Incorrect	NoHelp	No	No	No	Yes	No	No	No	No	Yes	?	No	\(-inf-13:	\(60.5-65	\(118.5-1	\(37.5-75	No	Yes
5	Male	80-84	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(133-17	\(69.5-in	\(167-21	\(75-112	No	Yes
6	Female	80-84	Yes	Divorced	No	No	Incorrect	Help	No	No	No	No	No	No	No	No	No	?	No	?	?	?	?	No	No
7	Female	85over	No	?	No	Yes	Correct	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(75-112	No	No
8	Female	80-84	No	?	No	No	Incorrect	NoHelp	No	No	No	No	No	No	Yes	Yes	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(37.5-75	No	No
9	Male	80-84	Yes	NowMarr	No	Yes	Incorrect	NoHelp	No	No	No	No	No	No	Yes	Yes	No	?	No	\(133-17	\(65-69.5	\(167-21	\(75-112	No	Yes
10	Female	80-84	Yes	Divorced	No	Yes	Incorrect	NoHelp	No	Yes	No	No	No	No	No	No	No	No	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
11	Male	80-84	No	?	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(172-21	\(65-69.5	\(167-21	\(75-112	No	No
12	Female	75-79	Yes	Divorced	No	No	Incorrect	NoHelp	No	No	No	Yes	No	No	No	No	Yes	?	No	\(133-17	\(60.5-65	?	?	Yes	Unknown
13	Male	80-84	Yes	NowMarr	Yes	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	Yes	Unknown
14	Female	80-84	Yes	Divorced	No	Yes	Incorrect	Help	No	Yes	No	No	No	No	No	No	No	?	No	\(172-21	\(60.5-65	\(118.5-1	\(37.5-75	No	No
15	Male	75-79	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	Yes	No	No	No	No	No	No	No	?	No	\(133-17	\(69.5-in	?	?	No	No
16	Female	80-84	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	Yes	?	No	\(-inf-13:	?	\(118.5-1	\(75-112	Yes	Unknown
17	Male	75-79	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(172-21	\(65-69.5	\(167-21	\(75-112	No	Yes
18	Male	75-79	Yes	NowMarr	Yes	No	Incorrect	NoHelp	Yes	Yes	Suspect	No	Suspect	No	No	No	No	?	No	\(172-21	\(69.5-in	\(118.5-1	\(37.5-75	No	No
19	Male	80-84	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(133-17	\(65-69.5	\(118.5-1	\(75-112	No	Yes
20	Female	75-79	Yes	NowMarr	No	Yes	Correct	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
21	Female	75-79	Yes	Divorced	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(60.5-65	\(-inf-111	\(37.5-75	No	No
22	Female	80-84	Yes	NowMarr	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(167-21	\(37.5-75	No	No
23	Female	75-79	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(-inf-13:	\(60.5-65	\(167-21	\(37.5-75	No	No
24	Male	75-79	Yes	Divorced	No	Yes	Incorrect	Help	No	No	No	No	No	No	Yes	Yes	No	Yes	No	\(133-17	\(69.5-in	\(-inf-111	\(37.5-75	No	Yes
25	Male	80-84	Yes	Divorced	Yes	Yes	Incorrect	NoHelp	Yes	Suspect	No	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(167-21	\(75-112	Yes	Unknown
26	Female	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	Yes	No	Yes	No	No	No	No	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(37.5-75	No	Yes
27	Male	70-74	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	Yes	No	No	No	No	No	?	No	\(211-inf	\(65-69.5	\(118.5-1	\(75-112	Yes	Unknown
28	Female	80-84	?	?	?	?	Correct	?	No	No	No	No	No	No	No	No	No	?	No	?	?	?	?	?	Unknown
29	Female	70-74	Yes	Separate	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(37.5-75	No	Yes
30	Male	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	Yes	No	No	No	No	?	No	\(133-17	\(65-69.5	\(118.5-1	\(37.5-75	No	No
31	Male	80-84	No	?	No	Yes	Incorrect	NoHelp	Yes	No	Yes	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	Yes
32	Female	70-74	Yes	Divorced	Yes	Yes	Incorrect	NoHelp	Yes	No	No	No	Yes	No	Yes	Yes	No	?	No	\(172-21	?	\(118.5-1	\(75-112	No	No
33	Female	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	Yes	Unknown
34	Male	70-74	Yes	NowMarr	No	Yes	Correct	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(172-21	\(65-69.5	\(118.5-1	\(37.5-75	No	Yes
35	Female	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	Yes	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
36	Male	under70	Yes	NowMarr	Yes	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(69.5-in	\(-inf-111	\(37.5-75	No	No
37	Female	70-74	Yes	Divorced	No	No	Incorrect	NoHelp	No	No	No	No	No	No	No	No	Yes	No	No	?	\(60.5-65	\(118.5-1	\(37.5-75	No	No
38	Male	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(211-inf	\(65-69.5	\(118.5-1	\(37.5-75	No	Yes
39	Female	80-84	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	Yes	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(60.5-65	\(118.5-1	\(37.5-75	Yes	Unknown
40	Female	75-79	Yes	Divorced	No	No	Incorrect	Help	Yes	No	No	No	No	No	Yes	Yes	Yes	No	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
41	Male	70-74	Yes	NowMarr	Yes	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(172-21	\(65-69.5	\(118.5-1	\(75-112	No	No
42	Female	80-84	Yes	Divorced	No	No	Incorrect	NoHelp	No	No	Yes	No	No	No	No	No	No	Yes	Yes	\(133-17	\(65-69.5	\(118.5-1	\(37.5-75	No	No
43	Female	75-79	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	No	No	\(133-17	\(-inf-60.	\(167-21	\(75-112	No	No
44	Male	under70	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	Yes	No	Yes	Yes	Yes	No	No	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	Yes
45	Female	75-79	No	?	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	?	\(60.5-65	\(167-21	\(75-112	No	No
46	Female	75-79	Yes	NowMarr	No	Yes	Correct	Help	No	Yes	No	No	No	No	Yes	Yes	No	Yes	Yes	\(-inf-13:	\(60.5-65	\(118.5-1	\(75-112	No	No



PainWalking

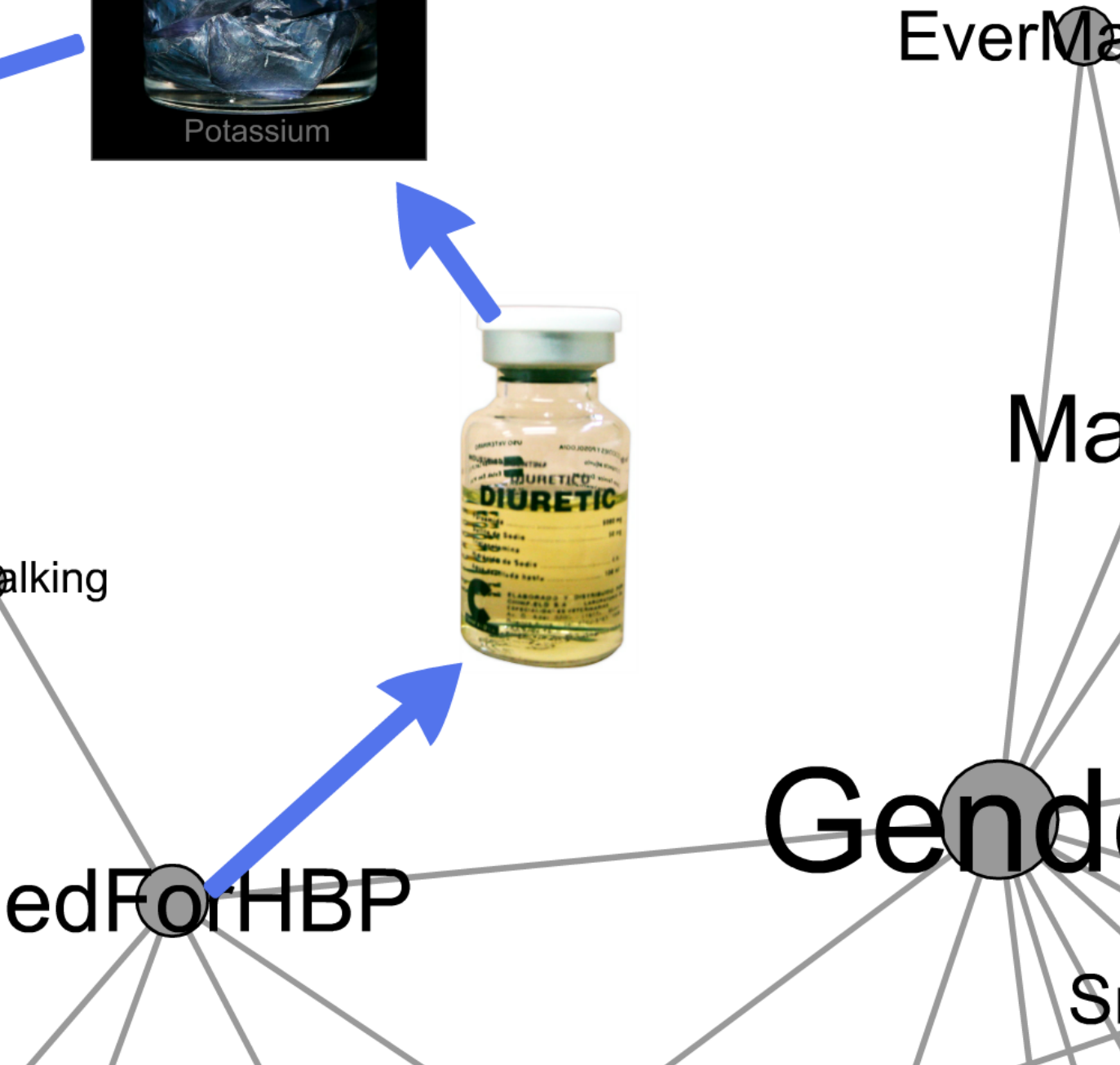
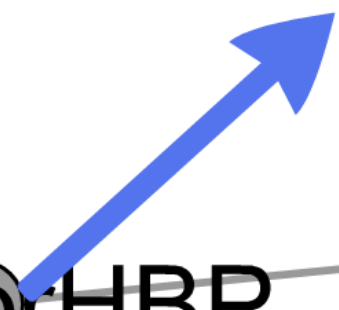
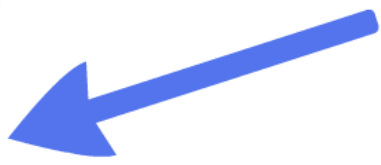
MedForHBP

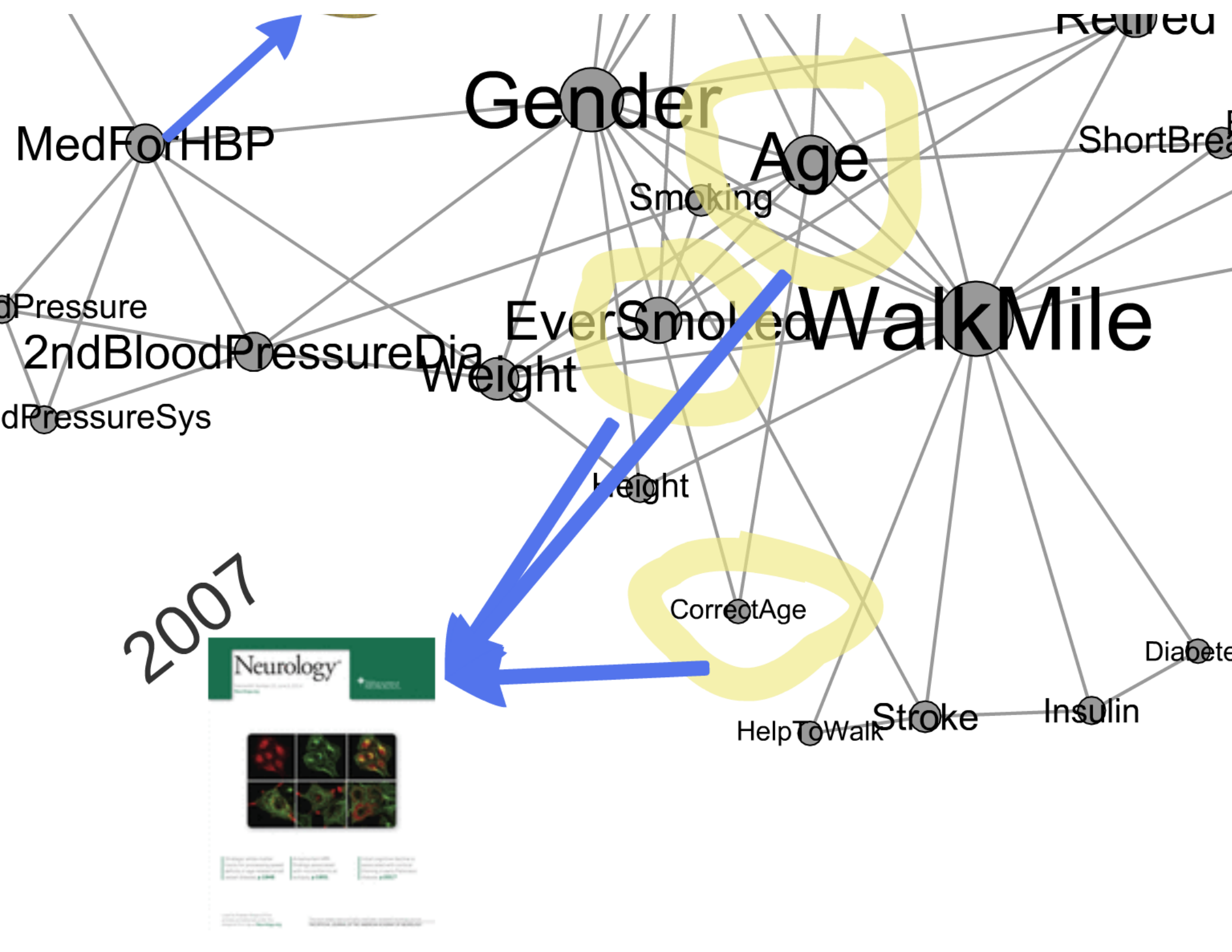
Gender

EverMa

Ma

S





Belief propagation

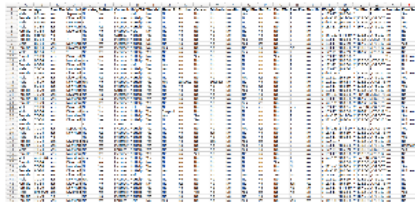
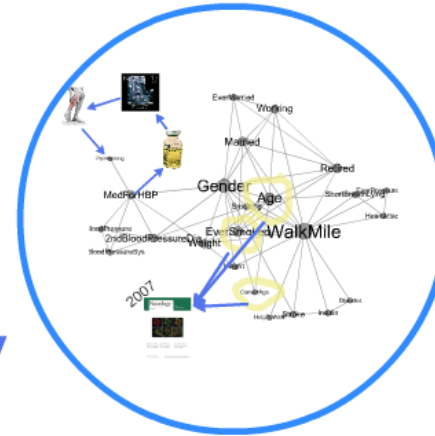
New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.



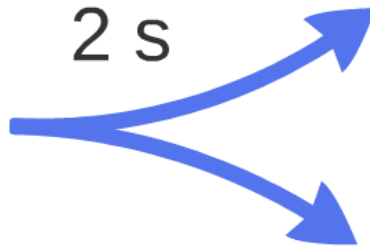
evidence	stroke	diabetes	heart attack
<i>female under 70</i>	5%	15%	10%
<i>+ married</i>	5%	15%	9%
<i>+ smoking</i>	7%	17%	12%
<i>+ BP=17/10</i>	8%	17%	13%
<i>+ no help to walk</i>	5%	16%	12%
<i>+ quit smoking?</i>	4%	14%	9%

Study of the elderly

- 25 variables
- 15,000 patients



2 s



Belief propagation

New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.

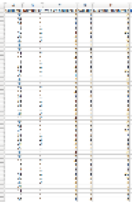


evidence	stroke	diabetes	heart attack
female under 70	5%	15%	10%
+ married	5%	15%	9%
+ smoking	7%	17%	12%
+ BP=17/10	8%	17%	13%
+ no help to walk	5%	16%	12%
+ quit smoking?	4%	14%	9%



Insu

-
-



Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probabilities: $P(x)$
 probability: $P(x_i)$
 Quantifying uncertainty: $P(x_i)$
 Not a trade-off in classical agent systems
 Aka: Compactly representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 In: $P(x)$
 Out: $P(x_i)$

Graphical models 101

Classes of graphical models

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data

Graph theory

Maximal cliques and minimal separators

What are decomposable models
 Decomposable models are a Markov Random Field for which the graph is chordal or triangulated

Properties of decomposable models

1. Closed form for $P(x)$
2. Very easy to optimize
3. Junction tree algorithms
4. No ∞ loops (local)
5. Linear-time inference
6. Interaction between IJ and MRF [2]

Useful algorithms

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:

1. scalable scoring
2. efficient search
3. scalable belief propagation

= all the results we will show here

Bottom line
 A score decomposable model is equivalent to:
 - A set of pairwise Bayesian networks
 - Any scoring function that has been developed for MRF in the literature can be used for decomposable models
 - MRF-based algorithms
 - MRF-based algorithms can be used and will work for the same of \mathcal{G}

Most scores are scalable
 Entropy [1] ✓
 Softmax L1/L2 [2] ✓
 G-soft [3] ✓
 Max-Ent [4] ✓

Break

Efficient search

Scoring in greedy search

Clique graph (CG)

Clique graph and greedy search

Search and statistical paradigm

The nitty-gritty

Counting efficiently
 Scoring: for example with KL, minimize when...
 What does it mean to compute $P(x_i)$?
 Efficient counting built on efficient counting

Counting efficiently (2)
 Many algorithms count by summing over all possible configurations of the variables in the graph

Memorization
 From high multiplicity to low multiplicity

Addition of the same edge to different reference models
 What we have seen so far: Changing the addition of edges into adding a subset of the graph

How fast can we get?

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell

1. Graphical models are everywhere
2. We can learn graphical models from data with 1,000+ variables
3. It is possible to do inference on graphs in the library that we are exploring in our selected class
4. There is still so much work to be done

Open problems

1. Efficient combinatorial search
2. Better scores (eg on Directed scoring on MRF)
3. Efficient scoring of marginal "bits"
4. Efficient data structures for counting
5. Learning out of core
6. Latent variables

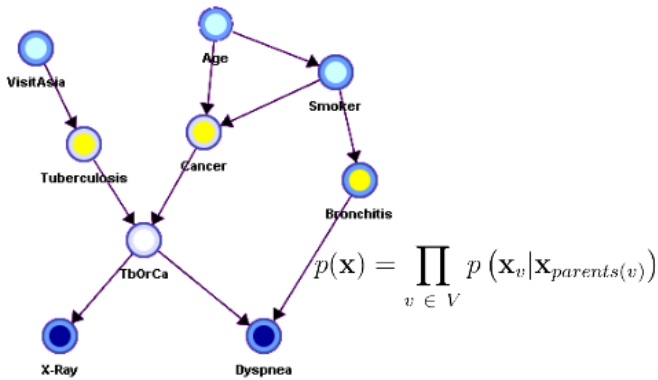
Open problems (2)

1. How to handle numerical variables
2. How to handle missing values?
3. Learning accurate parameters in large tables

We hope that you enjoyed our tutorial on...
 Scalable learning of graphical models
 François Fleuret and Geoff Gordon

Classes of graphical models

Bayesian Network



$$p(\mathbf{x}) = \prod_{v \in V} p(x_v | \mathbf{x}_{\text{parents}(v)})$$

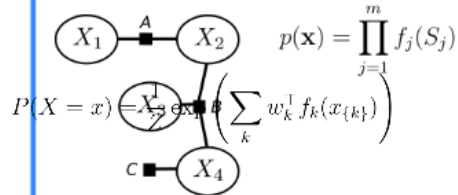
Possible causal interpretation

Markov networks or Markov Random Fields



Special case of log-linear models that have the property of being graphical

Factor graphs

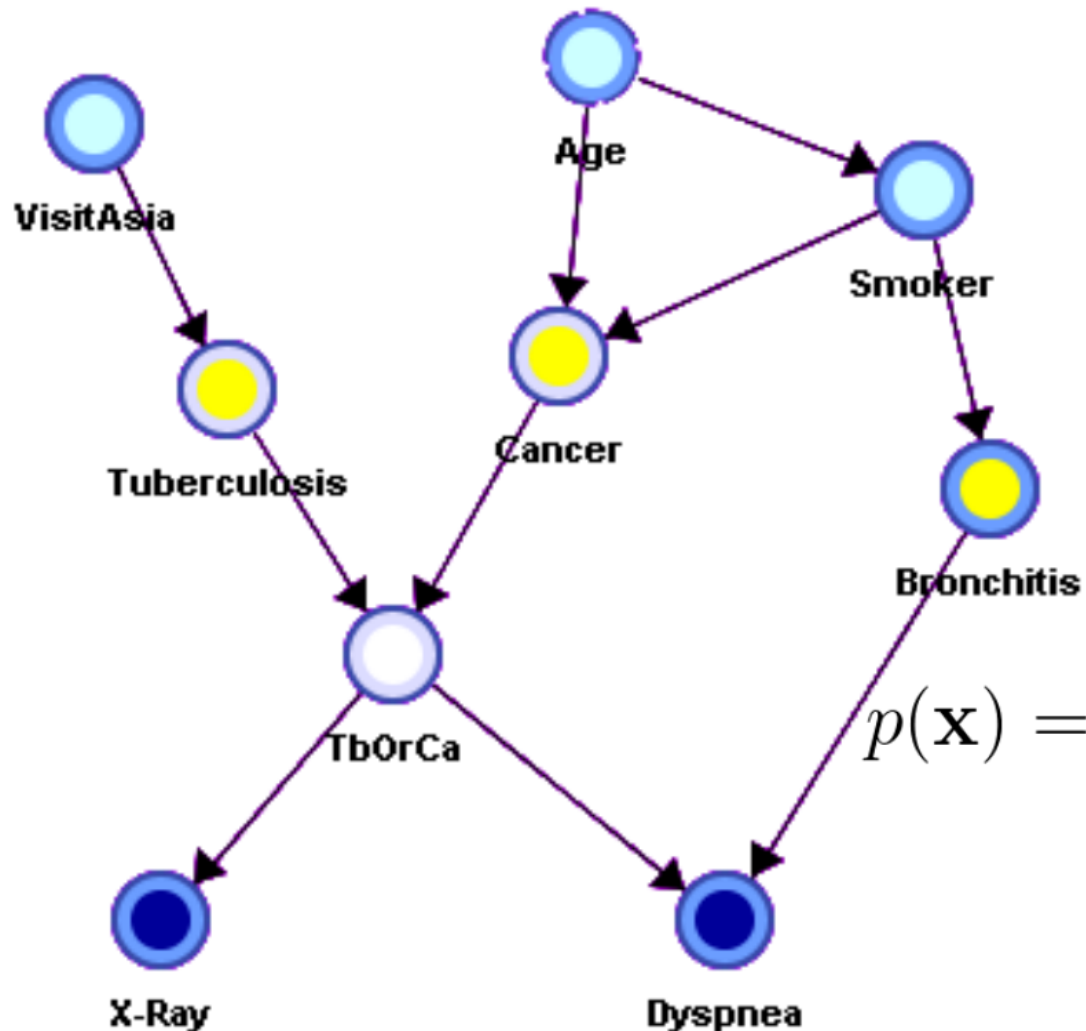


$$p(\mathbf{x}) = \prod_{j=1}^m f_j(S_j)$$

$$P(X = x) = \sum_{\mathbf{x}_{-i}} \prod_k w_k^i f_k(x_{\{k\}})$$



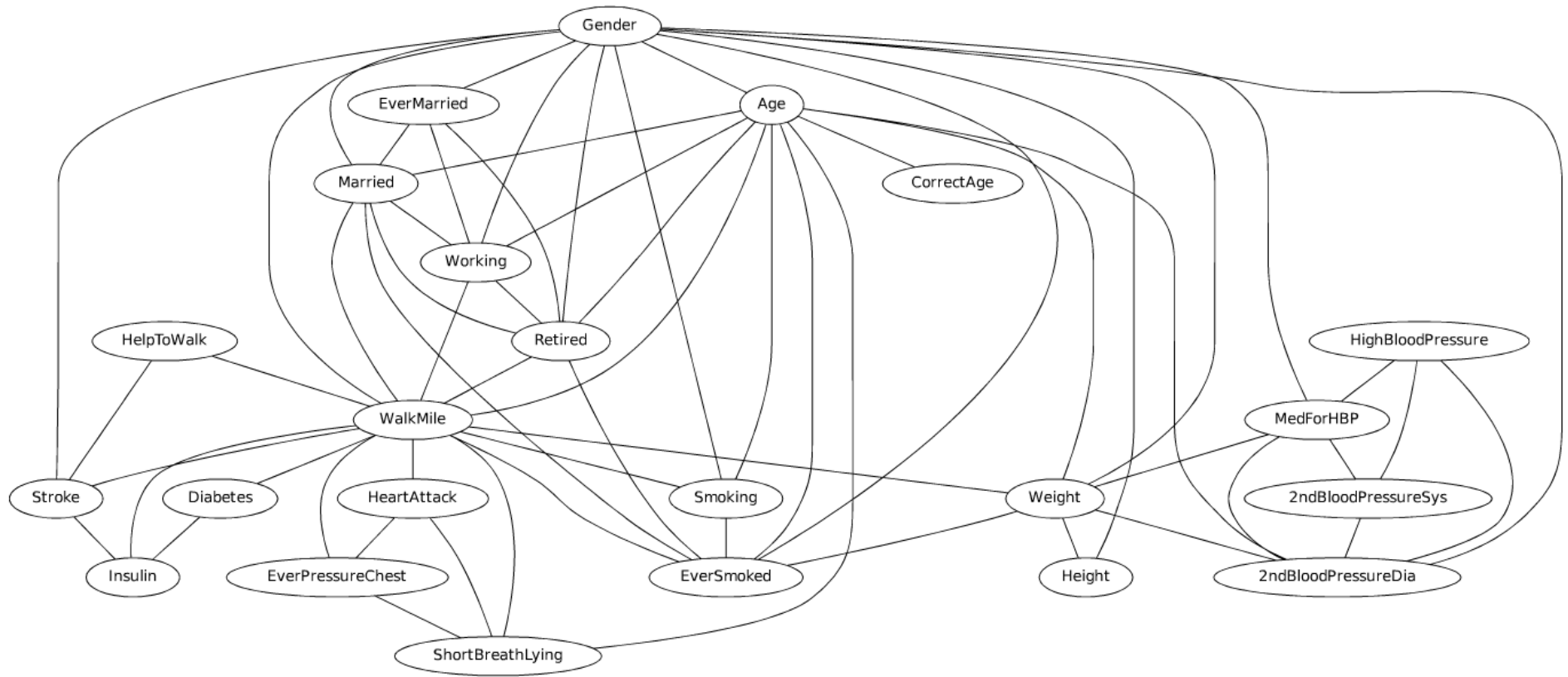
Bayesian Network



$$p(\mathbf{x}) = \prod_{v \in V} p(\mathbf{x}_v | \mathbf{x}_{\text{parents}(v)})$$

Possible causal interpretation

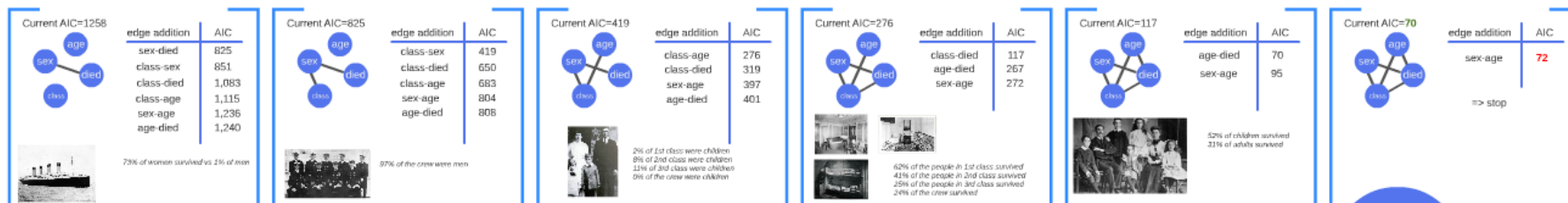
Markov networks or Markov Random Fields



Special case of log-linear models that have the property of being graphical

A simple example of structure learning

Hill-climbing search on MRF using AIC

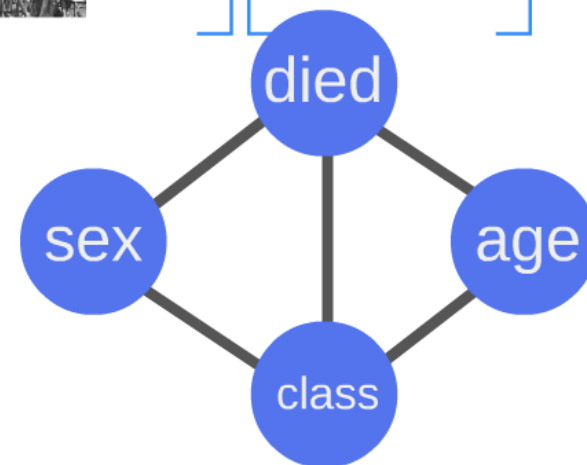


To redo this experiment
Just 4 lines of 

```

library(MarkovRandomField)
data Titanic
MRF = MRF(
  nodes = c("sex", "age", "class", "died"),
  edges = c("sex-died", "class-sex", "class-died", "class-age", "sex-age", "age-died")
)

```



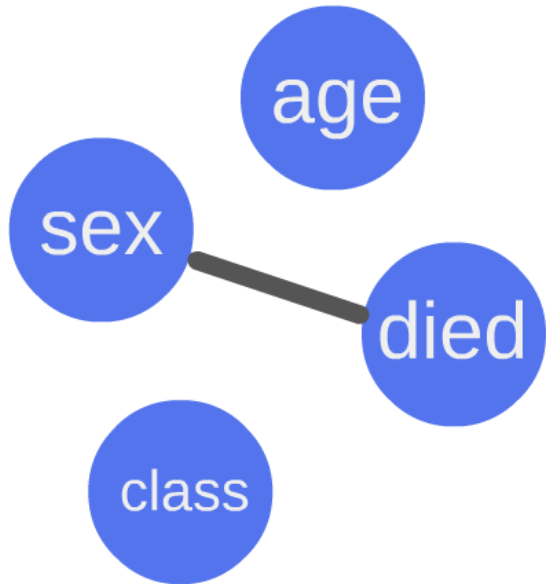
To predict survival:

- Yes, age matters
- Yes, class matters
- Yes, sex matters
- Yes, class and sex together matter (eg knowing that a particular man was in 1st class or crew)
- Yes, class and age together matter (eg knowing that a particular child was in 1st or 3rd class)
- No, sex and age don't matter together for a particular class (within each class, age and sex interact with survival independently of one another)

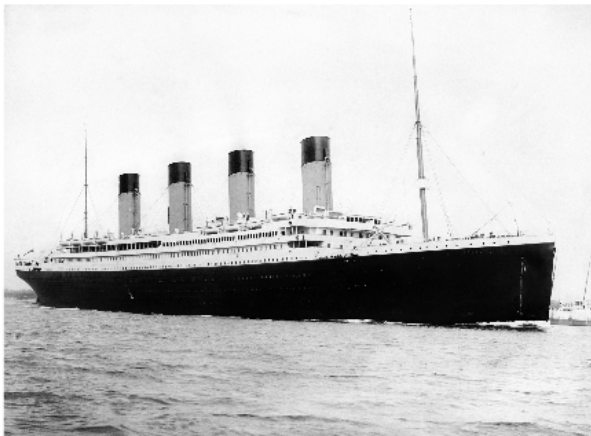


	A	B	C	D	E
1	Class	Sex	Age	Survived	Frequency
2	1st	Male	Child	No	0
3	2nd	Male	Child	No	0
4	3rd	Male	Child	No	35
5	Crew	Male	Child	No	0
6	1st	Female	Child	No	0
7	2nd	Female	Child	No	0
8	3rd	Female	Child	No	17
9	Crew	Female	Child	No	0
10	1st	Male	Adult	No	118
11	2nd	Male	Adult	No	154
12	3rd	Male	Adult	No	387
13	Crew	Male	Adult	No	670
14	1st	Female	Adult	No	4
15	2nd	Female	Adult	No	13
16	3rd	Female	Adult	No	89
17	Crew	Female	Adult	No	3
18	1st	Male	Child	Yes	5
19	2nd	Male	Child	Yes	11
20	3rd	Male	Child	Yes	13
21	Crew	Male	Child	Yes	0
22	1st	Female	Child	Yes	1
23	2nd	Female	Child	Yes	13
24	3rd	Female	Child	Yes	14
25	Crew	Female	Child	Yes	0
26	1st	Male	Adult	Yes	57
27	2nd	Male	Adult	Yes	14
28	3rd	Male	Adult	Yes	75
29	Crew	Male	Adult	Yes	192
30	1st	Female	Adult	Yes	140
31	2nd	Female	Adult	Yes	80
32	3rd	Female	Adult	Yes	76
33	Crew	Female	Adult	Yes	20

Current AIC=1258

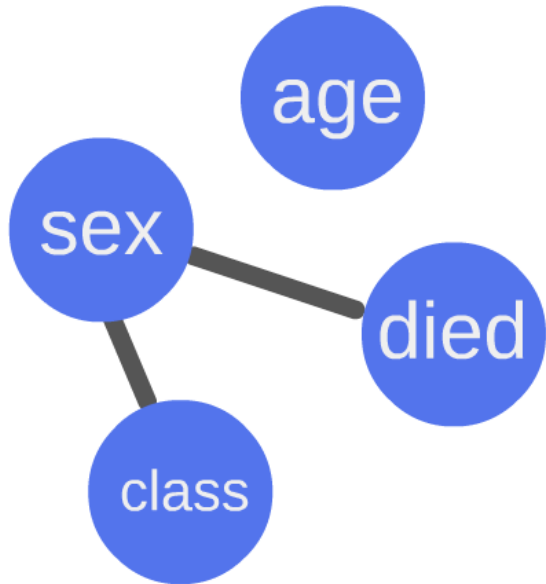


edge addition	AIC
sex-died	825
class-sex	851
class-died	1,083
class-age	1,115
sex-age	1,236
age-died	1,240



73% of women survived vs 1% of men

Current AIC=825

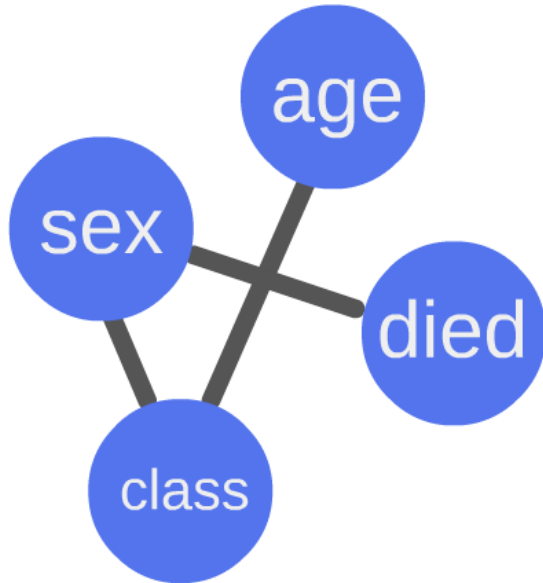


edge addition	AIC
class-sex	419
class-died	650
class-age	683
sex-age	804
age-died	808



97% of the crew were men

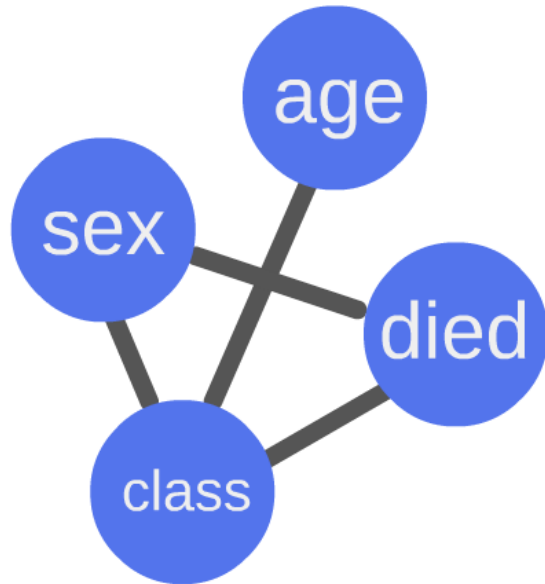
Current AIC=419



edge addition	AIC
class-age	276
class-died	319
sex-age	397
age-died	401

*2% of 1st class were children
8% of 2nd class were children
11% of 3rd class were children
0% of the crew were children*

Current AIC=276

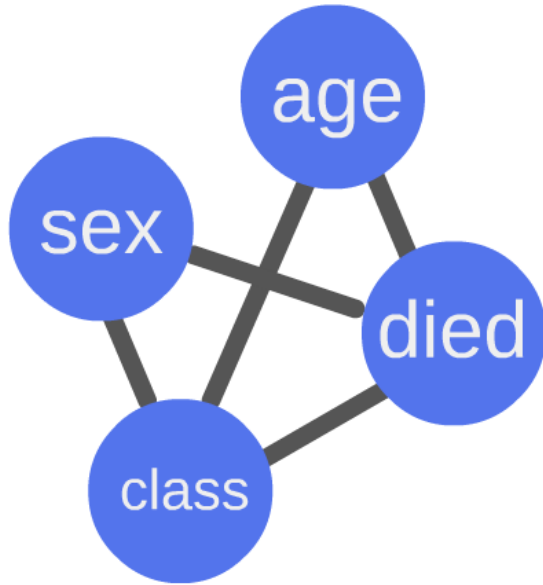


edge addition	AIC
class-died	117
age-died	267
sex-age	272



*62% of the people in 1st class survived
41% of the people in 2nd class survived
25% of the people in 3rd class survived
24% of the crew survived*

Current AIC=117

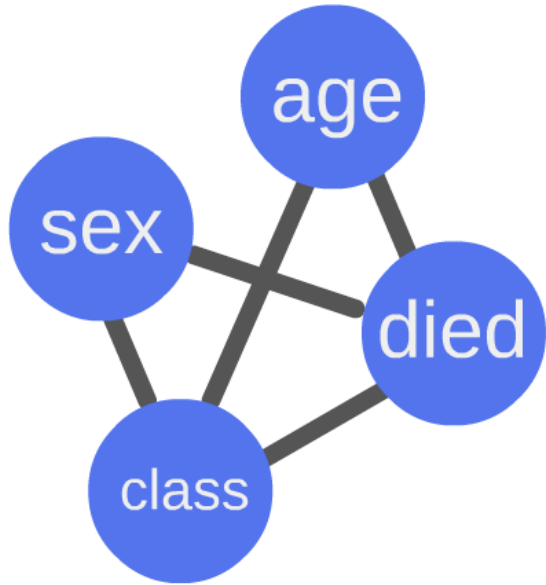


edge addition	AIC
age-died	70
sex-age	95



*52% of children survived
31% of adults survived*

Current AIC=70



edge addition

AIC

sex-age

72

=> stop

To redo this experiment

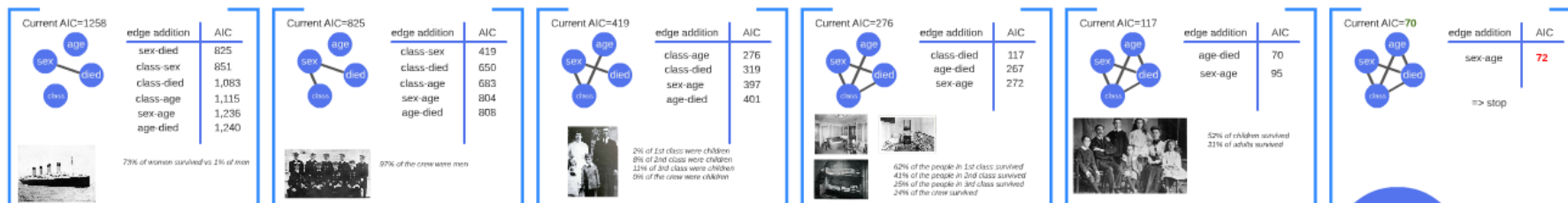
Just 4 lines of



```
> library(MASS)
> data(Titanic)
> independence=loglm(~Class+Sex+Survived+Age,data=Titanic)
> step(independence,scope="~.^2+.^3",direction="forward")
```

A simple example of structure learning

Hill-climbing search on MRF using AIC

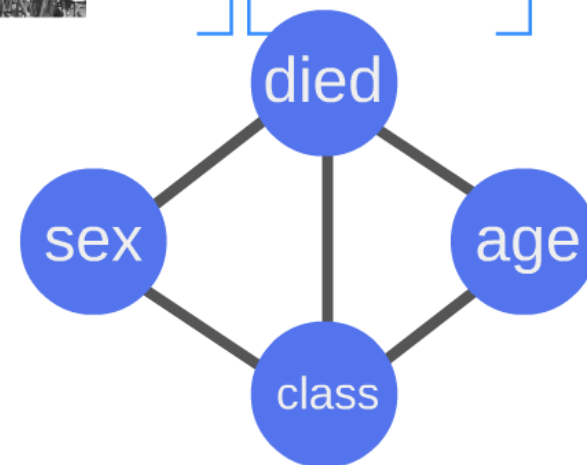


To redo this experiment
Just 4 lines of 

```

library(MarkovRandomField)
data Titanic
MRF = MRF(
  nodes = c("sex", "age", "class", "died"),
  edges = c("sex-died", "class-sex", "class-died", "class-age", "sex-age", "age-died")
)

```



To predict survival:

- Yes, age matters
- Yes, class matters
- Yes, sex matters
- Yes, class and sex together matter (eg knowing that a particular man was in 1st class or crew)
- Yes, class and age together matter (eg knowing that a particular child was in 1st or 3rd class)
- No, sex and age don't matter together for a particular class (within each class, age and sex interact with survival independently of one another)



Learning a model from data

Scoring



Bayesian approaches



Aim: Finding the model \mathcal{M} that, for a dataset \mathcal{D} maximizes $p(\mathcal{M}|\mathcal{D})$

$$p(\mathcal{M}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}) \times p(\mathcal{M})$$

Posterior probability \propto Likelihood \times Prior probability.

Hundreds of methods and references: BDeu, BD/BDe, MDL, NML, etc.; see details in [1,2].

[1] Koller and Friedman, Probabilistic Graphical Models, MIT Press, 2009 (esp. chapters 18 and 20)

[2] W. Buntine, A guide to the literature on learning probabilistic networks from Data, TRUC, 2005.

Frequentist approaches



Avoid the definition of priors

Also hundreds of methods and approaches using statistical tests (eg Chi-squared, likelihood-ratio tests).

→ See details in [1,2,3]

$$P(x) = \lim_{n \rightarrow \infty} \frac{\sum_{i=1}^n x_i}{n}$$

[1] Agresti, Categorical Data Analysis, Wiley, 2002.

[2] Koller and Friedman, Probabilistic Graphical Models, MIT Press, 2009 (esp. 18.2 and 20.7)

[3] Christensen, Log-linear models and logistic regression, 1996.

Search



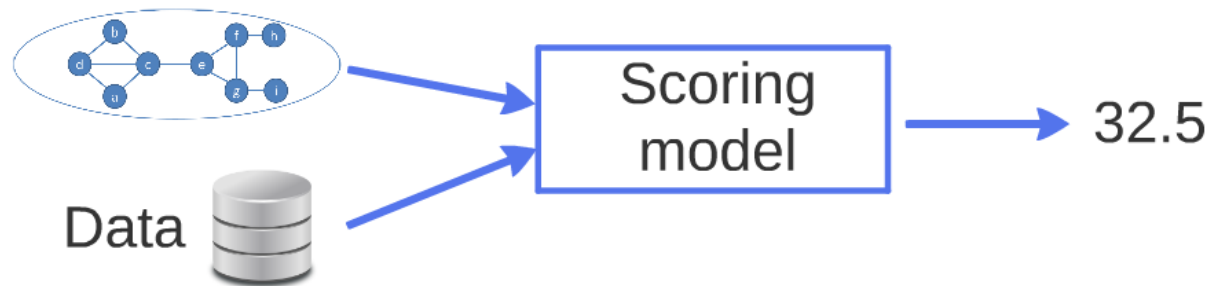
Traditional algorithms:

- local search (eg greedy, backward)
- simulated annealing
- genetic algorithms
- MCMC/Gibbs
- etc.

Note:

- BN: scores also require an order on the variables

Scoring



Bayesian approaches



Aim: Finding the model \mathcal{M} that, for a dataset \mathcal{D} maximizes $p(\mathcal{M}|\mathcal{D})$

$$p(\mathcal{M}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}) \times p(\mathcal{M})$$

Posterior probability \propto Likelihood \times Prior probability.

Hundreds of methods and references: BDeu, BD/BDe, MDL, NML, etc.; see details in [1,2].

[1] Koller and Friedman, Probabilistic Graphical Models, MIT Press, 2009 (esp. chapters 18 and 20)
[2] W. Buntine, A guide to the literature on learning probabilistic networks from Data, TKDE 1996.

Frequentist approaches



Avoid the definition of priors

Also hundreds of methods and approaches using statistical tests (eg Chi-squared, likelihood-ratio tests).

→ See details in [1,2,3]

$$P(\mathbf{x}) = \lim_{n_t \rightarrow \infty} \frac{n_{\mathbf{x}}}{n_t}$$

[1] Agresti, Categorical Data Analysis, Wiley, 2002.
[2] Koller and Friedman, Probabilistic Graphical Models, MIT Press, 2009 (esp. 18.2 and 20.7)
[3] Christensen, Log-linear models and logistic regression, 1996.

Bayesian approaches



Aim: Finding the model \mathcal{M} that, for a dataset \mathcal{D} maximizes $p(\mathcal{M}|\mathcal{D})$

$$p(\mathcal{M}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{M}) \times p(\mathcal{M})$$

Posterior probability \propto Likelihood \times Prior probability.

Hundreds of methods and references: BDeu, BD/BDe, MDL, NML, etc.; see details in [1,2].

[1] Koller and Friedman, Probabilistic Graphical Models, MIT Press, 2009 (esp. chapters 18 and 20)

[2] W. Buntine, A guide to the literature on learning probabilistic networks from Data, TKDE 1996.

Frequentist approaches



Avoid the definition of priors

Also hundreds of methods and approaches using statistical tests (eg Chi-squared, likelihood-ratio tests).

→ See details in [1,2,3]

$$P(\mathbf{x}) = \lim_{n_t \rightarrow \infty} \frac{n_{\mathbf{x}}}{n_t}$$

[1] Agresti, Categorical Data Analysis, Wiley, 2002.

[2] Koller and Friedman, Probabilistic Graphical Models, MIT Press, 2009 (esp. 18.2 and 20.7)

[3] Christensen, Log-linear models and logistic regression, 1996.

Search



Traditional algorithms:

- local search (eg greedy, backward)
- simulated annealing
- genetic algorithms
- MCMC/Gibbs
- etc.

Note:

- BN: scores also require an order on the variables

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probabilities: $P(x)$
 probability: $P(x_i)$
 Quantifying uncertainty: $P(x_i)$
 Not a trade-off in classical agent systems
 Aka: Compactly representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 What we will cover: **IN**
 What we will not cover: **OUT**

Graphical models 101

Classes of graphical models
 Undirected: $G = (V, E)$
 Directed: $G = (V, E)$
 Stochastic: $G = (V, E)$

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data
 Scoring: $\log P(x)$
 Search: $\log P(x)$

Graph theory

Maximal cliques and minimal separators
 Definition 1: A clique is a subset of nodes in a graph such that every two nodes in the subset are adjacent to each other.
 Definition 2: A minimal separator is a set of nodes that separates the graph into two or more components.

What are decomposable models
 Decomposable models are a Markov Random Field for which the graph is chordal or triangulated.

Properties of decomposable models
 1. Closed form for $P(x)$
 2. No. of log-likelihoods
 3. Junction tree algorithm
 4. No. of cliques $O(n^2)$
 5. Linear-time separable property [3, 4]
 6. Interaction between ICI and MRF [2]

Useful algorithms
 Junction tree algorithm
 Variable elimination

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:
 1. scalable scoring
 2. efficient search
 3. scalable belief propagation
 = all the results we will show here

Bottom line
 A score decomposable model is essential to:
 - AIC for MRFs
 - A set of operations to support MRFs
 - Any scoring function that has been identified for MRFs in the literature to be used for decomposable models
 - MRF-based applications
 - MRF-based applications that use the MRF as a model of a system
 - The scope of MRF

Most scores are scalable
 Entropy [1]
 Submodular Ladder [2] Because it is submodular when entropy is used
 Global statistics [3]
 Max. FMS [4, 5]

Break

Efficient search

Scoring in greedy search
 In this case, we only need:
 - Data
 - Scoring of edge (0,1) to node
 - 12.2

Clique graph (CG)
 Definition 1: A clique graph is a graph in which the nodes are cliques of a graph G and the edges are the maximal cliques of G .
 Definition 2: A maximal clique is a clique that cannot be extended by adding one more node adjacent to all of its nodes.
 Definition 3: A maximal clique is a clique that cannot be extended by adding one more node adjacent to all of its nodes.

Clique graph and greedy search
 We can extend the greedy search algorithm to the clique graph.

Search and statistical paradigm
 Search: $\log P(x)$
 Statistical: $\log P(x)$

The nitty-gritty

Counting efficiently
 Scoring: for example with KL, minimized when...
 What does it mean to compute $P(x)$?
 - Efficient search: ball down to efficient counting

Counting efficiently (2)
 Many different counting methods require making use of the CG.
 - Efficient search: ball down to efficient counting

Memorization
 From the high multiplicity of the CG, we can use the CG to memorize the CG.
 - Efficient search: ball down to efficient counting

Addition of the same edge to different reference models
 What we have seen so far:
 - Efficient search: ball down to efficient counting

How fast can we get?
 - Efficient search: ball down to efficient counting

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell
 1. Graphical models are everywhere!
 2. We can learn graphical models from data with 1,000+ variables
 3. It is possible to do this in a way that is efficient and scalable
 4. There is still so much work to be done

Open problems
 1. Efficient distributed search
 2. Better scores (eg on Directed scoring on MRF)
 3. Efficient scoring of marginal "bits"
 4. Efficient data structures for counting
 5. Learning out of core
 6. Latent variables

Open problems (2)
 7. How to handle numerical variables
 8. How to handle missing values?
 9. Learning accurate parameters in large tables
 10. Many problems are non-hanging (MRF) you just need to push them!

We hope that you enjoyed our tutorial on...
 Scalable learning of graphical models
 From the lecture and your study
 Many thanks to the organizers of the course

Maximal cliques and minimal separators

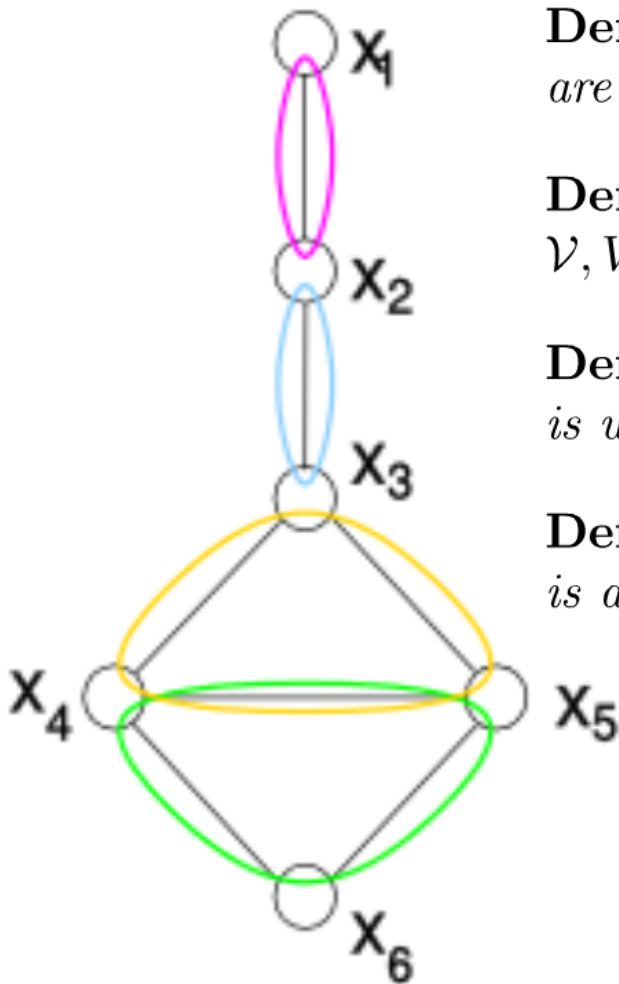
Let $\mathcal{G} = (\mathcal{V}, E)$ be the undirected graph, where \mathcal{V} is the set of variables and E the set of edges in \mathcal{G} .

Definition 1 A set $C \subseteq \mathcal{V}$ is a clique of \mathcal{G} iff all its vertices are pairwise adjacent.

Definition 2 A clique C is maximal iff there is no vertex $V \in \mathcal{V}, V \notin C$ such that $C \cup \{V\}$ is a clique.

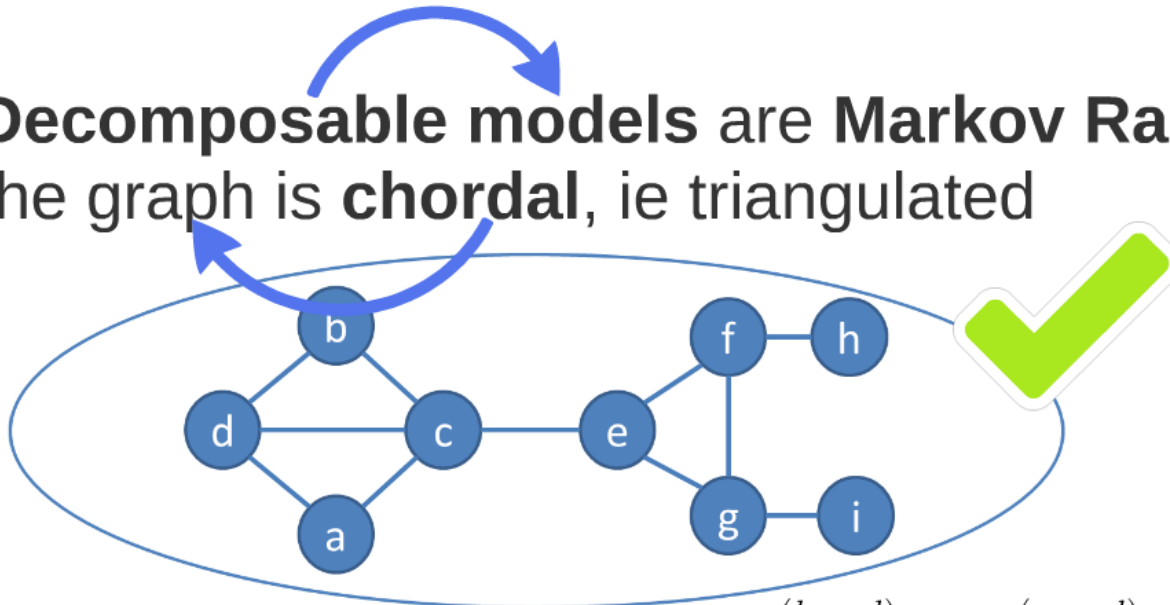
Definition 3 A set $S \subseteq \mathcal{V}$ is a separator of \mathcal{G} if $G = (\mathcal{V} - S, E)$ is unconnected.

Definition 4 A separator S of \mathcal{G} is minimal if no subset of S is a separator.

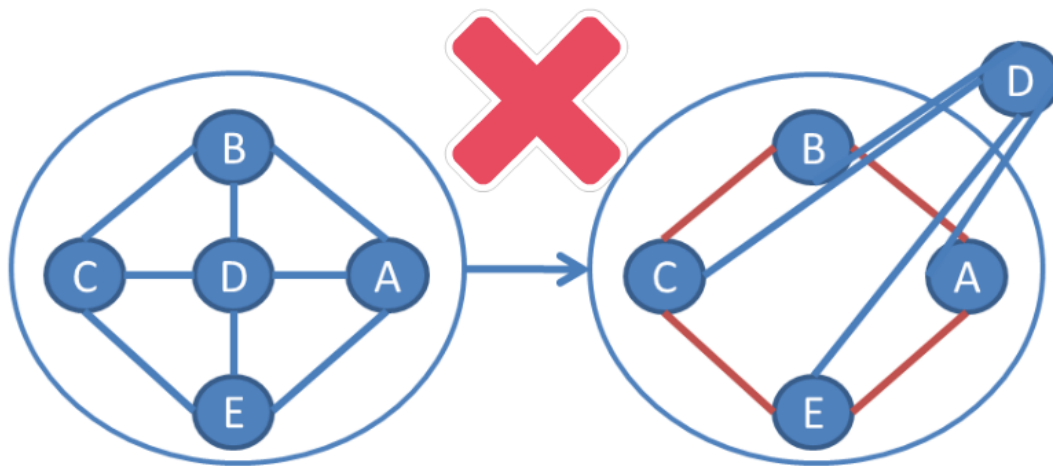


What are decomposable models

Decomposable models are Markov Random Fields for which the graph is **chordal**, ie triangulated



$$E_{a,\dots,i} = N \cdot \frac{p_{BCD}(b, c, d) \cdot p_{ACD}(a, c, d) \cdot p_{CE}(c, e) \cdot p_{EFG}(e, f, g) \cdot p_{FH}(f, h) \cdot p_{GI}(g, i)}{p_{CD}(c, d) \cdot p_C(c) \cdot p_E(e) \cdot p_F(f) \cdot p_G(g)}$$

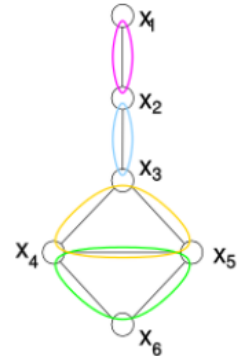


Properties of decomposable models

1. Closed form MLE $\iff p_{\mu}(\mathbf{x}) = \frac{\prod_{C \in \mathcal{C}} p_C(\mathbf{x})}{\prod_{S \in \mathcal{S}} p_S(\mathbf{x})}$

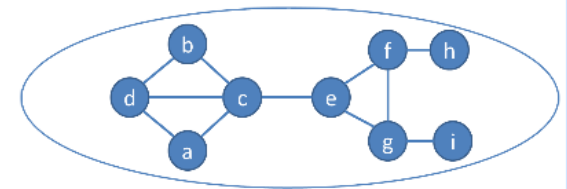
2. Not a big restriction:

- Every distribution that can be modeled by a graphical model can be exactly modeled by some decomposable model [1]



3. Junction-tree equivalence

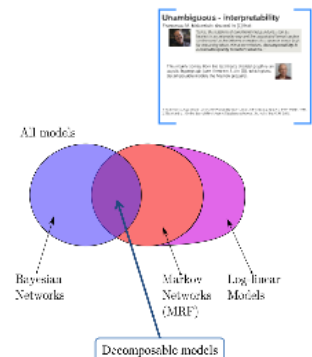
- Spanning tree over clique-graph
- Exact and efficient belief propagation



4. MLE always exist [2]

5. Unambiguous - desirable property [1,4]

6. Intersection between BN and MRF [3]



[1] Christensen, Log-linear models and logistic regression, 1997.

[2] Agresti, Categorical data analysis, 2002.

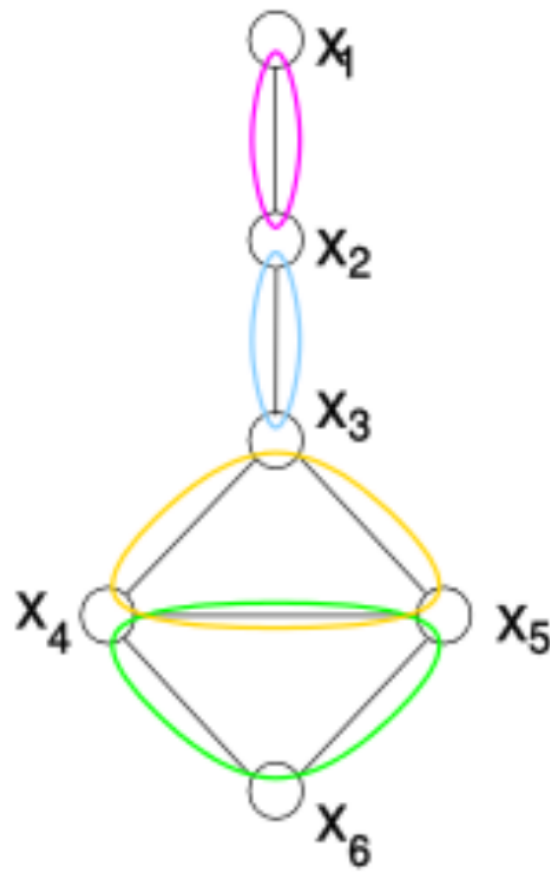
[3] Koller and Friedman, Probabilistic Graphical Models, 2009.

[4] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.

Probabilistic models

$$p_{\mu}(\mathbf{x}) = \frac{\prod_{C \in \mathcal{C}} p_C(\mathbf{x})}{\prod_{S \in \mathcal{S}} p_S(\mathbf{x})}$$

be modeled by a
actly modeled by
[1]
e



Properties of decomposable models

1. Closed form MLE $\iff p_{\mu}(\mathbf{x}) = \frac{\prod_{C \in \mathcal{C}} p_C(\mathbf{x})}{\prod_{S \in \mathcal{S}} p_S(\mathbf{x})}$

2. Not a big restriction:

- Every distribution that can be modeled by a graphical model can be exactly modeled by some decomposable model [1]

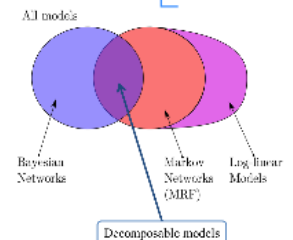
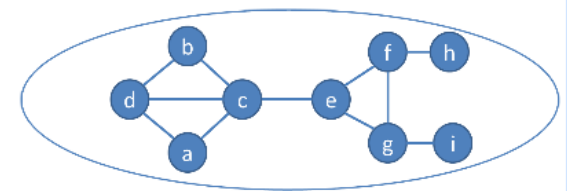
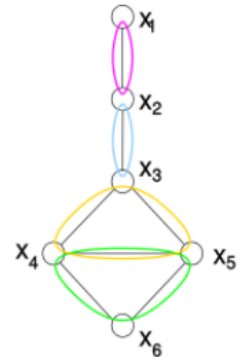
3. Junction-tree equivalence

- Spanning tree over clique-graph
- Exact and efficient belief propagation

4. MLE always exist [2]

5. Unambiguous - desirable property [1,4]

6. Intersection between BN and MRF [3]



[1] Christensen, Log-linear models and logistic regression, 1997.

[2] Agresti, Categorical data analysis, 2002.

[3] Koller and Friedman, Probabilistic Graphical Models, 2009.

[4] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.

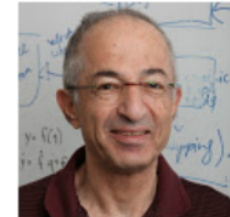
Unambiguous - interpretability

Francesco M. Malvestuto showed in [1] that:



*"Since the relations of conditional independence can be treated in an axiomatic way and the associated formal system can be used as the inference engine of a common sense logic for reasoning about relevance relations, **decomposability is a desirable quality fo belief networks.**"*

This mainly comes from the fact that a chordal graph is an acyclic hypergraph (see Theorem 3.4 in [2]), which gives decomposable models the Markov property.



[1] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.

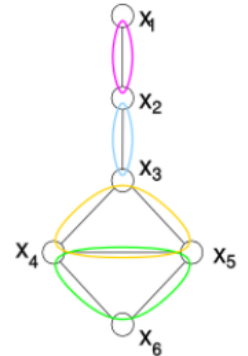
[2] Beeri and al., On the desirability of Acyclic Database Schemes, Journal of the ACM, 1983.

Properties of decomposable models

1. Closed form MLE $\iff p_{\mu}(\mathbf{x}) = \frac{\prod_{C \in \mathcal{C}} p_C(\mathbf{x})}{\prod_{S \in \mathcal{S}} p_S(\mathbf{x})}$

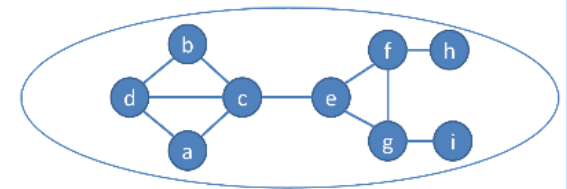
2. Not a big restriction:

- Every distribution that can be modeled by a graphical model can be exactly modeled by some decomposable model [1]



3. Junction-tree equivalence

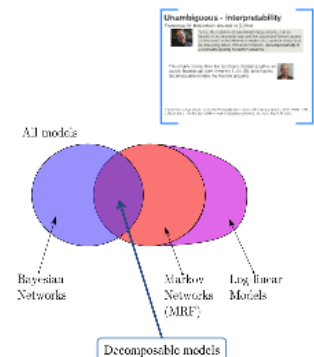
- Spanning tree over clique-graph
- Exact and efficient belief propagation



4. MLE always exist [2]

5. Unambiguous - desirable property [1,4]

6. Intersection between BN and MRF [3]



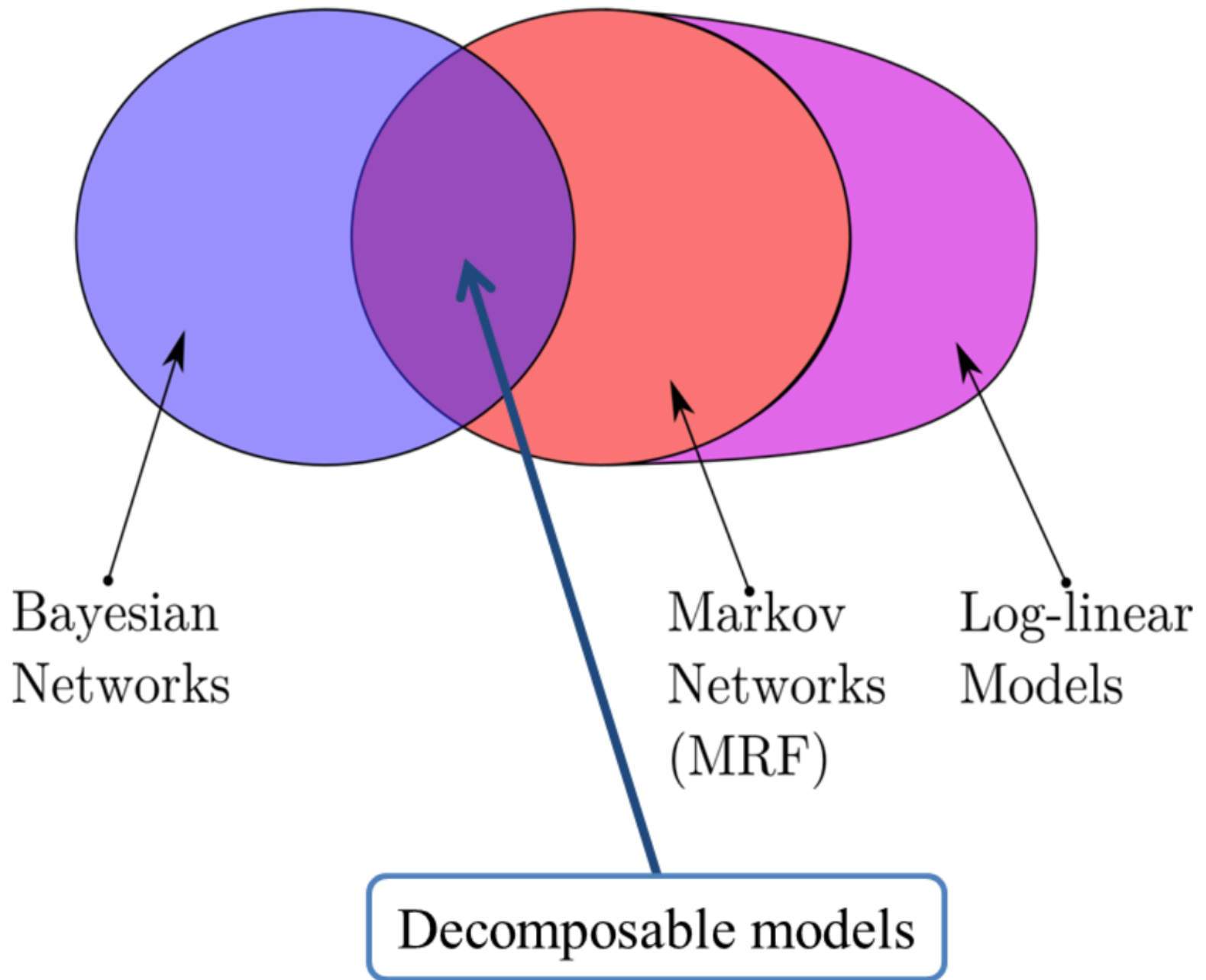
[1] Christensen, Log-linear models and logistic regression, 1997.

[2] Agresti, Categorical data analysis, 2002.

[3] Koller and Friedman, Probabilistic Graphical Models, 2009.

[4] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.

All models



Bayesian
Networks

Markov
Networks
(MRF)

Log-linear
Models

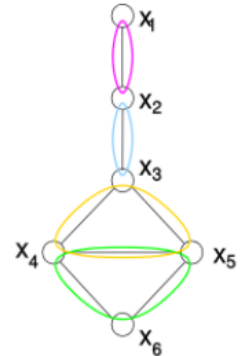
Decomposable models

Properties of decomposable models

1. Closed form MLE $\iff p_{\mu}(\mathbf{x}) = \frac{\prod_{C \in \mathcal{C}} p_C(\mathbf{x})}{\prod_{S \in \mathcal{S}} p_S(\mathbf{x})}$

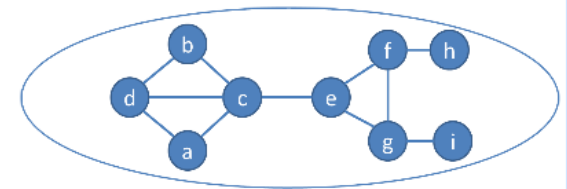
2. Not a big restriction:

- Every distribution that can be modeled by a graphical model can be exactly modeled by some decomposable model [1]



3. Junction-tree equivalence

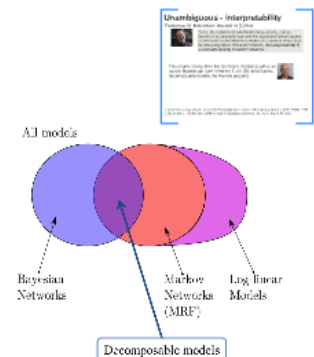
- Spanning tree over clique-graph
- Exact and efficient belief propagation



4. MLE always exist [2]

5. Unambiguous - desirable property [1,4]

6. Intersection between BN and MRF [3]



[1] Christensen, Log-linear models and logistic regression, 1997.

[2] Agresti, Categorical data analysis, 2002.

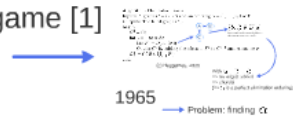
[3] Koller and Friedman, Probabilistic Graphical Models, 2009.

[4] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.

Useful algorithms

Verifying decomposability

Elimination game [1]



Lex-BFS [2] and MCS [3]

- can find a peo for a chordal graph in linear time

Verification:

1. find an vertex ordering α
2. chordal $\leftarrow (EliminationGame(G, \alpha) == G)$

→ Recognition in linear time $O(n+m)$

- [1] D.R. Fulkerson et al. Incidence matrices and interval graphs, Pacific J. Math. 1965.
 [2] D. Rose et al., Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput., 1976.
 [3] R.E. Tarjan et al., Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM J. Comput., 1984.
 [4] P. Heggernes, Minimal triangulations of graphs: A survey, Discrete Mathematics, 2006.

```

outp-
reg RS, RW, E, ENA...
reg [3:0] KEYO, CYCLE;
reg [4:0] DATA;
reg [4:0] KEY;
reg [7:0] DB;
reg [6:0] PULSE;

task ASK_01;
case (CYCLE)
4'h0:
begin
{RS, RW, E, ENABLE} = 4'b10;
DB [7:0] = 8'h35;
end
4'h1: {RS, RW, E, ENABLE} = 4'b...
4'h3: CYCLE = CYCLE - 4'h1;
endcase
endtask
    
```

wiseGEEK

Triangulation



Triangulation is easy

- eg Elimination game actually triangulates

Minimum triangulation = as few edges added as possible => NP-hard [1]

Minimal triangulation = only one chord per square [2,3] => $O(n^{2.376})$



Heuristics and simplifications for restricted classes

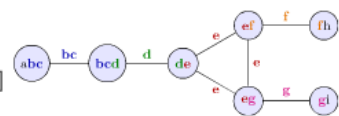
→ bounded degree, perfect, trapezoid, AT-free, planar, ...

- [1] M. Yannakakis, Computing the minimum fill-in is NP-complete, SIAM J. Algebraic Discrete Methods, 1981.
 [2] D. Rose et al., Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput., 1976.
 [3] P. Heggernes, Minimal triangulations of graphs: A survey, Discrete Mathematics, 2006.
 [4] P. Heggernes et al. Computing minimal triangulations in time $O(n \log n) = O(n^{2.376})$, SIAM J. Disc. Math.

Deriving junction-tree

Steps:

1. compute clique graph [1]
2. compute a maximum spanning tree on the clique graph - Kruskal's algorithm with negative weights [2]



→ Linear-time algorithms exist based on Maximum Cardinality Search [1,3]

- [1] P. Galinier et al., Chordal graphs and their clique graphs cliques of a chordal graph, Information Processing Letters, 2011.
 [2] J.B. Kruskal, On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proc. Amer. Math. Soc. 1956.
 [3] R.E. Tarjan et al., Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM J. Comput., 1984.

Verifying decomposability

Elimination game [1]



Algorithm Elimination Game
Input: A graph $G = (V, E)$ and an ordering $\alpha = (v_1, \dots, v_n)$ of V .
Output: The filled graph G_n^+ .

```
begin
   $G^0 = G$ ;
  for  $i = 1$  to  $n$  do
    Let  $F^i = D_{G^{i-1}}(v_i)$ ;
    Obtain  $G^i$  by adding the edges in  $F^i$  to  $G^{i-1}$  and removing  $v_i$ ;
     $G_n^+ = (V, E \cup \bigcup_{i=1}^n F^i)$ ;
  end
```

(c) Heggernes, 2006

With $\alpha = (b, a, c)$
 \Rightarrow no edges added
 \Rightarrow chordal
($\Rightarrow \alpha$ is a perfect elimination ordering)

1965

→ Problem: finding α

Lex-BFS [2] and MCS [3]

- can find a peo for a chordal graph in linear time

Verification:

1. find an vertex ordering α
2. chordal $\leftarrow (EliminationGame(G, \alpha) == G)$

→ Recognition in linear time **$O(n+m)$**

[1] D.R. Fulkerson et al. Incidence matrices and interval graphs, Pacific J. Math. 1965.
[2] D. Rose et al., Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput., 1976.
[3] R.E. Tarjan et al., Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM J. Comput., 1984.
[4] P. Heggernes, Minimal triangulations of graphs: A survey, Discrete Mathematics, 2006.

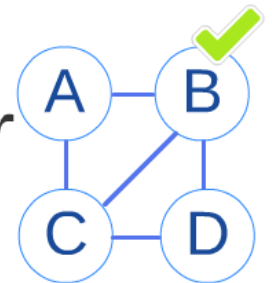
Triangulation

Triangulation is easy

- eg Elimination game actually triangulates

Minimum triangulation = as few edges added as possible => NP-hard [1]

Minimal triangulation = only one chord per square [2,3] => $O(n^{2.376})$



Heuristics and simplifications for restricted classes

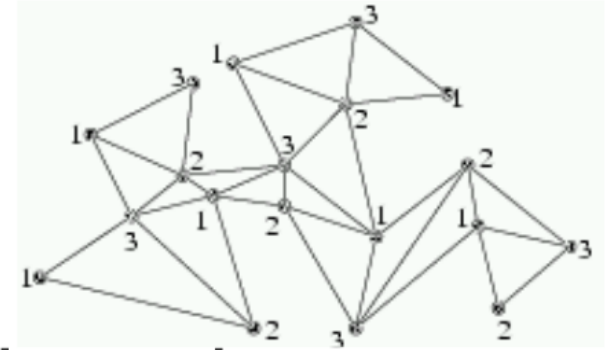
→ bounded degree, perfect, trapezoid, AT-free, planar, ...

[1] M. Yannakakis, Computing the minimum fill-in is NP-complete, SIAM J. Algebraic Discrete Methods, 1981.

[2] D. Rose et al., Algorithmic aspects of vertex elimination on graphs, SIAM J. Comput., 1976.

[3] P. Heggernes, Minimal triangulations of graphs: A survey, Discrete Mathematics, 2006.

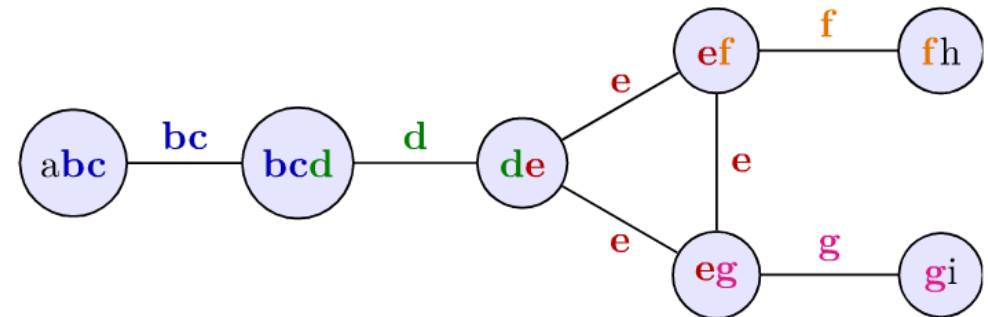
[4] P. Heggernes et al. Computing minimal triangulations in time $O(n \log n) = o(n^{2.376})$, SIAM J. Disc. Math.



Deriving junction-tree

Steps:

1. compute clique graph [1]



2. compute a maximum spanning tree on the clique graph - Kruskal's algorithm with negative weights [2]

→ Linear-time algorithms exist based on Maximum Cardinality Search [1,3]

[1] P. Galinier et al., Chordal graphs and their clique graphs
cliques of a chordal graph, Information Processing Letters, 2011.

[2] J.B. Kruskal, On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proc. Amer. Math. Soc. 1956.

[3] R.E. Tarjan et al., Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs, SIAM J. Comput., 1984.

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probabilities: $P(x)$
 Probability: $P(x)$
 Quantifying uncertainty
 Not a trade-off in classical agent systems
 Aids: Complexity representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 In: $P(x)$
 Out: $P(x)$

Graphical models 101

Classes of graphical models

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data

Graph theory

Maximal cliques and minimal separators

What are decomposable models
 Decomposable models are a Markov Random Field for which the graph is chordal or triangulated

Properties of decomposable models

1. Closed form for $P(x)$
2. Very easy to optimize
3. Junction tree algorithms
4. No ∞ loops (local)
5. Linear-time inference
6. Interaction between ICI and MRF [2]

Useful algorithms

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:

1. scalable scoring
2. efficient search
3. scalable belief propagation

= all the results we will show here

Bottom line
 A score decomposable models is equivalent to:
 - AIC for MRFs
 - A set of operations (Bayesian networks)

Most scores are scalable
 Entropy [1] ✓
 Bayesian Ladder [2] ✓
 G-robust [3] ✓
 Max. FMS [4] ✓

Break

Efficient search

Scoring in greedy search

Clique graph (CG)

Clique graph and greedy search

Search and statistical paradigm

The nitty-gritty

Counting efficiently
 Scoring for example with KL minimized when...
 What does it mean to compute $P(x)$?
 Efficient counting built upon efficient counting

Counting efficiently (2)
 Many algorithms count by summing over all possible configurations of the variables

Memorization
 From high multiplicity to low multiplicity

Addition of the same edge to different reference models
 What we have seen so far
 Curvature
 How often does that happen?
 How can we use this information?

How fast can we get?

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell

1. Graphical models are everywhere
2. We can learn graphical models from data with 1,000+ variables
3. It is crucial to be able to use the library that we are providing on your tablet device
4. There is still so much work to be done

Open problems

1. Efficient constrained search
2. Better scores (eg on Directed scoring on MRF)
3. Efficient scoring of marginal "bits"
4. Efficient data structures for counting
5. Learning out of core
6. Latent variables

Open problems (2)

1. How to handle numerical variables
2. How to handle missing values?
3. Learning accurate parameters in large tables

Scalable learning of graphical models
 From the lecture and your study

Decomposable models are essential for scalability, because...

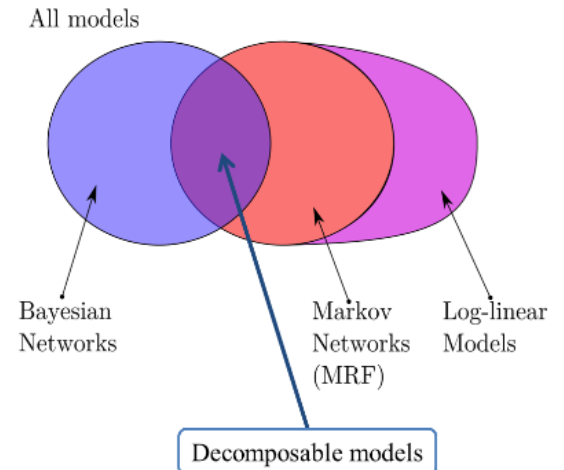
... we need:

1. scalable scoring

2. efficient search

3. scalable belief propagation

+ all the results we will show here



Efficient scoring

In the general case, most scoring functions are in $O(d^n)$

Example: likelihood ratio test

$$\frac{\partial \log \mathcal{L}(\theta)}{\partial \theta_i} = \sum_{j=1}^n \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta_j} \frac{\partial \theta_j}{\partial \theta_i}$$

Need to focus on Bayesian Networks:

1. which have closed-form MLEs
2. for which most scores are decomposable

Efficient search

Searching the space of BNs is not efficient because:

1. we often need to first define a total order ζ over the variables
2. many BN structures are indistinguishable from data

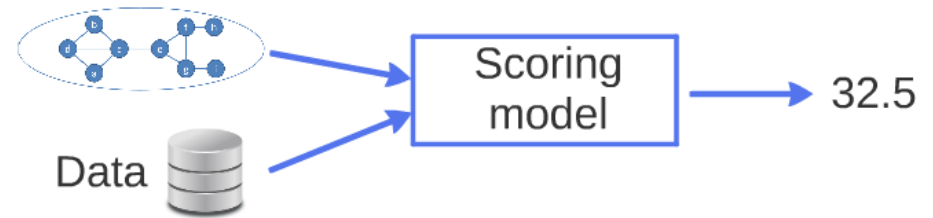
Scalable belief propagation

To be scalable and exact, we have to use a decomposable model

\Rightarrow so we might as well directly learn from this class

Note: Transforming a BN into a decomposable model is not easy

Efficient scoring



In the general case, most scoring functions are in $O(d^n)$

Example - likelihood ratio test

$$G^2(\mathcal{M}) = 2 \cdot \sum_{x_1 \in \text{Dom}(X_1)} \cdots \sum_{x_n \in \text{Dom}(X_n)} O_{x_1, \dots, x_n} \cdot \ln \left(\frac{O_{x_1, \dots, x_n}}{E_{x_1, \dots, x_n}} \right)$$

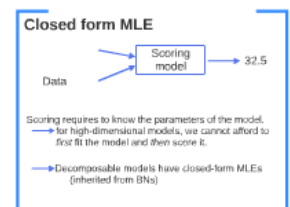
Exponential with the number of variables

Need to fit model *first*

KL divergence, negative log-likelihood, most MDL scores, etc.

Need to focus on Bayesian Networks:

1. which have closed-form MLEs
2. for which most scores are decomposable



1,000 binary variables

10 ... 000000 operations

300

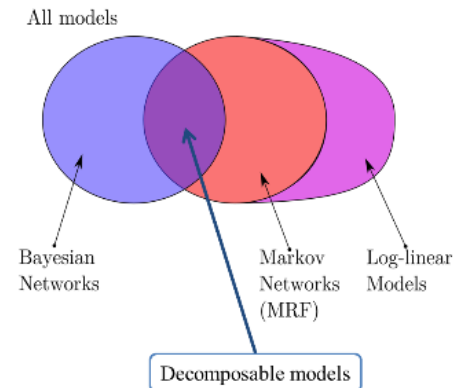
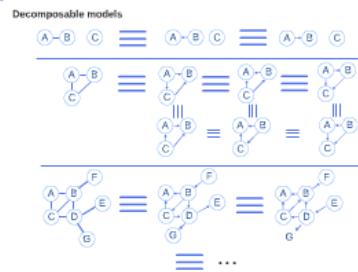


10^{82} atoms in the
observable
universe...

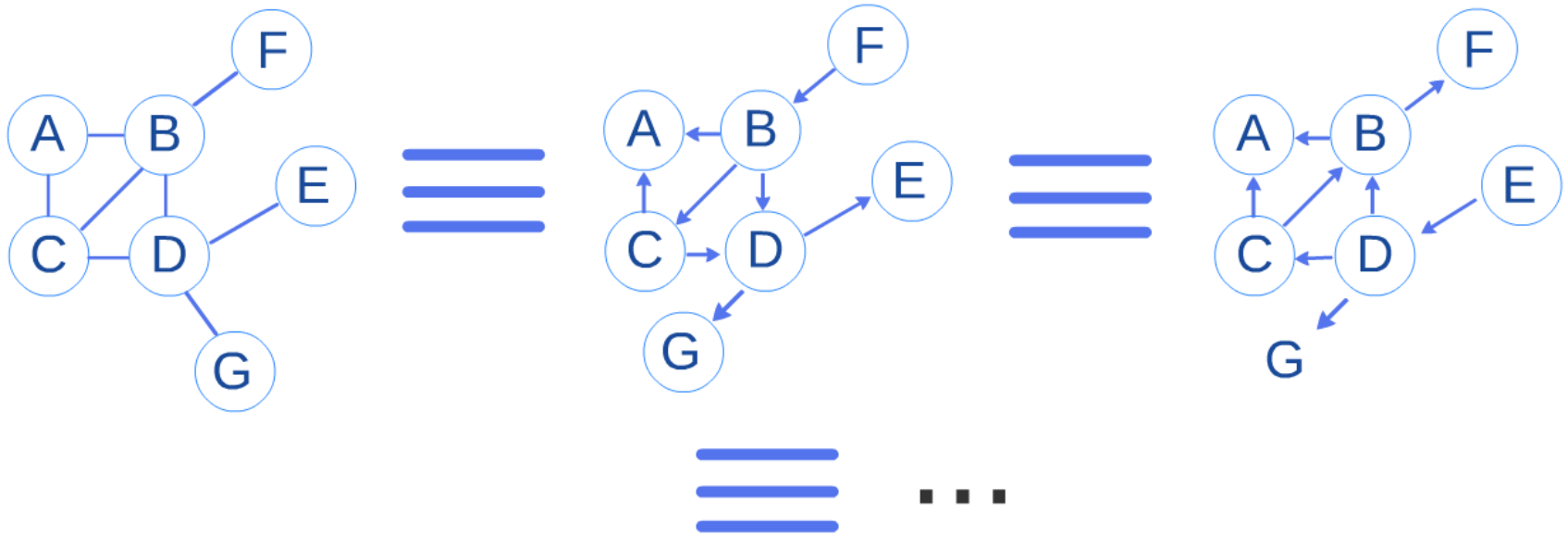
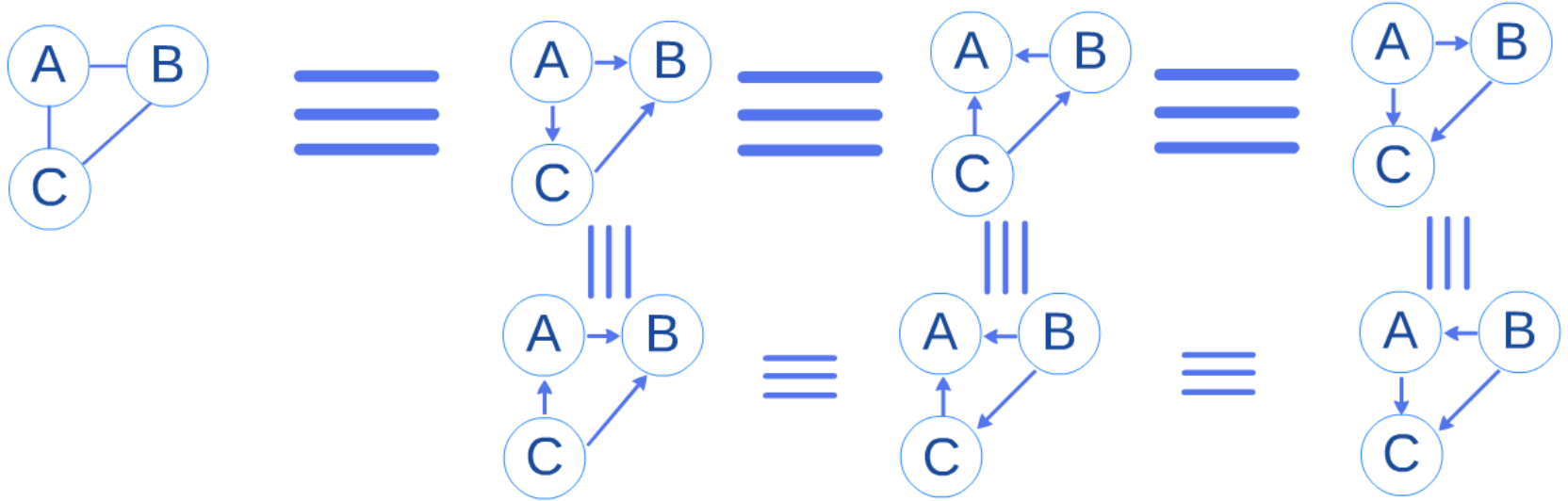
Efficient search

Searching the space of BNs is not efficient because:

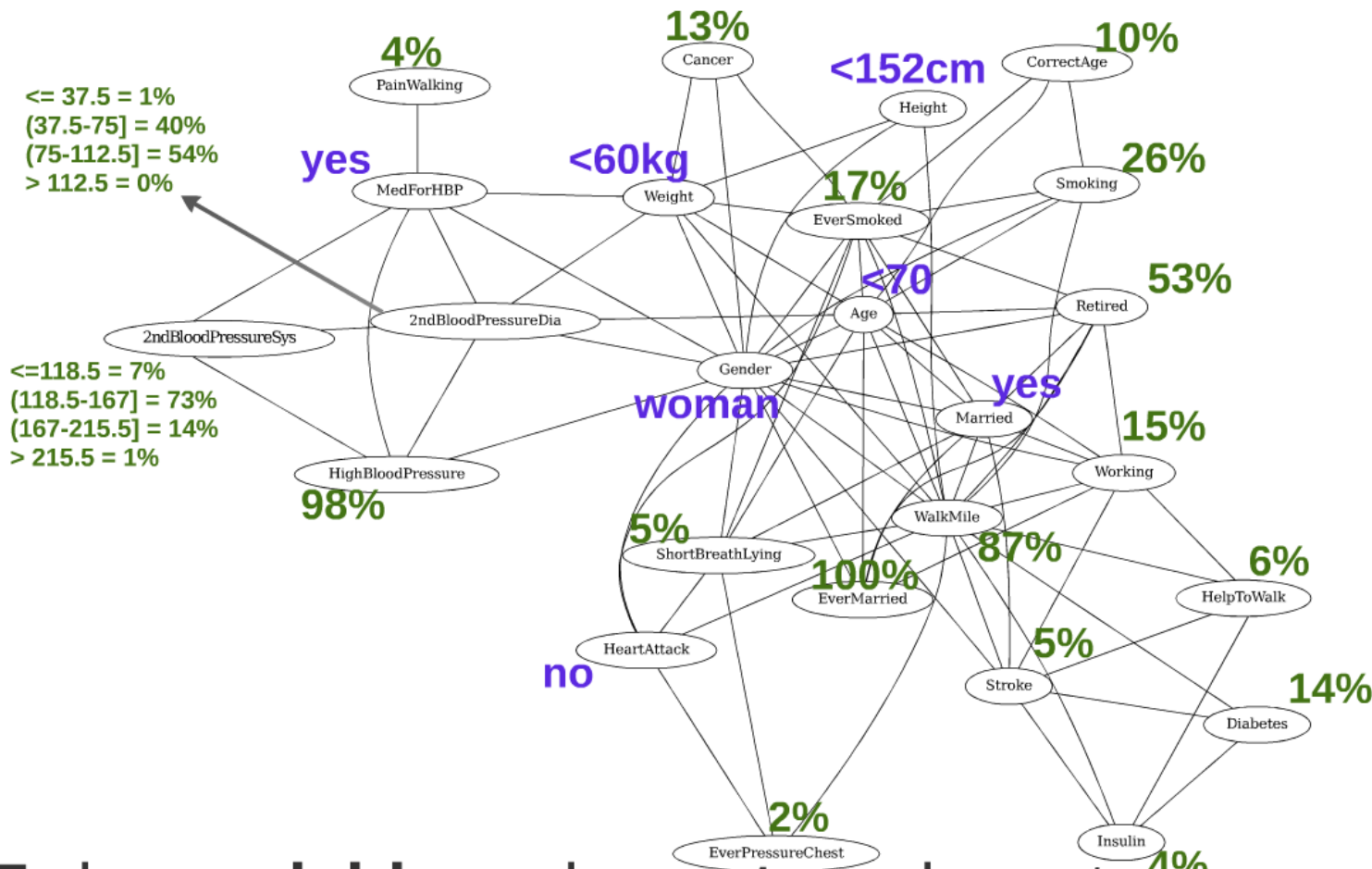
1. we often need to first define a total order ζ over the variables
2. many BN structures are indiscernible from data



Decomposable models



Scalable belief propagation



To be **scalable** and **exact**, we have to use a decomposable model

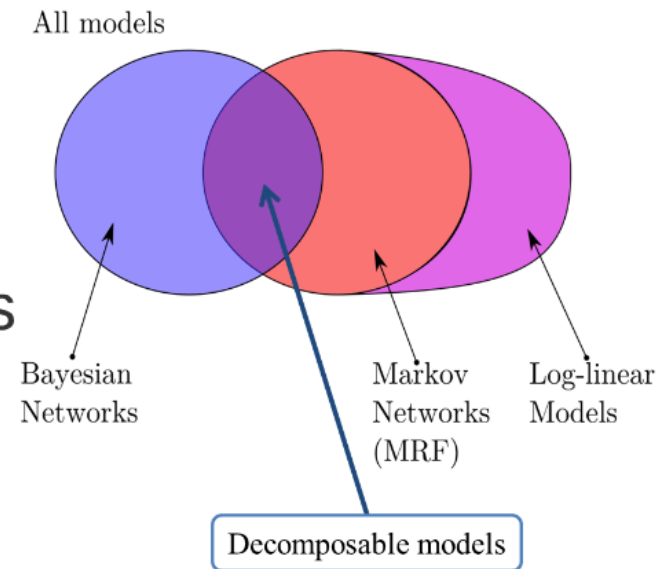
=> so we might as well directly learn from this class

Note: transforming a BN into a decomposable model is not easy.

Bottom line

A decomposable model is equivalent to:

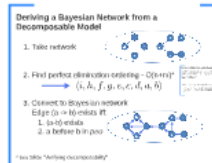
- a Markov Network
- a set of equivalent Bayesian Networks



→ Any scoring function that has been developed for **MRFs*** or for **BNs can be used** for decomposable models

→ MRF: direct applicability

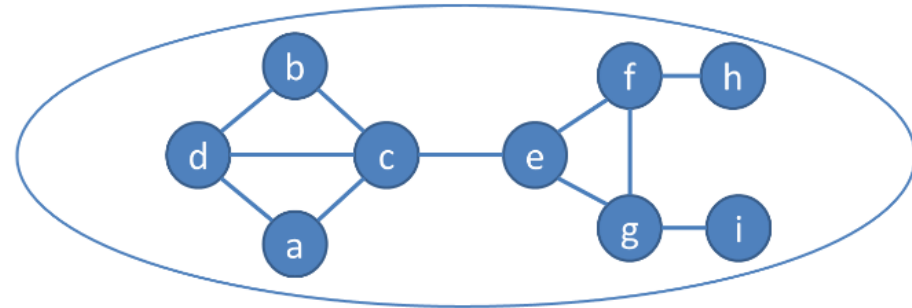
→ BN: derive an equivalent BN first and then use the score on it



* this implies metrics developed for log-linear models as well

Deriving a Bayesian Network from a Decomposable Model

1. Take network



2. Find perfect elimination ordering - $O(n+m)^*$

→ $\langle i, h, f, g, e, c, d, a, b \rangle$

Perfect elimination ordering (peo)

An ordering on the vertices $\langle v_1, v_2, \dots, v_n \rangle$ is perfect iff each v_i is **simplicial** in $G(\{v_1, \dots, v_n\})$ (the graph where the vertices before v_i have been removed).

A vertex is **simplicial** if its neighbors form a clique.

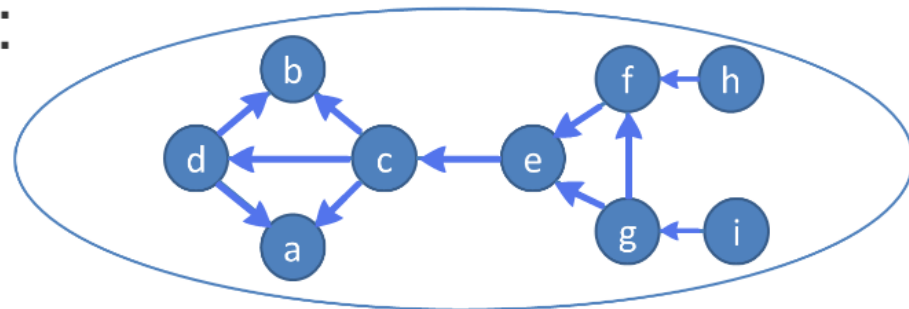
One peo: $\langle i, h, f, g, e, c, d, a, b \rangle$ because:

- $N_i(i) = \{g\}$, $N_i(h) = \{f\}$, which are cliques
- $N_i(f) = \{e, g\}$ (not h), which is a clique
- etc

3. Convert to Bayesian network

Edge (a → b) exists iff:

1. (a-b) exists
2. a before b in *peo*

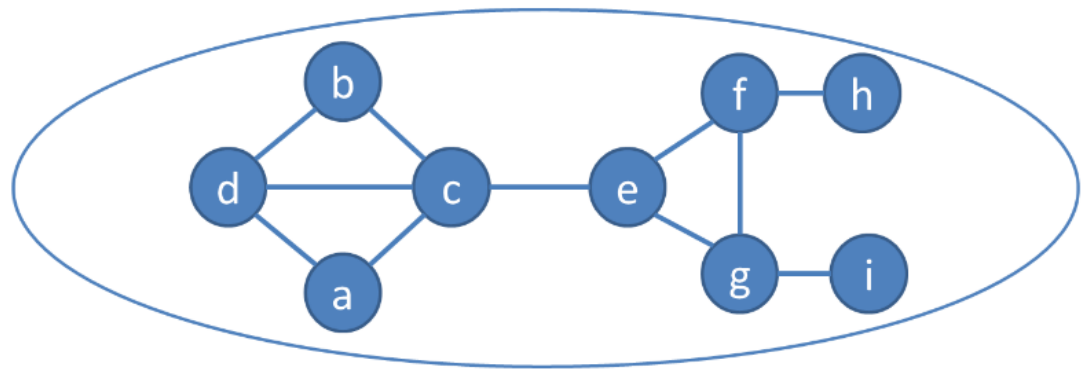


* see Slide "Verifying decomposability"

Perfect elimination ordering (peo)

An ordering on the vertices $\langle v_1, v_2, \dots, v_m \rangle$ is perfect iff each v_i is **simplicial** in $G(\{v_i, \dots, v_m\})$ (the graph where the vertices before v_i have been removed).

A vertex is **simplicial** if its neighbors form a clique.



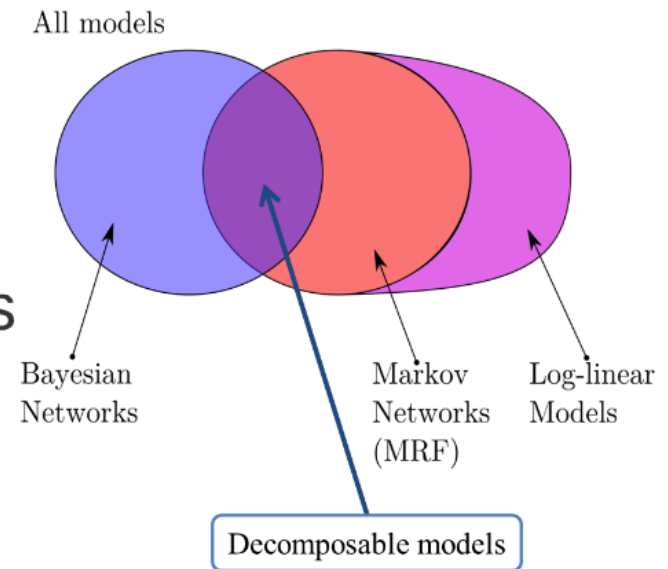
One peo: $\langle i, h, f, g, e, c, d, a, b \rangle$ because:

- $N_1(i) = \{g\}, N_2(h) = \{f\}$, which are cliques
- $N_3(f) = \{e, g\}$ (not h), which is a clique
- etc

Bottom line

A decomposable model is equivalent to:

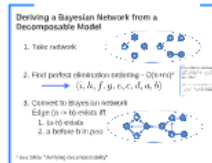
- a Markov Network
- a set of equivalent Bayesian Networks



→ Any scoring function that has been developed for **MRFs*** or for **BNs can be used** for decomposable models

→ MRF: direct applicability

→ BN: derive an equivalent BN first and then use the score on it



* this implies metrics developed for log-linear models as well

Most scores are scalable

Entropy [1] ✓

$$\begin{aligned} D(\theta) &= - \sum_{i=1}^n \sum_{j=1}^k \theta_{ij} (\theta_{ij}^{-1} - \log \theta_{ij}^{-1}) \\ &= \sum_{i=1}^n H(\theta_i) - \sum_{i=1}^n H(\theta_i) \\ &\rightarrow O(2^n) \approx O(2^k) \\ &\text{where } k \text{ is the size of the biggest clique} \end{aligned}$$

Kullback Leibler [1,2] (because is minimized when entropy is also) ✓

G-test statistic [3] ✓

MML / MDL [4,5] ✓

- [1] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.
- [2] Deshpande et al, Efficient Stepwise Selection in Decomposable Models, UAI 2001.
- [3] **Petitjean**, Nicholson and **Webb**, Scaling log-linear analysis to high-dimensional data, IEEE ICDM 2013.
- [4] Altmueller and Haralick, Approximating High Dimensional Probability Distributions, ICPR 2004.
- [5] **Petitjean**, Allison and **Webb**, A statistically efficient and scalable method for log-linear analysis of high-dimensional data, IEEE ICDM 2014.

$$\begin{aligned} H(\mathcal{M}) &= - \sum_{x_1 \in \text{Dom}(X_1)} \cdots \sum_{x_n \in \text{Dom}(X_n)} p_\mu(x_1, \dots, x_n) \cdot \ln p_\mu(x_1, \dots, x_n) \\ &= \sum_{C \in \mathcal{C}} H(X_C) - \sum_{S \in \mathcal{S}} H(X_S) \end{aligned}$$

→ $O(2^n) \Rightarrow O(2^k)$
where k is the size of the biggest clique

Most scores are scalable

Entropy [1] ✓

$$\begin{aligned} D(\theta) &= - \sum_{i=1}^n \sum_{j=1}^k \theta_{ij} (\theta_{ij}^{-1} - \log \theta_{ij}^{-1}) \\ &= \sum_{i=1}^n H(\theta_i) - \sum_{i=1}^n H(\theta_i) \\ &\rightarrow O(2^n) \approx O(2^k) \\ &\text{where } k \text{ is the size of the biggest clique} \end{aligned}$$

Kullback Leibler [1,2] (because is minimized when entropy is also) ✓

G-test statistic [3] ✓

MML / MDL [4,5] ✓

- [1] Malvestuto, Approximating Discrete Probability Distributions with Decomp. Models, IEEE TSMC, 1991.
- [2] Deshpande et al, Efficient Stepwise Selection in Decomposable Models, UAI 2001.
- [3] **Petitjean**, Nicholson and **Webb**, Scaling log-linear analysis to high-dimensional data, IEEE ICDM 2013.
- [4] Altmueller and Haralick, Approximating High Dimensional Probability Distributions, ICPR 2004.
- [5] **Petitjean**, Allison and **Webb**, A statistically efficient and scalable method for log-linear analysis of high-dimensional data, IEEE ICDM 2014.

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
Probability theory + Graph Theory
probability
Quantifying uncertainty
Not a trade-off in classical agent systems
Able: Compactly representing probability distributions

What are graphical models useful for?
- the thousands of applications of these methods...

What we will and will not cover
What we will cover: In, Out
What we will not cover: Out

Graphical models 101

Classes of graphical models

A simple example of structure learning
Hill-climbing search on MRF using AIC

Learning a model from data
Scoring Search

Graph theory

Maximal cliques and minimal separators

What are decomposable models
Decomposable models are a Markov Random Fields for which the graph is chordal or triangulated

Properties of decomposable models
1. Closed form for Z
2. No. of log-likelihood
3. Junction tree algorithm
4. No. of cliques $O(n^2)$
5. Linear-time separable property [3, 4]
6. Interaction between IJ and MRF [2]

Useful algorithms

Evaluation - Scoring

Decomposable models are essential for scalability, because...
... we need:
1. scalable scoring
2. efficient search
3. scalable belief propagation
= all the results we will show here

Bottom line
A score decomposable models is equivalent to:
- A set of pairwise Bayesian networks
- Any scoring function that has been developed for MRF in the literature can be used for decomposable models
- MRF-based approaches
- MRF-based approaches are faster and more accurate than the state-of-the-art

Most scores are scalable
Entropy [1]
Submodular Ladder [2] Because it is submodular when entropy is used
Gini index [3]
Max. Entropy [4]

Break

Efficient search

Scoring in greedy search
In this case, we only need:
Data
Scoring of edge (i, j) in G
12.2

Clique graph (CG)
Submodular decomposition [1]
- Maximal cliques are the nodes of the clique graph
- Maximal cliques are the nodes of the clique graph
- Maximal cliques are the nodes of the clique graph

Clique graph and greedy search
We can extend the greedy search to the clique graph [1]
The search that can be used for the greedy search will be the same as the search that can be used for the greedy search

Search and statistical paradigm

The nitty-gritty

Counting efficiently
Scoring for example with KL minimized when...
What does it mean to compute $D(X||Y)$?
Efficient counting built on efficient counting

Counting efficiently (2)
Many different counting techniques require making use of the CG
What does it mean to compute $D(X||Y)$?
Efficient counting built on efficient counting

Memorization
From the high multiplicative complexity of the CG to the low multiplicative complexity of the CG
Efficient counting built on efficient counting

Addition of the same edge to different reference models
What we have seen so far
Counting the addition of an edge into the CG
Corollary
How often does that happen?
How can we use this information?

How fast can we get?

Use cases

Study of the elderly
- 25 variables
- 15,000 patients

Insurance customer management
- 93 variables
- 6,000 customers

Portfolio management
- 500 variables
- 20 years of trading

Wrapping up!

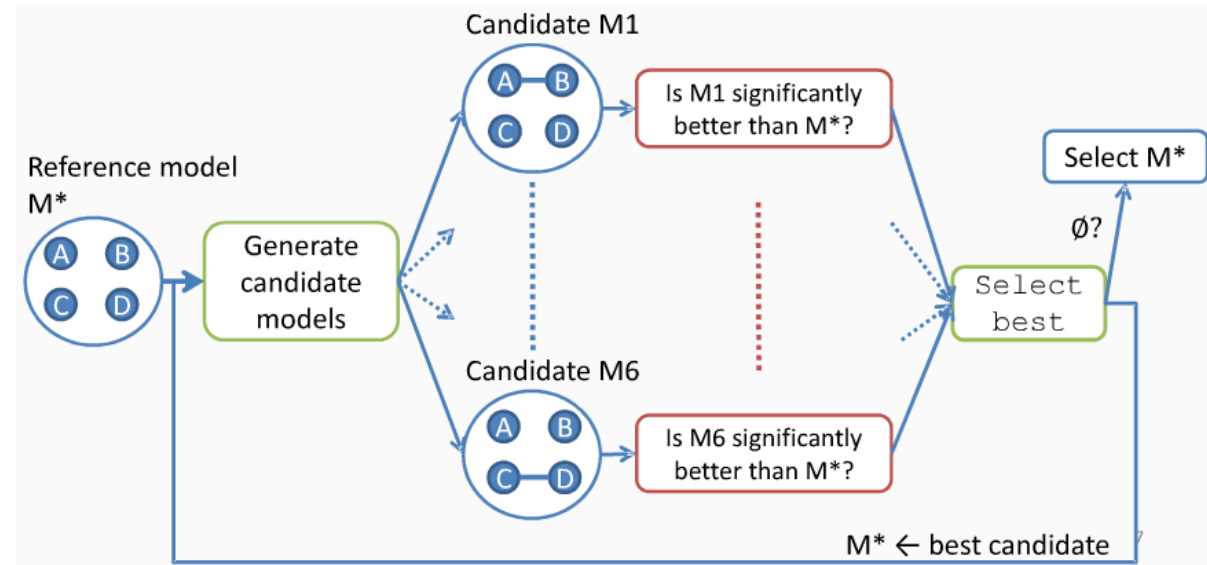
This tutorial in a nutshell
1. Graphical models are not merely useful
2. Graphical models are not merely useful
3. Graphical models are not merely useful

Open problems
1. Efficient constrained search
2. Better scores (eg on Directed scoring on MRF)
3. Efficient scoring of marginal "bits"
4. Efficient data structures for counting
5. Learning out of core
6. Latent variables

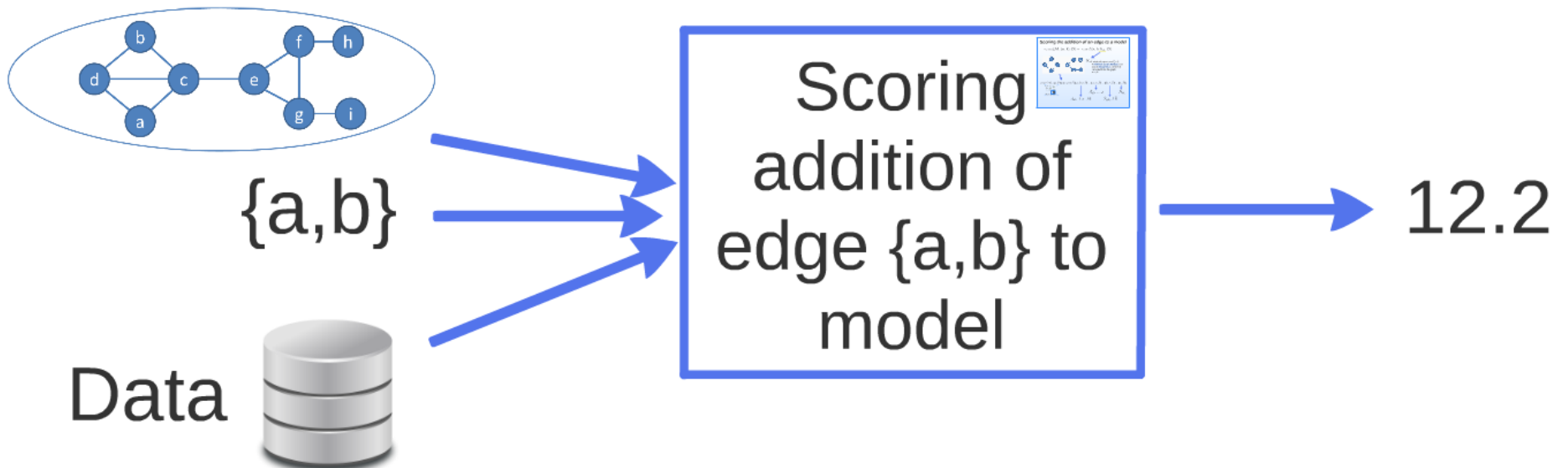
Open problems (2)
7. How to handle numerical variables
8. How to handle missing values?
9. Learning accurate parameters in large tables
10. Learning accurate parameters in large tables

Scalable learning of graphical models
From the lecture and your study
Many problems are fun-hanging
That you just need to push them!

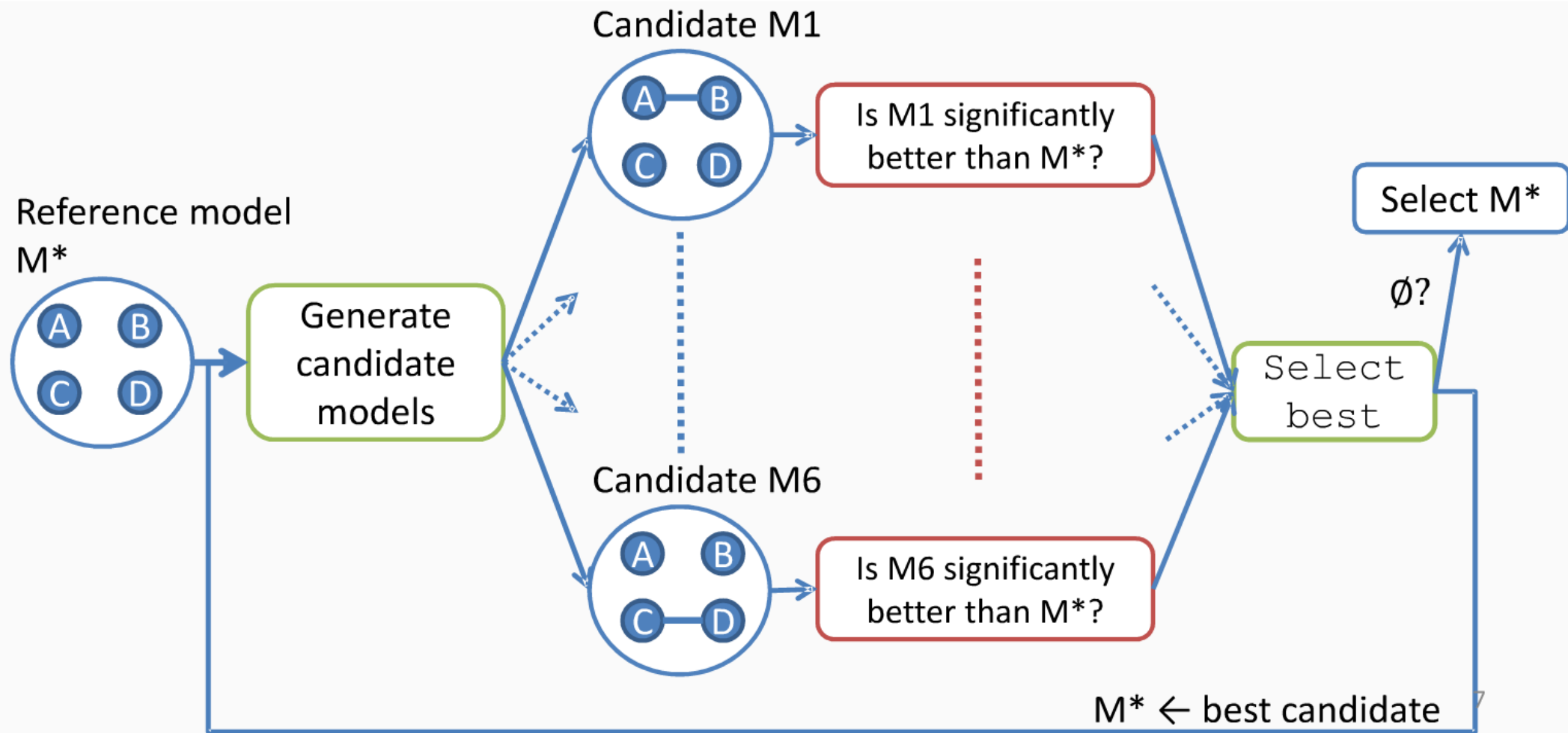
Scoring in greedy search



In this case, we only need...

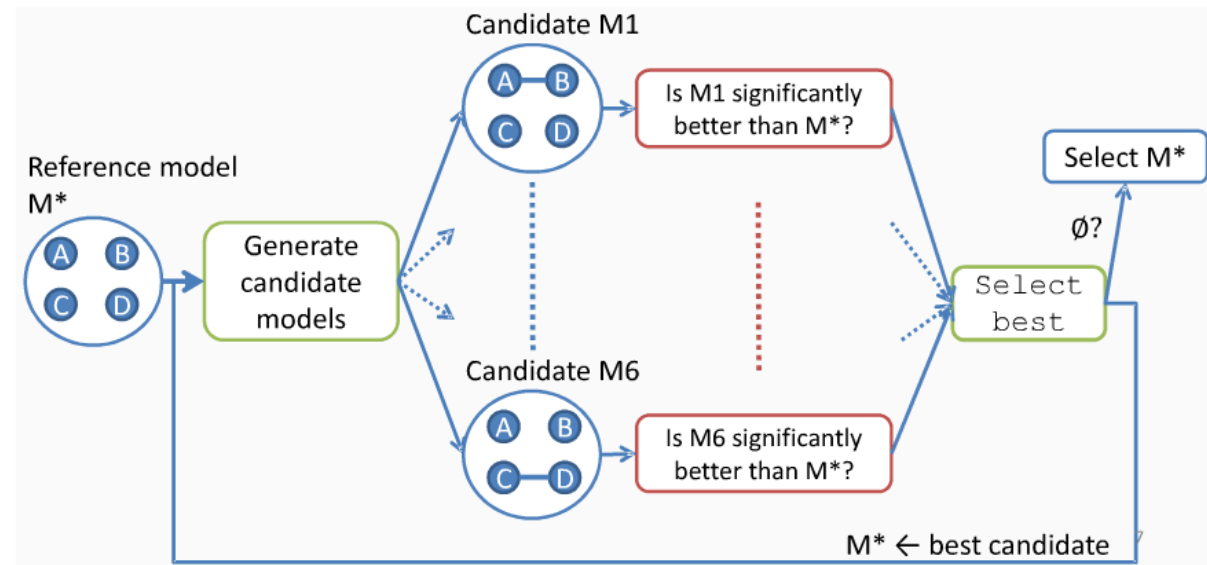


greedy search

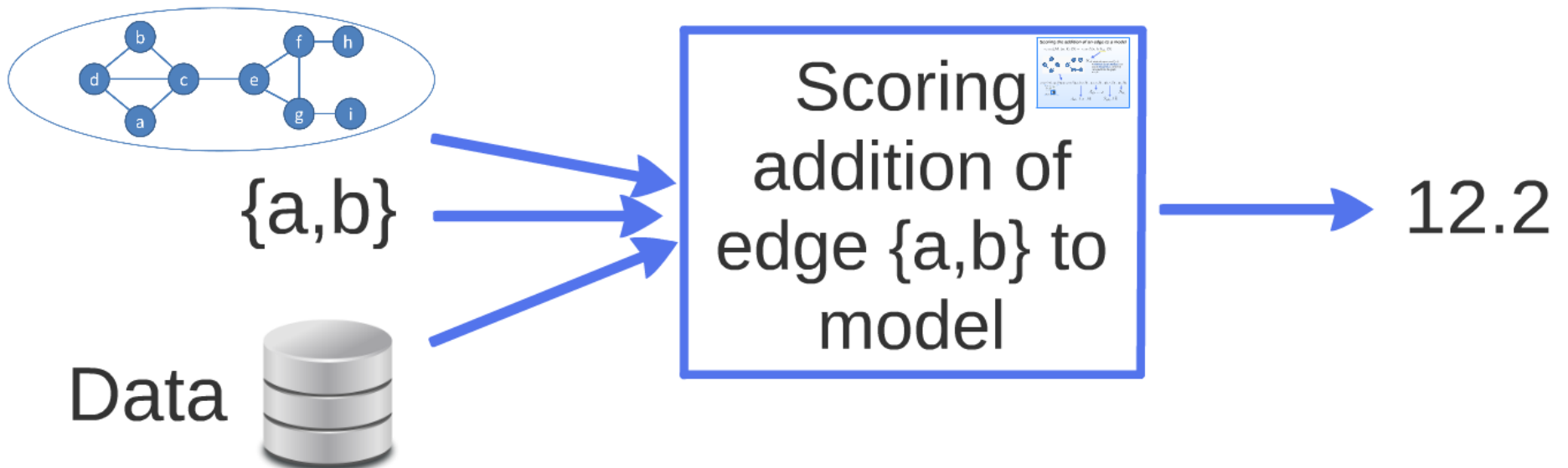


we only need...

Scoring in greedy search

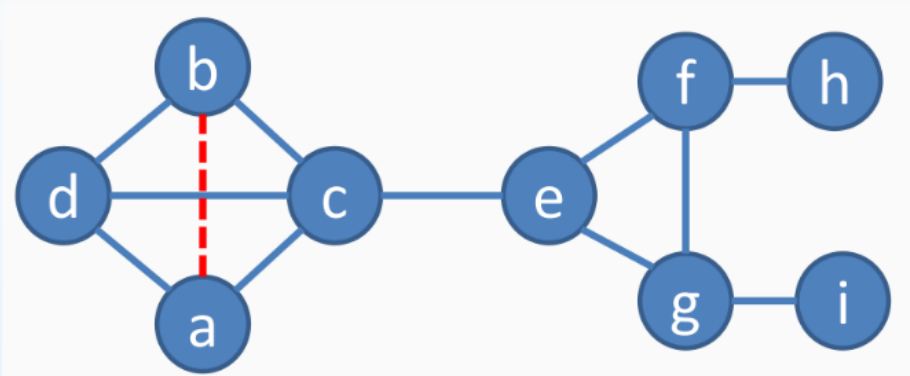


In this case, we only need...



Scoring the addition of an edge to a model

$$\text{score}(\mathcal{M}, (a, b), \mathcal{D}) = \text{score}'(a, b, S_{ab}, \mathcal{D})$$



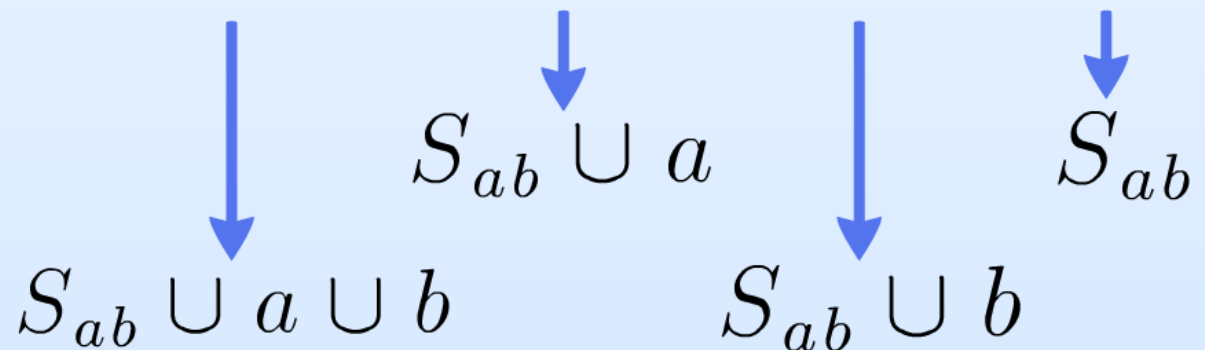
S_{ab} : minimal separator of (a,b)
 = **minimal set of vertices** that would **disconnect** a from b if removed from the graph
 = {c,d}

$$\text{score}(\mathcal{M}, \{a, b\}) = \text{score}'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

This has been proven for different scorings

- Statistical tests (G-test) [1] ✓
- MML/MDL [2] ✓
- Entropy / KL divergence [3] ✓

[1] P. Poignon et al., "Scoring log-linear analysis by high dimensional MML", in ICML 2013.
 [2] P. Poignon et al., "A statistically efficient and scalable method for Bayesian model selection in high-dimensional data", in ICML 2014.
 [3] A. Chervinskii et al., "Efficient variable selection in decomposable models", in IJAI 2011.

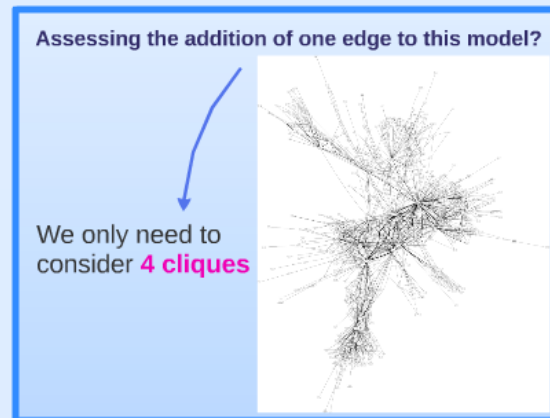


This has been proven for different scorings

- Statistical tests (G-test) [1] ✓

- MML/MDL [2] ✓

- Entropy / KL divergence [3] ✓



[1]: F. Petitjean *et al.*, "Scaling log-linear analysis to high-dimensional data," in *ICDM 2013*.

[2]: F. Petitjean *et al.*, "A statistically efficient and scalable method for log-linear analysis of high-dimensional data," in *ICDM 2014*.

[3]: A. Deshpande *et al.*, "Efficient stepwise selection in decomposable models," in *UAI 2001*.

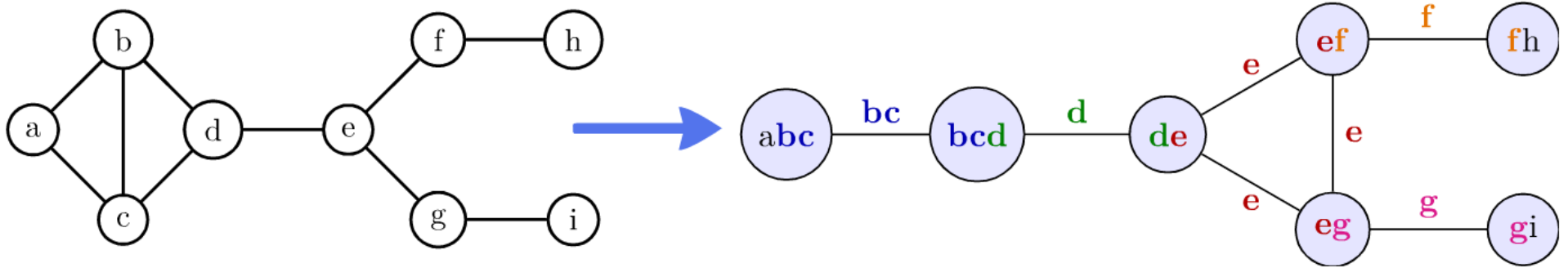
Assessing the addition of one edge to this model?



We only need to consider **4 cliques**



Clique graph (CG)



Definition of a clique-graph: [1]

- Maximal cliques of the graph \Rightarrow nodes of the clique-graph (CG)
- (C_1, C_2) in CG iff $\forall a \in (C_1 \setminus C_2), \forall b \in (C_2 \setminus C_1), S_{ab} = C_1 \cap C_2$

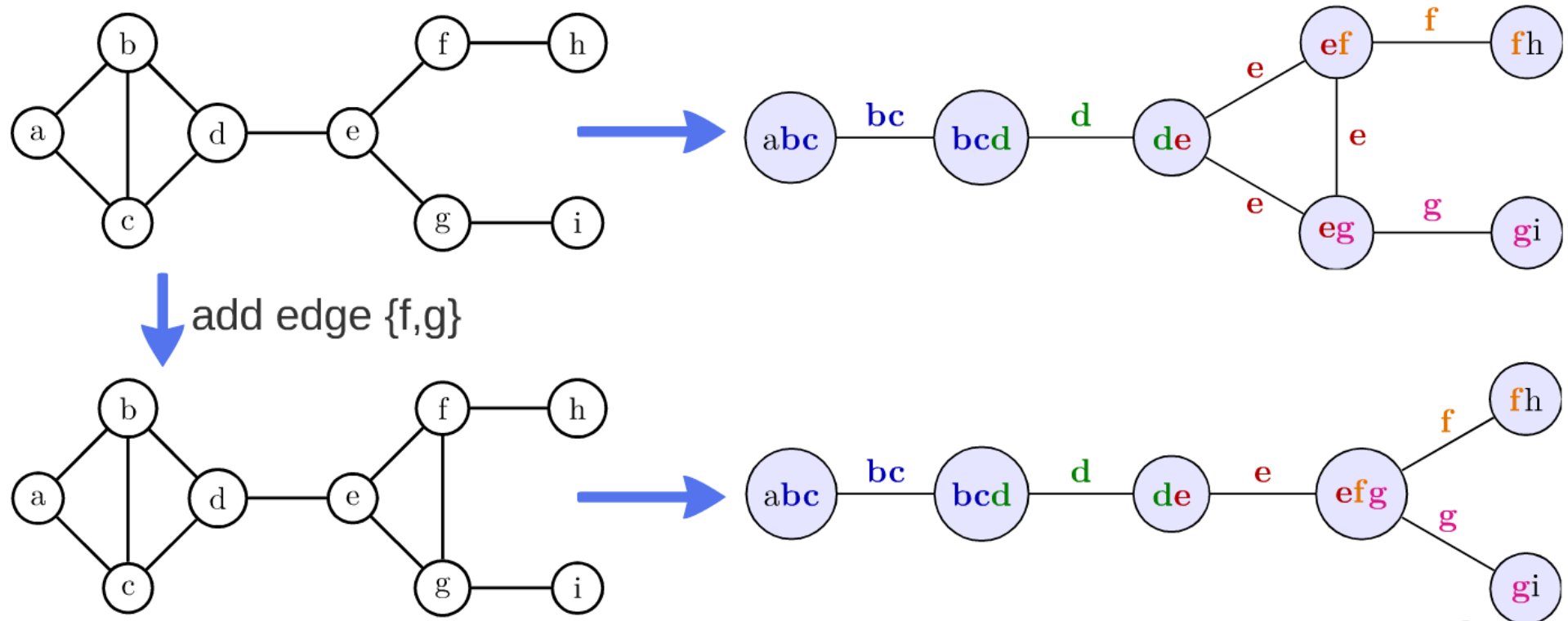
→ The **clique-graph** holds the information about the minimal vertex separators of all potential edges [1].

→ The **clique-graph** can directly tell us if an edge can be added to the graph while keeping it chordal.

→ Maximal cliques computed in $O(n+m)$ with MCS or BFS.

→ Edges computed in one pass over the cliques (see "Weak Triangulation Lemma" in [1])

Clique graph and greedy search



We can directly update the structure of the clique graph [1] ✓

This means that we can quickly identify minimal separators and thus know what cliques to use when scoring the addition of an edge to the current model.

$$\text{score}(\mathcal{M}, \{a, b\}) = \text{score}'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

Search and statistical paradigm

Frequentist approaches



- Currently best statistical efficiency [1]
- Parameter-free (no priors to define)
- Only greedy search, because can only score the comparison of nested models
- Growing criticism of the community when used directly for decision making (see for example [2])

[1] Petitjean, Nicholson and Webb, Scaling log-linear analysis to high-dimensional data, IEEE ICDM 2013.
 [2] Nuzzo, "Scientific method: Statistical errors", Nature 2014.

$$P(\mathbf{x}) = \lim_{n_t \rightarrow \infty} \frac{n_{\mathbf{x}}}{n_t}$$

Bayesian approaches



- Randomized search available, because it scores models independently
- Makes it possible to integrate priors
- Easier integration in a decision making process
- Not parameter-free
- Currently inferior statistical performance*

* So far, no Bayesian scoring has been specifically developed for decomposable models (only MDL/MML [1,2]) → **Open**

[1] Altmueller and Haraick, Approximating High Dimensional Probability Distributions, ICPR 2004.
 [2] Petitjean, Allison and Webb, A statistically efficient and scalable method for log-linear analysis of high-dimensional data, IEEE ICDM 2014.

aside note

Multiple testing

Frequentist approaches:

1. Choose a significance level α (eg = 0.01)
2. Assess probability p of observing data given null hypothesis
3. If $p < \alpha$ then reject null hypothesis

→ This guarantees that the chance of falsely rejecting the null hypothesis is less than α

Why do we need multiple testing corrections?

$$\begin{aligned}
 p(\text{making an error in 1 test} \mid \text{null is true}) &= \alpha \\
 p(\text{not making an error in 1 test} \mid \text{null is true}) &= 1 - \alpha \\
 p(\text{not making an error in } T \text{ tests} \mid \text{null is true}) &= (1 - \alpha)^T \\
 p(\text{making at least one error in } T \text{ tests} \mid \text{null is true}) &= 1 - (1 - \alpha)^T
 \end{aligned}$$

Standard solution: choose $\alpha' = \frac{\alpha}{T}$ (Bonferroni)

But, for model selection, we do not know T

solutions



Frequentist approaches



- Currently best statistical efficiency [1]
- Parameter-free (no priors to define)



- Only greedy search, because can only score the comparison of nested models
- Growing criticism of the community when used directly for decision making (see for example [2])

[1] **Petitjean**, Nicholson and **Webb**, Scaling log-linear analysis to high-dimensional data, IEEE ICDM 2013.

[2] Nuzzo, "Scientific method: Statistical errors", Nature 2014.

$$P(\mathbf{x}) = \lim_{n_t \rightarrow \infty} \frac{n_{\mathbf{x}}}{n_t}$$


Bayesian approaches



- Randomized search available, because it scores models independently
- Makes it possible to integrate priors
- Easier integration in a decision making process

- Not parameter-free
- Currently inferior statistical performance*



* So far, no Bayesian scoring has been specifically developed for decomposable models (only MDL/MML [1,2])  **Open**

[1] Altmueller and Haralick, Approximating High Dimensional Probability Distributions, ICPR 2004.

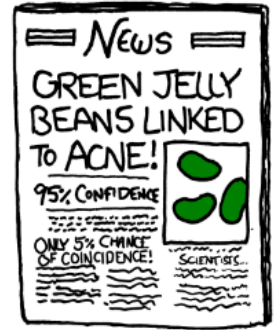
[2] **Petitjean**, Allison and **Webb**, A statistically efficient and scalable method for log-linear analysis of high-dimensional data, IEEE ICDM 2014.

Multiple testing

Frequentist approaches:

1. Choose a significance level α (eg $= 0.01$)
2. Assess probability p of observing data given null hypothesis
3. If $p < \alpha$ then reject null hypothesis

→ This guarantees that the chance of falsely rejecting the null hypothesis is less than α



Why do we need multiple testing corrections?

$$p(\text{making an error in 1 test} \mid \text{null is true}) = \alpha$$

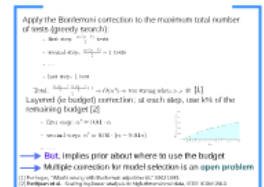
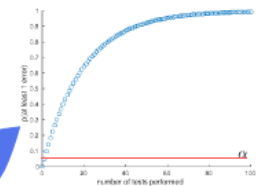
$$p(\text{not making an error in 1 test} \mid \text{null is true}) = 1 - \alpha$$

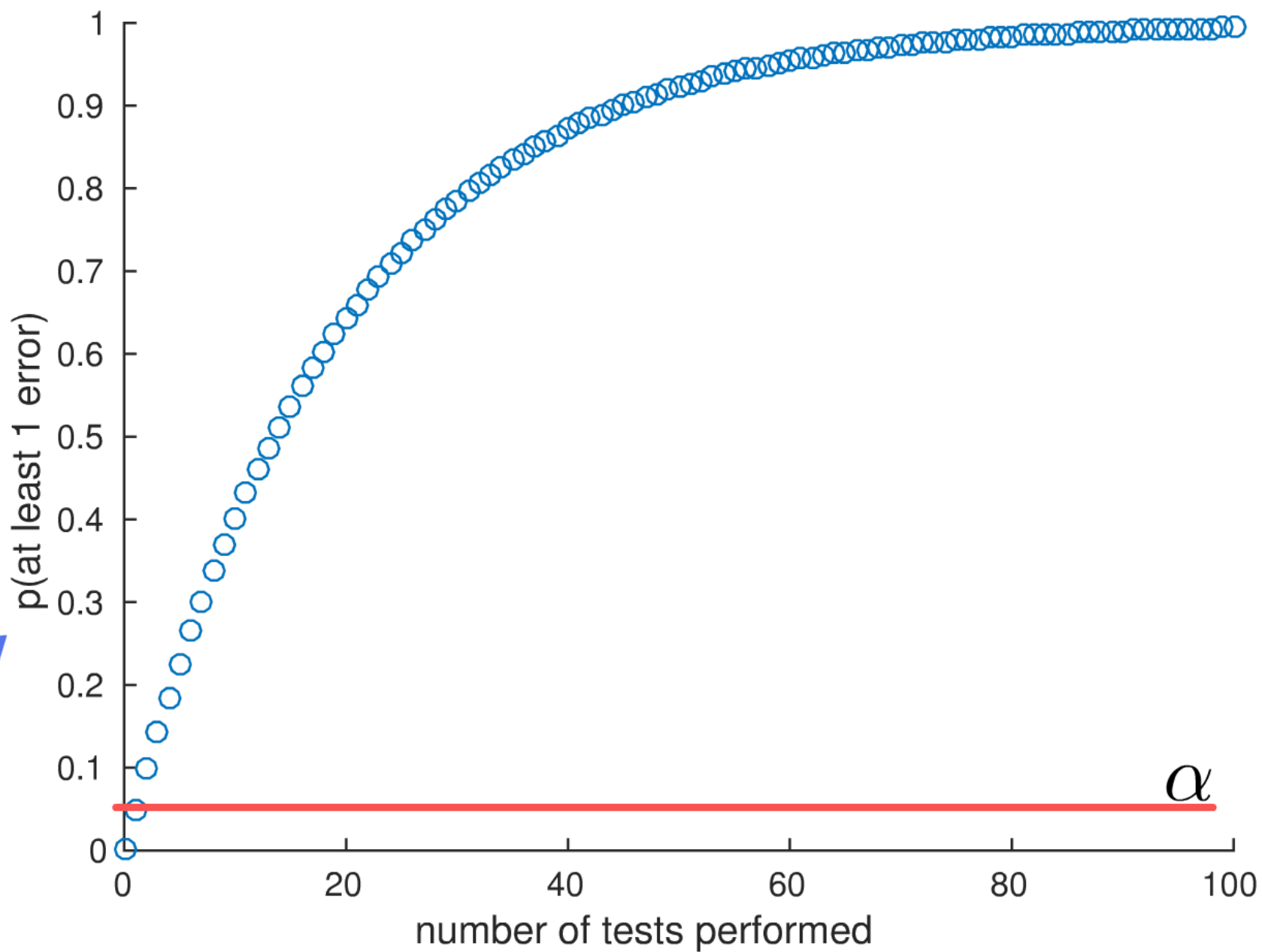
$$p(\text{not making an error in } T \text{ tests} \mid \text{null is true}) = (1 - \alpha)^T$$

$$p(\text{making at least one error in } T \text{ tests} \mid \text{null is true}) = 1 - (1 - \alpha)^T$$

Standard solution: choose $\alpha' = \frac{\alpha}{T}$ (Bonferroni)

But, for model selection, we do not know T solutions



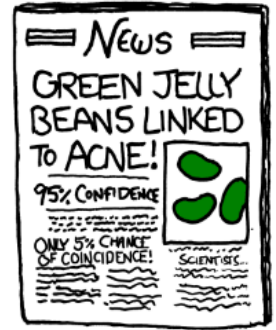


Multiple testing

Frequentist approaches:

1. Choose a significance level α (eg $= 0.01$)
2. Assess probability p of observing data given null hypothesis
3. If $p < \alpha$ then reject null hypothesis

→ This guarantees that the chance of falsely rejecting the null hypothesis is less than α



Why do we need multiple testing corrections?

$$p(\text{making an error in 1 test} \mid \text{null is true}) = \alpha$$

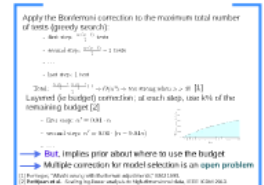
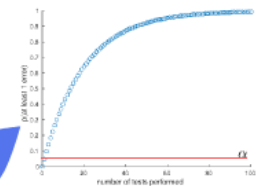
$$p(\text{not making an error in 1 test} \mid \text{null is true}) = 1 - \alpha$$

$$p(\text{not making an error in } T \text{ tests} \mid \text{null is true}) = (1 - \alpha)^T$$

$$p(\text{making at least one error in } T \text{ tests} \mid \text{null is true}) = 1 - (1 - \alpha)^T$$

Standard solution: choose $\alpha' = \frac{\alpha}{T}$ (Bonferroni)

But, for model selection, we do not know T solutions



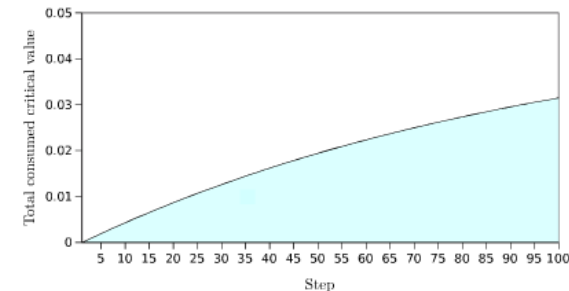
Apply the Bonferroni correction to the maximum total number of tests (greedy search):

- first step: $\frac{n \cdot (n-1)}{2}$ tests
- second step: $\frac{n \cdot (n-1)}{2} - 1$ tests
- ...
- last step: 1 test

Total: $\frac{\frac{n \cdot (n-1)}{2} \cdot \frac{n \cdot (n-1)}{2} + 1}{2} \Rightarrow O(n^4) \Rightarrow$ too strong when $n > 30$ [1]

Layered (ie budget) correction; at each step, use k% of the remaining budget [2]

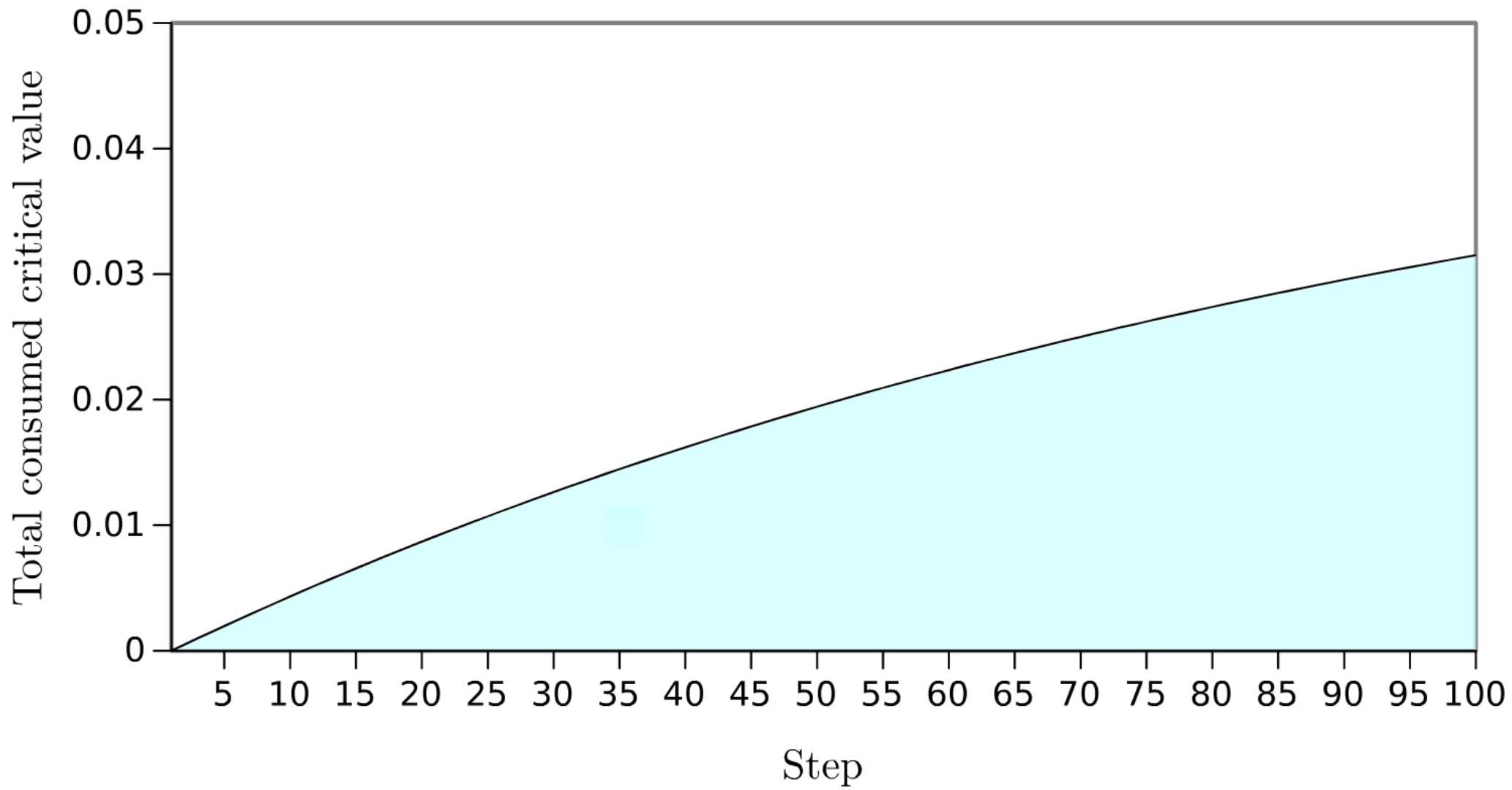
- first step: $\alpha' = 0.01 \cdot \alpha$
- second step: $\alpha' = 0.01 \cdot (\alpha - 0.01\alpha)$
- ...



- ➔ **But**, implies prior about where to use the budget
- ➔ Multiple correction for model selection is an **open problem**

[1] Perneger, "What's wrong with Bonferroni adjustments," BMJ 1998.

[2] Petitjean et al., Scaling log-linear analysis to high-dimensional data, IEEE ICDM 2013.



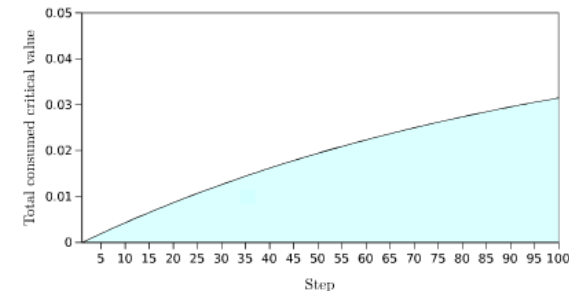
Apply the Bonferroni correction to the maximum total number of tests (greedy search):

- first step: $\frac{n \cdot (n-1)}{2}$ tests
- second step: $\frac{n \cdot (n-1)}{2} - 1$ tests
- ...
- last step: 1 test

Total: $\frac{\frac{n \cdot (n-1)}{2} \cdot \frac{n \cdot (n-1)}{2} + 1}{2} \Rightarrow O(n^4) \Rightarrow$ too strong when $n > 30$ [1]

Layered (ie budget) correction; at each step, use k% of the remaining budget [2]

- first step: $\alpha' = 0.01 \cdot \alpha$
- second step: $\alpha' = 0.01 \cdot (\alpha - 0.01\alpha)$
- ...



- ➔ **But**, implies prior about where to use the budget
- ➔ Multiple correction for model selection is an **open problem**

[1] Perneger, "What's wrong with Bonferroni adjustments," BMJ 1998.

[2] Petitjean et al., Scaling log-linear analysis to high-dimensional data, IEEE ICDM 2013.

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model? Probabilistic graphical models (PGMs) are a compact representation of probability distributions. They are used for: Quantifying uncertainty, Not a trade-off in classical AI, and Aids Complexity representing probability distributions.

What are graphical models useful for? They are used for: Image classification, Recommendation systems, and many other applications. They are the thousands of applications of these methods.

What we will and will not cover: We will cover: Graphical models, Inference, Learning. We will not cover: Deep learning, Reinforcement learning, etc.

Graphical models 101

Classes of graphical models: Directed Acyclic Graphs (DAGs), Undirected Graphs (UGs), and Markov Random Fields (MRFs).

A simple example of structure learning: Hill-climbing search on MRF using AIC. Example: Image classification.

Graph theory

Maximal cliques and minimal separators: Definitions and examples.

What are decomposable models: Decomposable models are a Markov Random Field for which the graph is chordal. Example: Bayesian network.

Properties of decomposable models: 1. Closed form for the partition function, 2. No global optimization, 3. Junction tree algorithm, 4. No 2ⁿ global search, 5. Linear-time inference, 6. Interaction between BN and MRF.

Evaluation - Scoring

Decomposable models are essential for scalability, because... We need: 1. scalable scoring, 2. efficient search, 3. scalable belief propagation.

Bottom line: A score decomposable models is essential to: 1. AIC for MRFs, 2. A set of operations to compute MRFs.

Most scores are scalable: Entropy [1], Mutual Information [2], etc.

Break

Efficient search

Scoring in greedy search: In this case, we only need: Scoring of edge (i,j) to 12.2.

Clique graph (CG): Definition and properties.

Clique graph and greedy search: How to use the clique graph for scoring.

Search and statistical paradigm: Comparison of different search methods.

The nitty-gritty

Counting efficiently: Scoring for example with KL minimized when... What does it mean to compute $D(X||Y)$?

Counting efficiently (2): How to count efficiently using the clique graph.

Memorization: How to use the clique graph for counting.

Addition of the same edge to different reference models: How to add an edge to a reference model.

Use cases

Study of the elderly: 25 variables, 15,000 patients.

Insurance customer management: 93 variables, 6,000 customers.

Portfolio management: 500 variables, 20 years of trading.

Wrapping up!

This tutorial in a nutshell: 1. Graphical models are everywhere, 2. Graphical models are a compact representation of high-order probabilistic distributions, 3. We can learn graphical models from datasets with 1,000+ variables, 4. It is possible to do inference on graphs in the library that are exponentially larger than dataset.

Open problems: 1. Efficient constrained search, 2. Better scores (eg on Directed acyclic graphs), 3. Efficient coding of marginal "bits", 4. Efficient data structures for counting, 5. Learning out of core, 6. Latent variables.

Open problems (2): 1. How to handle numerical variables, 2. How to handle missing values?, 3. Learning accurate parameters in large tables, 4. Many problems are low-hanging fruit, you just need to push them!

Counting efficiently



Scoring - for example with KL minimized when...

$$\sum_{C \in \mathcal{C}} H(X_C) - \sum_{S \in \mathcal{S}} H(X_S) \text{ is minimized.}$$

What does it mean to compute $H(X_C)$?

Take *clique* = ABC :

$H(A, B, C)$ → efficient scoring boils down to **efficient counting**

$$= -\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

where $O_{A=a, B=b, C=c}$ is how many instances in the dataset have $A = a$ and $B = b$ and $C = c$.

Counting efficiently (2)



$$H(ABC) = -\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Being able to quickly count how many instances with this configuration of A,B,C

→ Vertical representation of the dataset

What does that change?

→ How many **tall females** in the dataset?

$$O_{G=female, H=tall} = \left| TIDs(\text{Gender} = \text{female}) \cap TIDs(\text{Height} = \text{tall}) \right|$$

→ Data structure for fast intersection

Vertical representation

	TID	Gender	Age	Height
Horizontal	1	female	60+	tall
	2	female	10-20	short
	3	male	40-50	tall
	⋮	⋮	⋮	⋮
	14,329	female	10-20	tall
14,330	male	60+	short	
Vertical	TIDs(Gender = female) =		{1, 2, ..., 14329}	
	TIDs(Gender = male) =		{3, ..., 14330}	
	TIDs(Height = tall) =		{1, 3, ..., 14329}	

Data structures for TID sets

Sorted sets of integers

TIDs(Gender = female) = {1, 2, ..., 14329}

Advantage: intersection in O(size of the largest TID set)

Drawback: storage (N x 32bits)

=> Good for sparse data

Bitmaps

Advantages:

- intersection time independent of data sparsity
- storage N x 1 x "avg attribute cardinality" bits

Drawback: intersection in O(N) - but fast implementation

... see also compressed bitmaps (Roaring bitmaps [1], Concise [2], etc.)

[1] Charbiv et al., "Better bitmap performance with Roaring bitmaps," in Software Practice and Experience (to appear)
 [2] Coluberto et al., "Concise: Compressed W-Composable Integer Set," Information Processing Letters, 2019

Vertical representation

Horizontal

TID	Gender	Age	Height
1	female	60+	tall
2	female	10-20	short
3	male	40-50	tall
⋮	⋮	⋮	⋮
14,329	female	10-20	tall
14,330	male	60+	short

Vertical



$$\begin{aligned}TIDs(\text{Gender} = \text{female}) &= \{1, 2, \dots, 14329\} \\TIDs(\text{Gender} = \text{male}) &= \{3, \dots, 14330\} \\&\vdots \\TIDs(\text{Height} = \text{tall}) &= \{1, 3, \dots, 14329\}\end{aligned}$$

Counting efficiently (2)



$$H(ABC) = -\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Being able to quickly count how many instances with this configuration of A,B,C

→ Vertical representation of the dataset

What does that change?

→ How many **tall females** in the dataset?

$$O_{G=female, H=tall} = \left| TIDs(\text{Gender} = \text{female}) \cap TIDs(\text{Height} = \text{tall}) \right|$$

→ Data structure for fast intersection

Vertical representation

	TID	Gender	Age	Height
Horizontal	1	female	60+	tall
	2	female	10-20	short
	3	male	40-50	tall
	⋮	⋮	⋮	⋮
	14,329	female	10-20	tall
14,330	male	60+	short	
Vertical	TIDs(Gender = female) =		{1, 2, ..., 14329}	
	TIDs(Gender = male) =		{3, ..., 14330}	
	TIDs(Height = tall) =		{1, 3, ..., 14329}	

Data structures for TID sets

Sorted sets of integers

TIDs(Gender = female) = {1, 2, ..., 14329}

Advantage: intersection in O(size of the largest TID set)

Drawback: storage (N x 32bits)

=> Good for sparse data

Bitmaps

Advantages:

- intersection time independent of data sparsity
- storage N x 1 x "avg attribute cardinality" bits

Drawback: intersection in O(N) - but fast implementation

... see also compressed bitmaps (Roaring bitmaps [1], Concise [2], etc.)

[1] Charb et al., "Better bitmap performance with Roaring bitmaps," in Software Practice and Experience (to appear)
 [2] Coluberto et al., "Concise: Compressed W-Composable Integer Set," Information Processing Letters, 2019

Data structures for TID sets

Sorted sets of integers

$$TIDs(\text{Gender} = \text{female}) = \{1, 2, \dots, 14329\}$$

Advantage: intersection in $O(\text{size of the largest TID set})$

Drawback: storage ($N \times 32\text{bits}$)

=> Good for sparse data

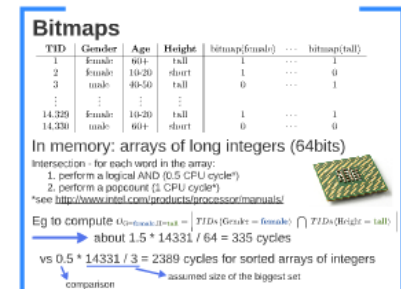
Bitmaps

Advantages:

- intersection time independent of data sparsity
- storage $N \times 1 \times$ "avg attribute cardinality" bits

Drawback:

- intersection in $O(N)$ - but **fast implementation**



... see also compressed bitmaps
(Roaring bitmaps [1], Concise [2], etc.)

[1] Chambi et al., "Better bitmap performance with Roaring bitmaps," in *Software: Practice and Experience* (to appear)

[2] Colantonio et al., "Concise: Compressed 'n' Composible Integer Set," *Information Processing Letters*, 2010

Bitmaps

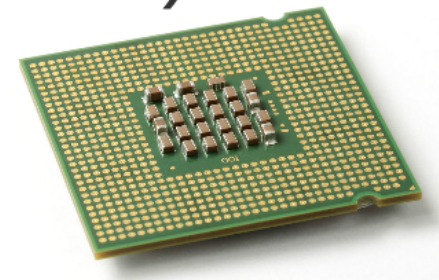
TID	Gender	Age	Height	bitmap(female)	...	bitmap(tall)
1	female	60+	tall	1	...	1
2	female	10-20	short	1	...	0
3	male	40-50	tall	0	...	1
⋮	⋮	⋮	⋮			
14,329	female	10-20	tall	1	...	1
14,330	male	60+	short	0	...	0

In memory: arrays of long integers (64bits)

Intersection - for each word in the array:

1. perform a logical AND (0.5 CPU cycle*)
2. perform a popcount (1 CPU cycle*)

*see <http://www.intel.com/products/processor/manuals/>



Eg to compute $O_{G=female, H=tall} = \left| TIDs(\text{Gender} = \text{female}) \cap TIDs(\text{Height} = \text{tall}) \right|$

→ about $1.5 * 14331 / 64 = 335$ cycles

vs $0.5 * \underline{14331} / 3 = 2389$ cycles for sorted arrays of integers

comparison

assumed size of the biggest set

Data structures for TID sets

Sorted sets of integers

$$TIDs(\text{Gender} = \text{female}) = \{1, 2, \dots, 14329\}$$

Advantage: intersection in $O(\text{size of the largest TID set})$

Drawback: storage ($N \times 32\text{bits}$)

=> Good for sparse data

Bitmaps

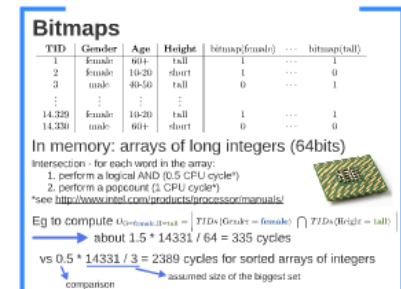
Advantages:

- intersection time independent of data sparsity
- storage $N \times 1 \times$ "avg attribute cardinality" bits

Drawback:

- intersection in $O(N)$ - but **fast implementation**

... see also compressed bitmaps
(Roaring bitmaps [1], Concise [2], etc.)



[1] Chambi et al., "Better bitmap performance with Roaring bitmaps," in *Software: Practice and Experience* (to appear)
[2] Colantonio et al., "Concise: Compressed 'n' Composible Integer Set," *Information Processing Letters*, 2010

Counting efficiently (2)



$$H(ABC) = -\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Being able to quickly count how many instances with this configuration of A,B,C

→ Vertical representation of the dataset

What does that change?

→ How many **tall females** in the dataset?

$$O_{G=female, H=tall} = \left| TIDs(\text{Gender} = \text{female}) \cap TIDs(\text{Height} = \text{tall}) \right|$$

→ Data structure for fast intersection

Vertical representation

	TID	Gender	Age	Height
Horizontal	1	female	60+	tall
	2	female	10-20	short
	3	male	40-50	tall
	⋮	⋮	⋮	⋮
	14,329	female	10-20	tall
14,330	male	60+	short	
Vertical	TIDs(Gender = female) =		{1, 2, ..., 14329}	
	TIDs(Gender = male) =		{3, ..., 14330}	
	TIDs(Height = tall) =		{1, 3, ..., 14329}	

Data structures for TID sets

Sorted sets of integers

TIDs(Gender = female) = {1, 2, ..., 14329}

Advantage: intersection in O(size of the largest TID set)

Drawback: storage (N x 32bits)

=> Good for sparse data

Bitmaps

Advantages:

- intersection time independent of data sparsity
- storage N x 1 x "avg attribute cardinality" bits

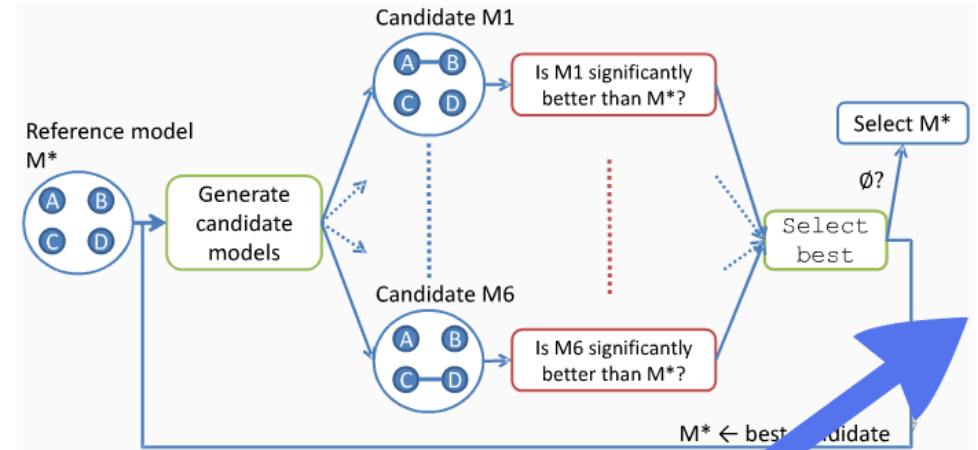
Drawback: intersection in O(N) - but fast implementation

... see also compressed bitmaps (Roaring bitmaps [1], Concise [2], etc.)

[1] Charab et al., "Better bitmap performance with Roaring bitmaps," in Software Practice and Experience (to appear)
 [2] Coluberto et al., "Concise: Compressed W-Comparable Integer Set," Information Processing Letters, 2019

Memoization

From the high-level perspective, many elements of the process will be repeated:



→ Addition of same edge considered several times (to different models)

→ Different edges' scores might share sub-scores

$$score(\mathcal{M}, \{a, b\}) = score'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

→ Different sub-scores scores might share elements

$$-\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Memoization of clique sub-scores

Reminder: with 4 values per variables, a clique of size 4 will have to iterate over 65,535 combinations of values, eg summing over 65,535 cells → not negligible

Use a **hashmap** to sub-score associated to each clique.

Hashing function: $V = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$
 $ECML = 0000000001$
 $A(ECML) = 7366$

standard java hash

```
public int hashCode() {
    long h = 1234;
    long[] words = new long[words.length];
    for (int i = words.length; i <= words[i]; i++)
        h = Math.multiplyMod(h * 31 + words[i], N);
    return (int)(h ^ 21) * N;
}
```

Memoization and Entropy computation

Reminder: most clique scores are functions of the entropy (KL divergence, G-test, MDL, etc.)

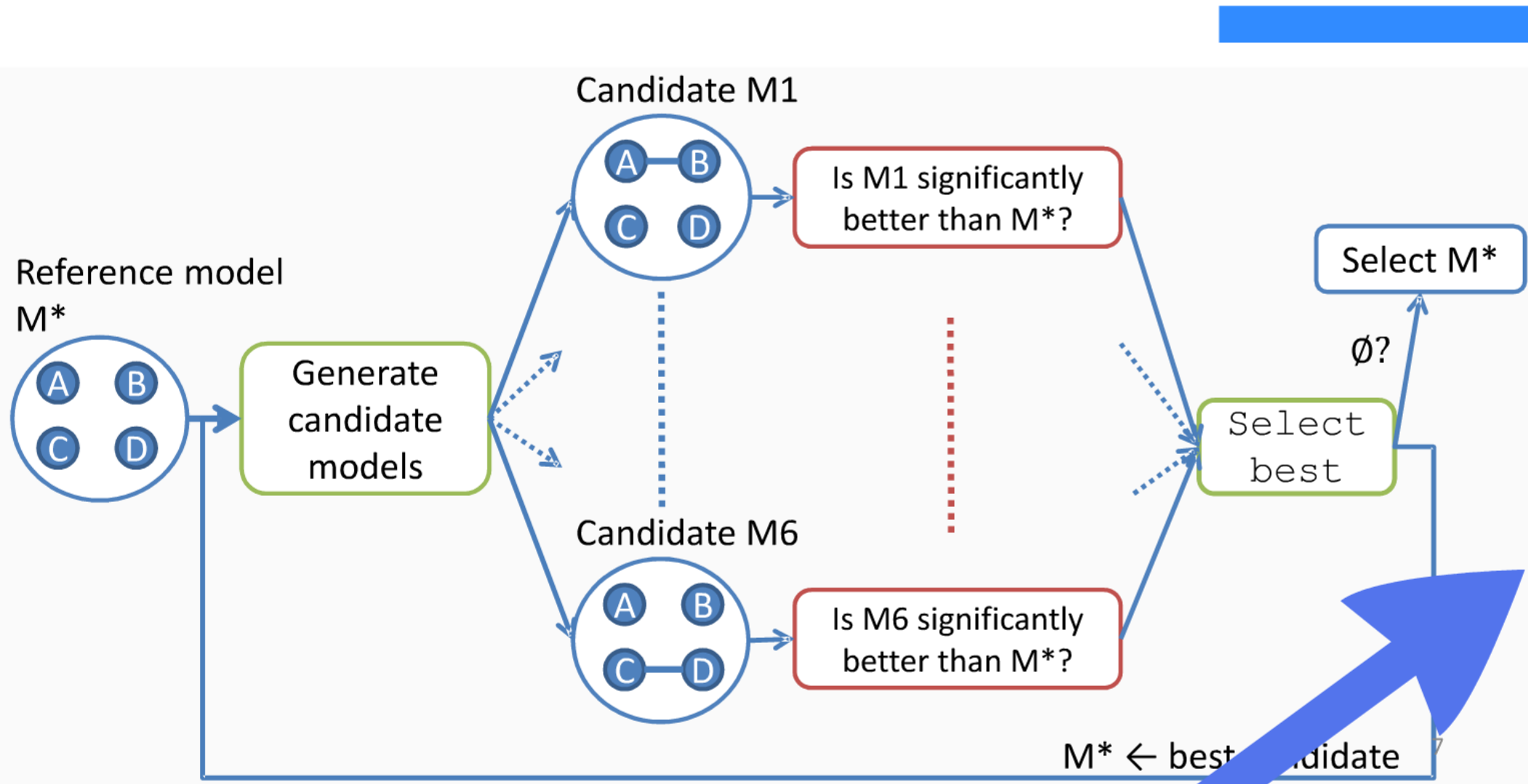
$$H(A) = -\frac{1}{N} \sum_{a \in A} O_a^A (\ln O_a^A - \ln N)$$

$$= -\frac{1}{N} \sum_{a \in A} \text{partial_entropy}(O_a^A)$$

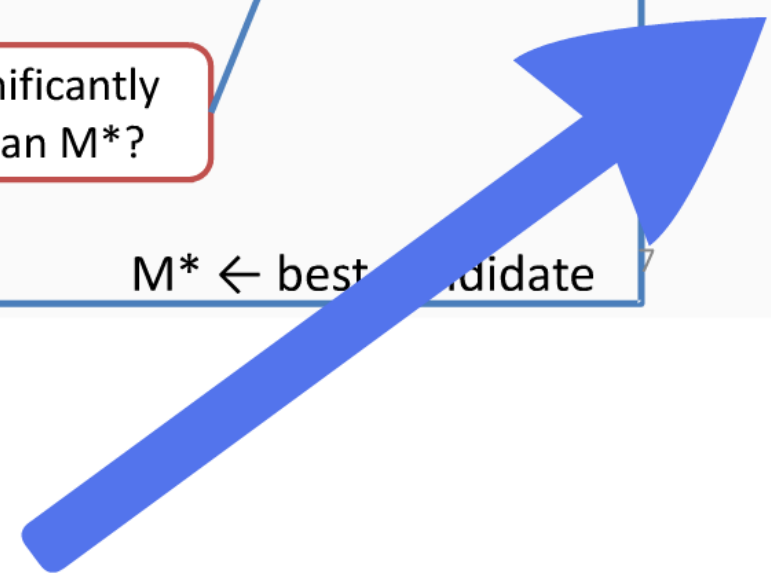
and... $\forall A, \forall a, O_a^A \in [0, N] \subset \mathbb{N}$

→ This means that we can precompute all possible "partial entropies" and store them in an array

This memoization makes the time spent in computing entropies to go from more than 99% to less than 1%

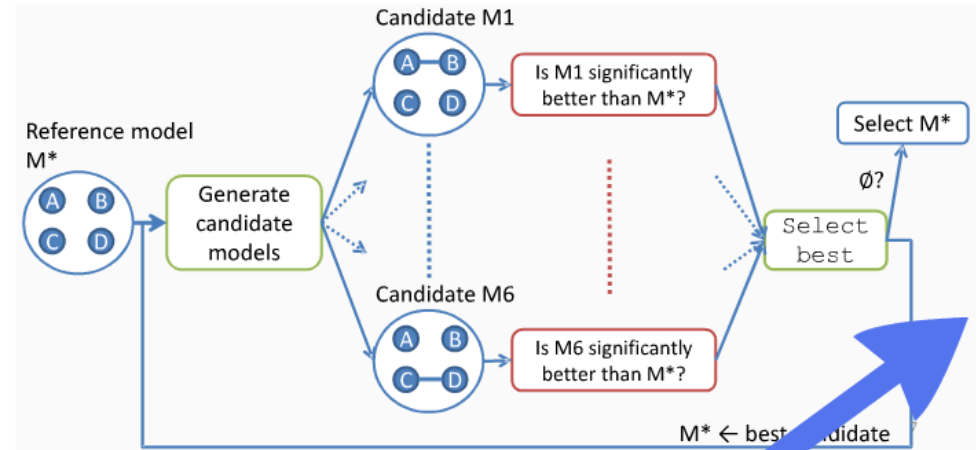


dered several times



Memoization

From the high-level perspective, many elements of the process will be repeated:



→ Addition of same edge considered several times (to different models)

→ Different edges' scores might share sub-scores

$$score(\mathcal{M}, \{a, b\}) = score'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

→ Different sub-scores scores might share elements

$$-\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Memoization of clique sub-scores

Reminder: with 4 values per variables, a clique of size 4 will have to iterate over 65,535 combinations of values, eg summing over 65,535 cells → not negligible

Use a **hashmap** to sub-score associated to each clique.

Hashing function: $V = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$
 $ECML = 0000000001$
 $A(ECML) = 7366$

standard java hash

```
public int hashCode() {
    long h = 1234;
    for (int i = 0; i < words.length; i++)
        for (int j = 0; j < words[i].length; j++)
            h = h * words[i][j] + (i * j);
    return (int)(h ^ 21) * 81;
}
```

Memoization and Entropy computation

Reminder: most clique scores are functions of the entropy (KL divergence, G-test, MDL, etc.)

$$H(A) = -\frac{1}{N} \sum_{a \in A} O_a^A (\ln O_a^A - \ln N)$$

$$= -\frac{1}{N} \sum_{a \in A} \text{partial_entropy}(O_a^A)$$

and... $\forall A, \forall a, O_a^A \in [0, N] \subset \mathbb{N}$

→ This means that we can precompute all possible "partial entropies" and store them in an array

This memoization makes the time spent in computing entropies to go from more than 99% to less than 1%

Memoization and Entropy computation

Reminder: most clique scores are functions of the entropy (KL divergence, G-test, MDL, etc.)

$$\begin{aligned} H(A) &= -\frac{1}{N} \sum_{\mathbf{x} \in A} O_{\mathbf{x}}^A \cdot (\ln O_{\mathbf{x}}^A - \ln N) \\ &= -\frac{1}{N} \sum_{\mathbf{x} \in A} \text{partial_entropy}(O_{\mathbf{x}}^A) \end{aligned}$$

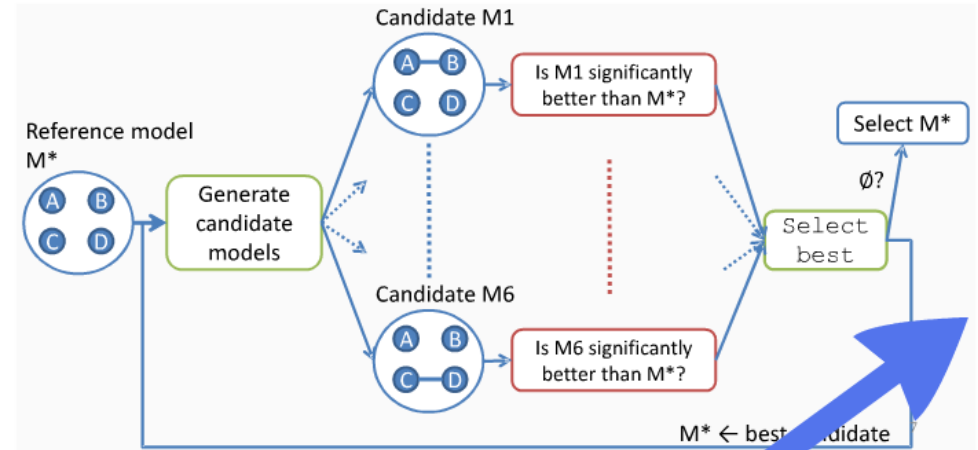
and... $\forall A, \forall x, O_{\mathbf{x}}^A \in [0, N] \subset \mathbb{N}$

→ This means that we can precompute all possible "partial entropies" and store them in an array

This memoization makes the time spent in computing entropies to go from more than **99%** to less than **1%**

Memoization

From the high-level perspective, many elements of the process will be repeated:



➔ Addition of same edge considered several times (to different models)

➔ Different edges' scores might share sub-scores

$$score(\mathcal{M}, \{a, b\}) = score'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

➔ Different sub-scores scores might share elements

$$-\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Memoization of clique sub-scores

Reminder: with 4 values per variables, a clique of size 4 will have to iterate over 65,535 combinations of values, eg summing over 65,535 cells → not negligible

Use a **hashmap** to sub-score associated to each clique.

Hashing function: $V = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$
 $ECML = 0000000001$
 $A(ECML) = 7366$

standard java hash

```
public int hashCode() {
    long h = 1234;
    long[] words = new long[words.length];
    for (int i = words.length; i <= words[i]; i++)
        h = Math.multiplyMod(h * 31 + words[i], N);
    return (int)(h ^ 21) * N;
}
```

Memoization and Entropy computation

Reminder: most clique scores are functions of the entropy (KL divergence, G-test, MDL, etc.)

$$H(A) = -\frac{1}{N} \sum_{a \in A} O_a^A (\ln O_a^A - \ln N)$$

$$= -\frac{1}{N} \sum_{a \in A} \text{partial_entropy}(O_a^A)$$

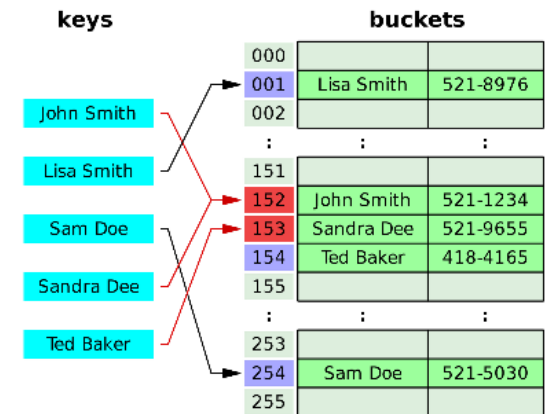
and... $\forall A, \forall a, O_a^A \in [0, N] \subset \mathbb{N}$

➔ This means that we can precompute all possible "partial entropies" and store them in an array

This memoization makes the time spent in computing entropies to go from more than 99% to less than 1%

Memoization of clique sub-scores

Reminder: with 4 values per variables, a clique of size 8 will have to iterate over 65,535 combinations of values, eg summing over 65,535 cells \longrightarrow not negligible



Use a **hashmap** to sub-score associated to each clique.

Hashing function: $\mathcal{V} = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$

ECML : 0010100000011

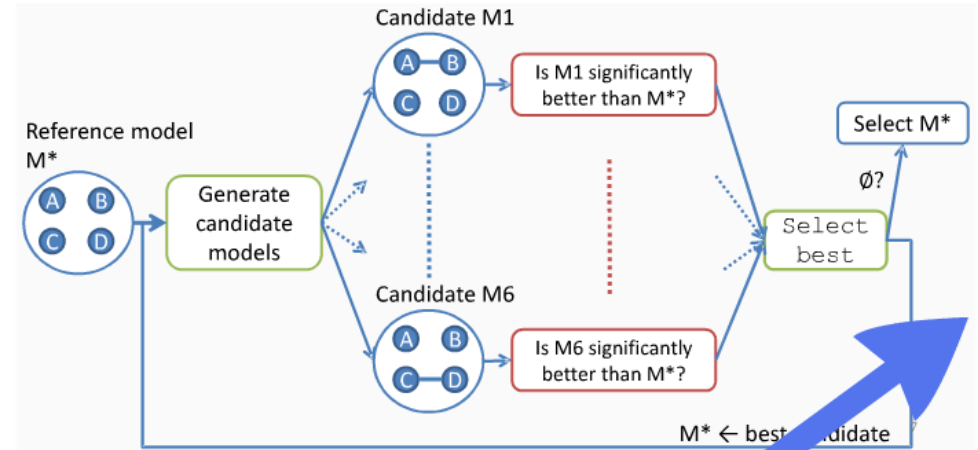
$h(ECML) = 7366$

standard java hash

```
public int hashCode() {
    long h = 1234;
    long[] words = toLongArray();
    for (int i = words.length; --i >= 0; )
        h ^= words[i] * (i + 1);
    return (int)((h >> 32) ^ h);
}
```


Memoization

From the high-level perspective, many elements of the process will be repeated:



→ Addition of same edge considered several times (to different models)

→ Different edges' scores might share sub-scores

$$score(\mathcal{M}, \{a, b\}) = score'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

→ Different sub-scores scores might share elements

$$-\frac{1}{N} \sum_{a \in A} \sum_{b \in B} \sum_{c \in C} O_{A=a, B=b, C=c} \cdot (\ln O_{A=a, B=b, C=c} - \ln N)$$

Memoization of clique sub-scores

Reminder: with 4 values per variables, a clique of size 4 will have to iterate over 65,535 combinations of values, eg summing over 65,535 cells → not negligible

Use a **hashmap** to sub-score associated to each clique.

Hashing function: $V = \{A, B, C, D, E, F, G, H, I, J, K, L, M\}$
 $ECML = 0000000001$
 $A(ECML) = 7366$

standard java hash

```

public int hashCode() {
    long h = 1234;
    for (int i = 0; i < words.length; i++)
        for (int j = 0; j < words[i].length; j++)
            h = h * words[i][j] + (i * j);
    return (int)(h ^ 21) * 11;
}
    
```

Memoization and Entropy computation

Reminder: most clique scores are functions of the entropy (KL divergence, G-test, MDL, etc.)

$$H(A) = -\frac{1}{N} \sum_{a \in A} O_a^A (\ln O_a^A - \ln N)$$

$$= -\frac{1}{N} \sum_{a \in A} \text{partial_entropy}(O_a^A)$$

and... $\forall A, \forall a, O_a^A \in [0, N] \subset \mathbb{N}$

→ This means that we can precompute all possible "partial entropies" and store them in an array

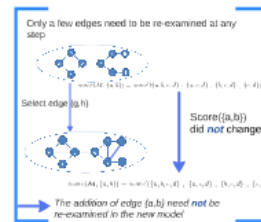
This memoization makes the time spent in computing entropies to go from more than 99% to less than 1%

Addition of the same edge to different reference models

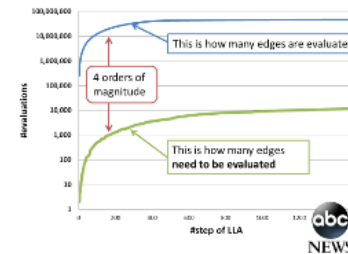
What we have seen so far:

- Evaluating the addition of an edge only depends upon 4 cliques of the graph

Our intuition



How often does that happen?



How can we use this information?



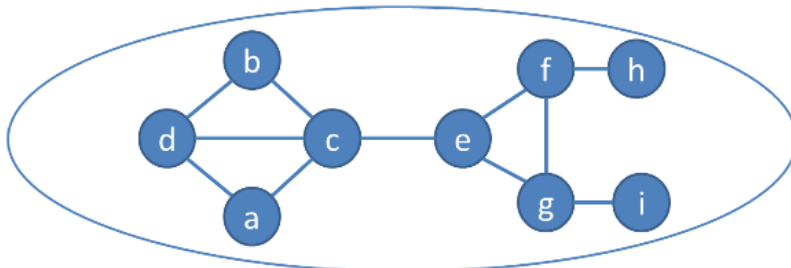
We know: if $S_{a,b}$ does not change between different modifications of the graph, then the addition of (a,b) need not be re-examined

1. Use a data structure that gives direct access to minimal separators for every potential edge
2. Keep track of the minimal separators for every potential edge
3. Maintain an ordered list of all the potential edges (priority queue)

Priority queue

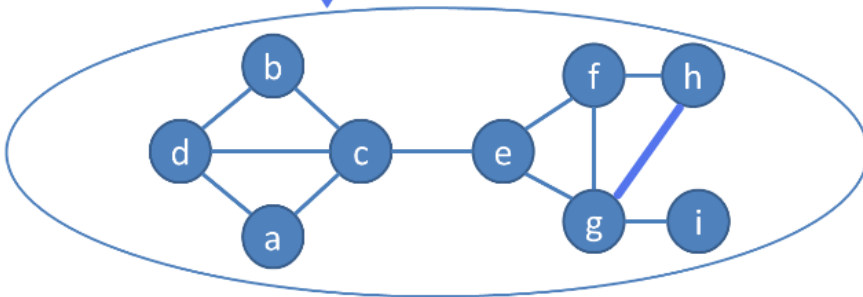
$n^2 \rightarrow n \cdot \log n$

Only a few edges need to be re-examined at any step



$$\text{score}(\mathcal{M}, \{a, b\}) = \text{score}'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

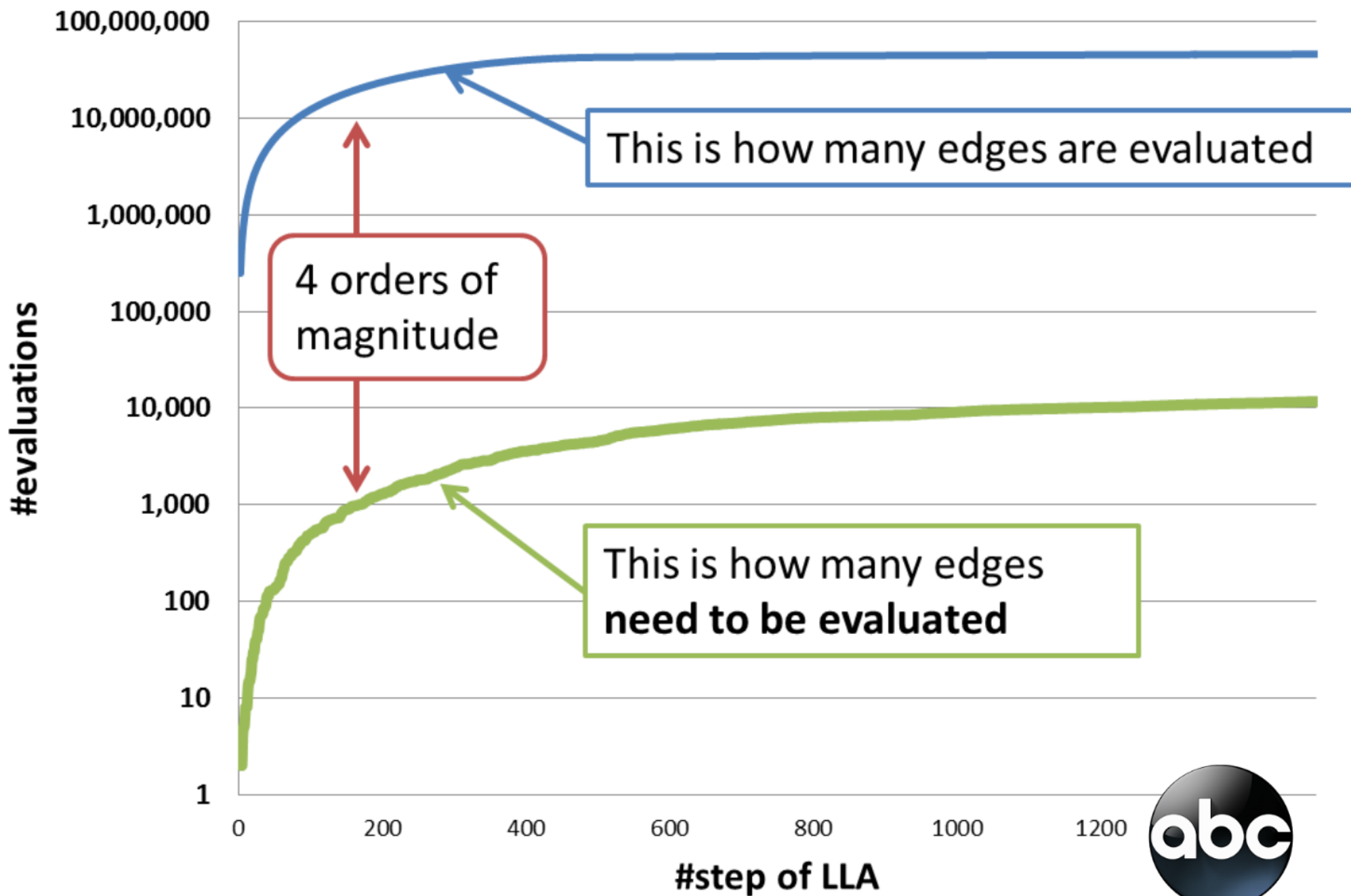
Select edge {g,h}



Score({a,b})
did **not** change

$$\text{score}(\mathcal{M}, \{a, b\}) = \text{score}'(\{a, b, c, d\}, \{a, c, d\}, \{b, c, d\}, \{c, d\})$$

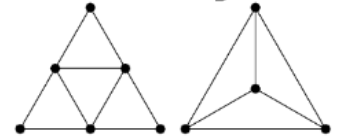
*The addition of edge {a,b} need **not** be re-examined in the new model*



We know: if S_{ab} does not change between different modifications of the graph, then the addition of $\{a,b\}$ need not be re-examined



1. Use a data structure that gives direct access to minimal separators for every potential edge



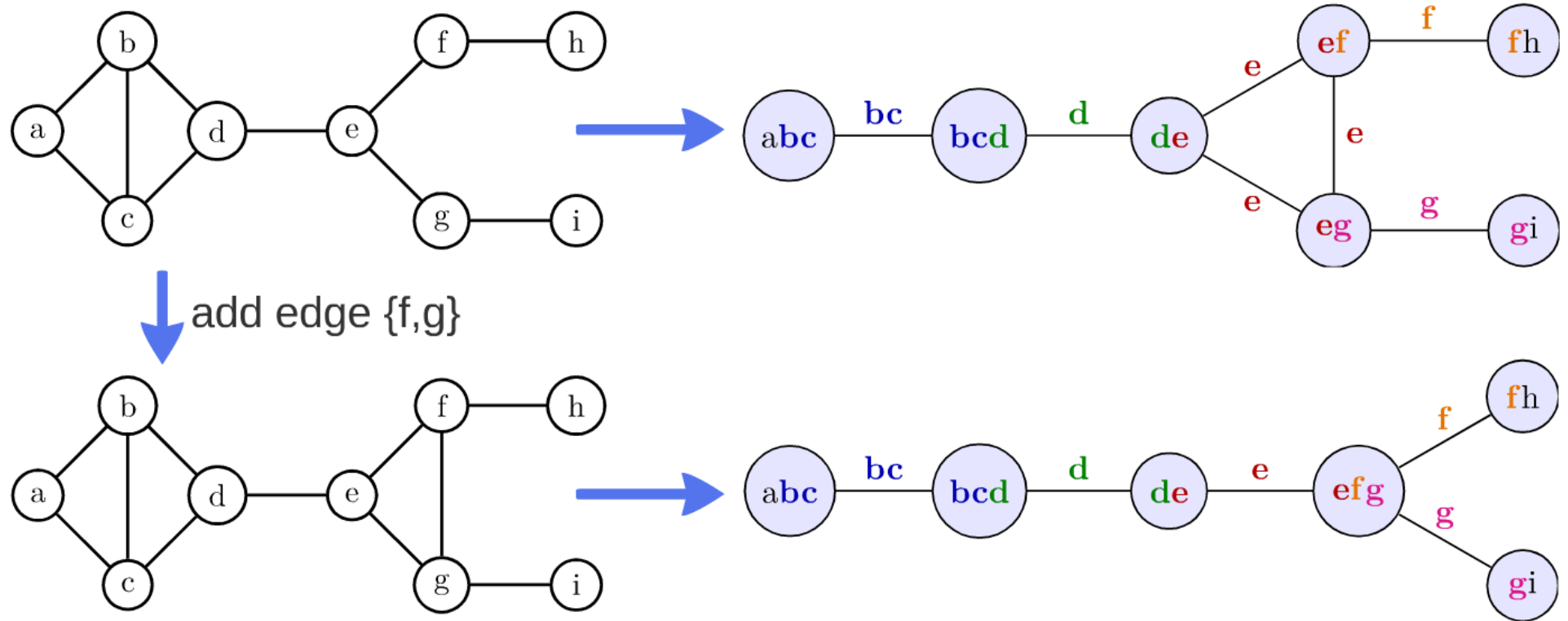
2. Keep track of the minimal separators for every potential edge



3. Maintain an ordered list of all the potential edges (priority queue)



Clique graph

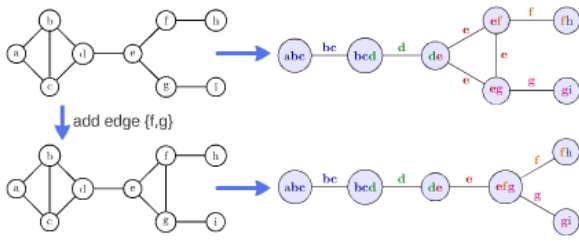


There are algorithms that can directly update the clique-graph [1,2]

[1] A. Deshpande *et al.*, "Efficient stepwise selection in decomposable models," in *UAI 2001*.

[2] F. Petitjean *et al.*, "Scaling log-linear analysis to datasets with thousands of variables" In *SDM 2015*.

Clique graph



There are algorithms that can directly update the clique-graph [1,2]

[1] A. Deshpande et al., "Efficient stepwise selection in decomposable models," in UAI 2001.
[2] F. Petitjean et al., "Scaling log-linear analysis to datasets with thousands of variables" In SDM 2015.

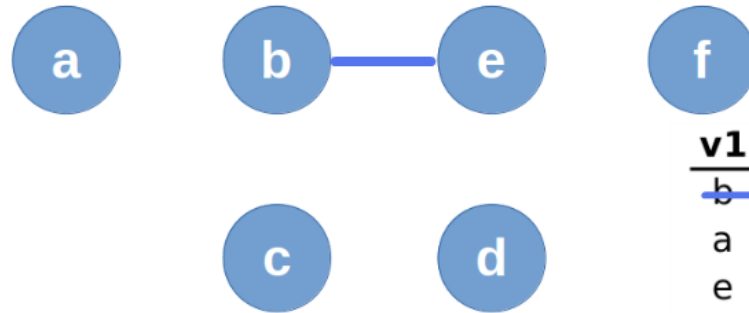
n^2



$n \cdot \log n$

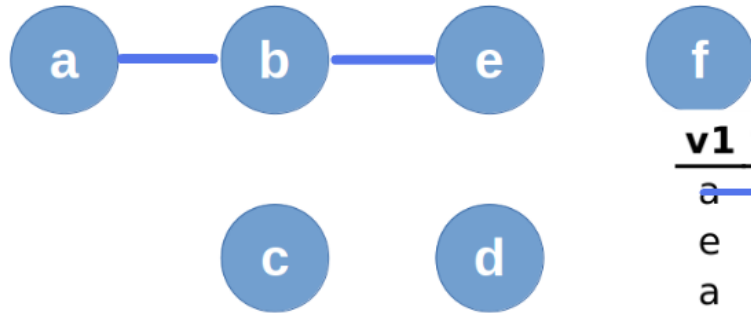


Priority queue



Add b-e

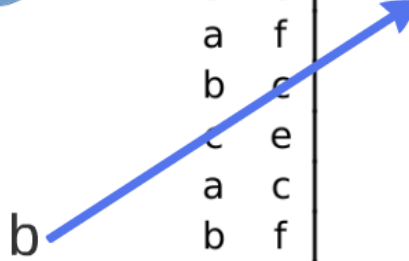
v1	v2	separator	score
b	e	{ }	96.2
a	b	{ }	72.8
e	f	{ }	60.9
a	e	{ }	49.5
a	f	{ }	42.8
b	c	{ }	31.4
c	e	{ }	31.0
a	c	{ }	28.8
b	f	{ }	17.1
c	d	{ }	16.9
b	c	{ }	12.7
c	f	{ }	8.1
d	e	{ }	7.3
d	f	{ }	4.8
e	f	{ }	4.6



v1	v2	separator	score
a	b	{ }	72.8
e	f	{ }	60.9
a	e	{ }	49.5
a	f	{ }	42.8
b	e	{ }	31.4
c	e	{ }	31.0
a	c	{ }	28.8
b	f	{ }	17.1
c	d	{ }	16.9
b	c	{ }	12.7
c	f	{ }	8.1
d	e	{ }	7.3
d	f	{ }	4.8
e	f	{ }	4.6

Add a-b
 • update a-e

b



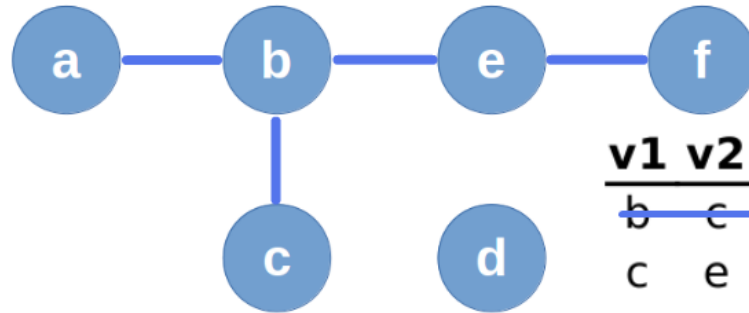


v1	v2	separator	score
e	f	{}	60.9
a	f	{}	42.8
b	c	{}	31.4
c	e	{}	31.0
a	c	{}	28.8
b	f	{}	17.1
c	d	{}	16.9
b	c	{}	12.7
a	e	{b}	12.4
c	f	{}	8.1
d	e	{}	7.3
d	f	{}	4.8
e	f	{}	4.6

Add e-f

- update b-f
- disable a-f

e

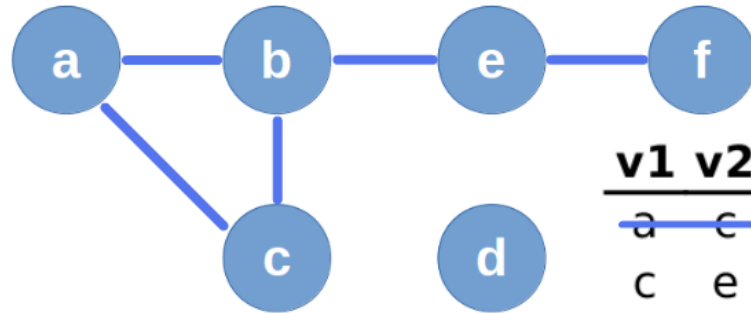


v1	v2	separator	score
b	c	{}	31.4
c	e	{}	31.0
a	c	{}	28.8
c	d	{}	16.9
b	f	{e}	14.0
b	c	{}	12.7
a	e	{b}	12.4
c	f	{}	8.1
d	e	{}	7.3
d	f	{}	4.8
e	f	{}	4.6

Add b-c

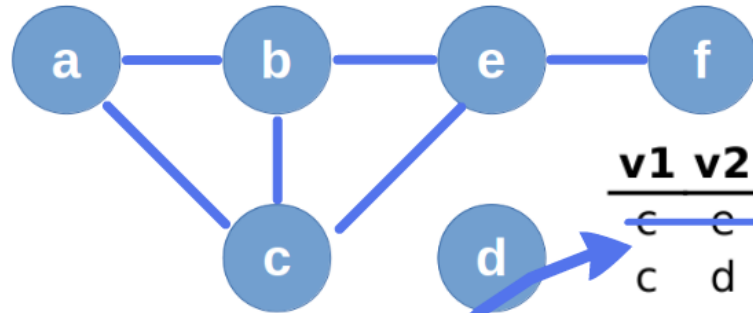
- update a-c
- update c-e
- disable c-f

b



Add a-c

v1	v2	separator	score
a	c	{b}	84.5
c	e	{b}	24.2
c	d	{}	16.9
b	f	{e}	14.0
b	c	{}	12.7
a	e	{b}	12.4
d	e	{}	7.3
d	f	{}	4.8
e	f	{}	4.6



c f | {e} | 49.4

v1	v2	separator	score
c	e	{b}	24.2
c	d	{}	16.9
b	f	{e}	14.0
b	c	{}	12.7
a	e	{b}	12.4
d	e	{}	7.3
d	f	{}	4.8
e	f	{}	4.6

C

Add c-e

- update a-e
- enable c-f

How fast can we get?



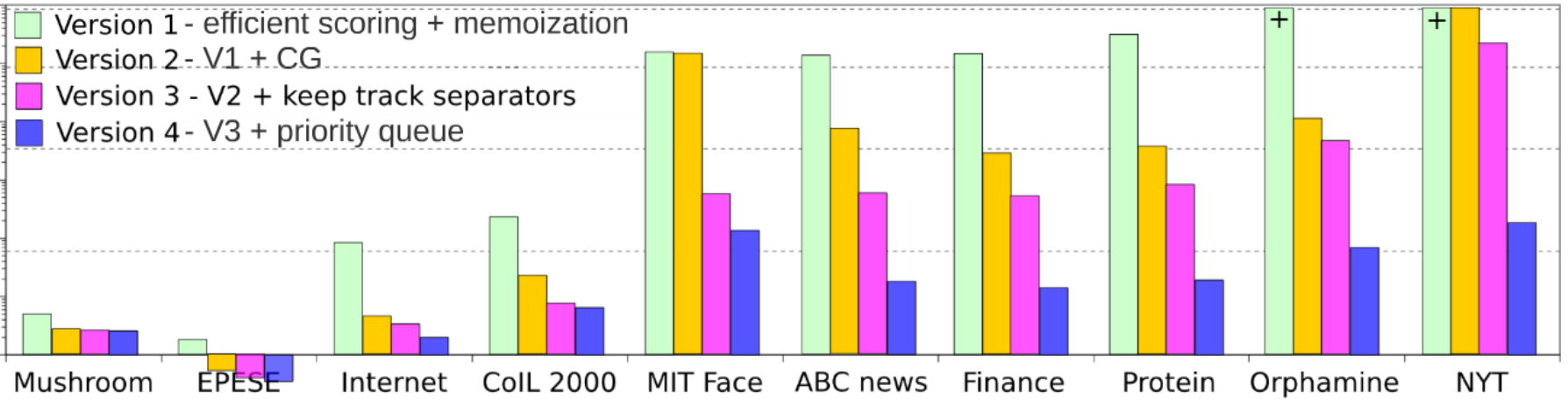
10 days

1 day

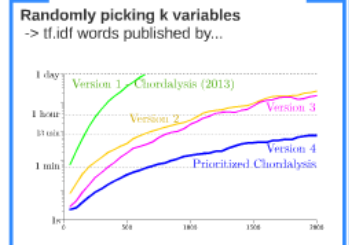
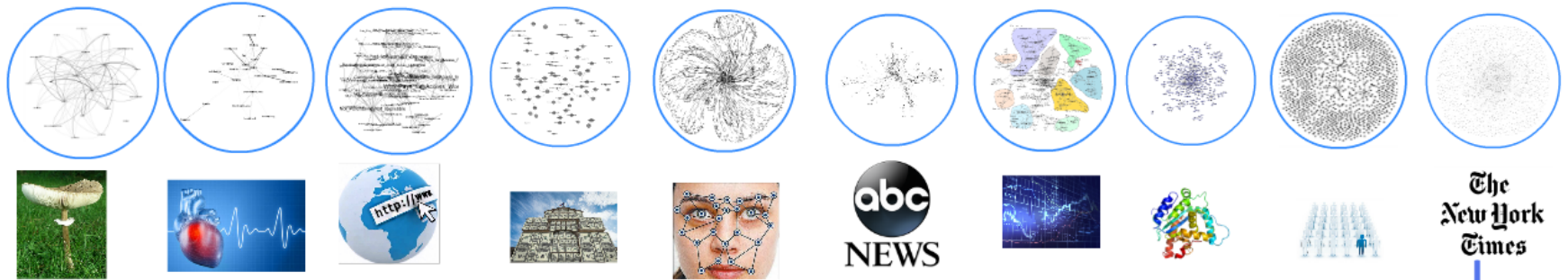
1 hour

1 min

1s



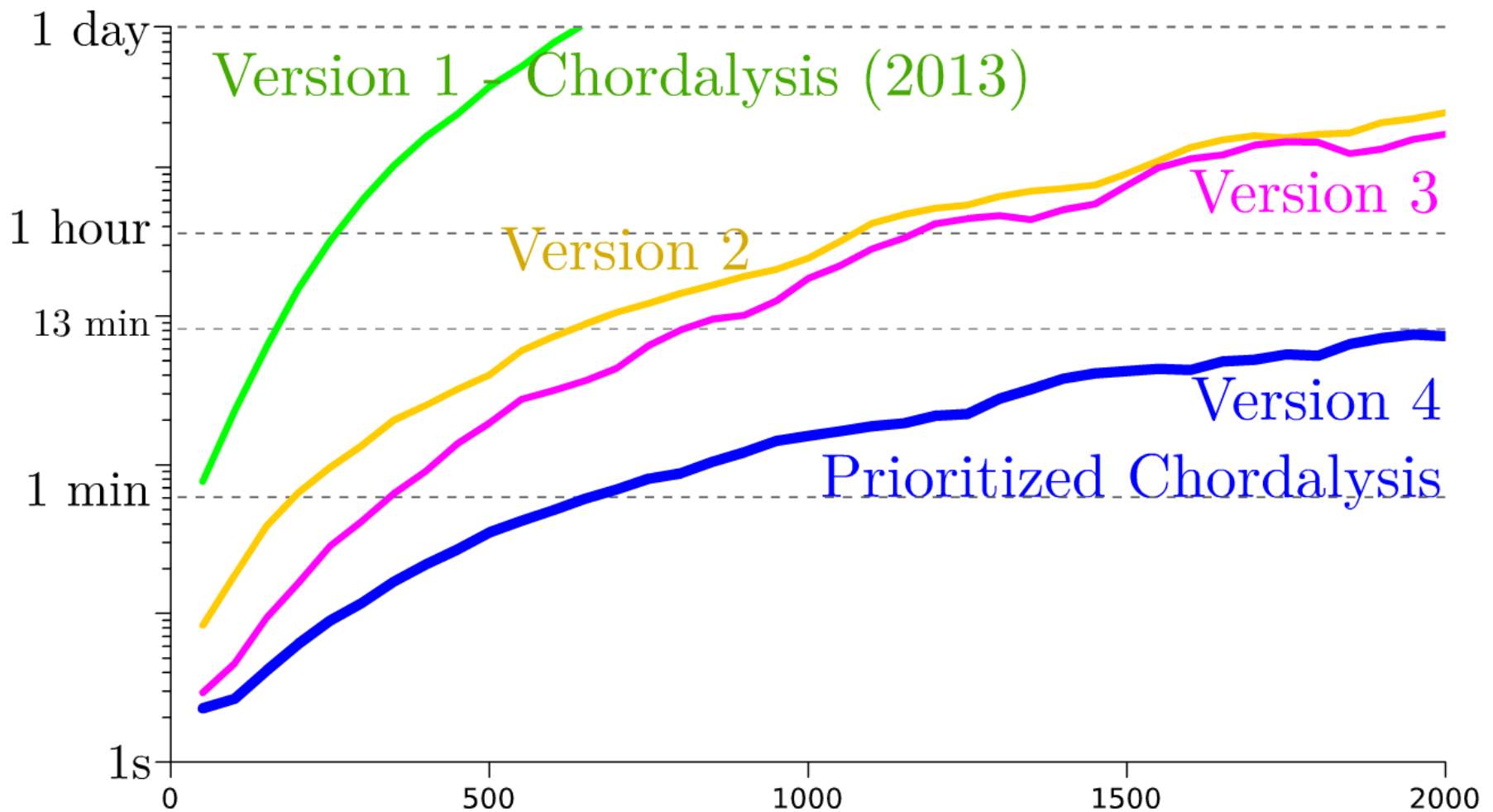
#Vars 20 25 70 85 300 500 500 700 1,200 2,000



Randomly picking k variables

-> tf.idf words published by...

The
New York
Times



Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model? Probabilistic graphical models (PGMs) are a compact representation of probability distributions. They are used for: Quantifying uncertainty, Not a trade-off in classical AI, and Aids Complexity (representing probability distributions).

What are graphical models useful for? They are used for: Image classification, Medical diagnosis, and Robotics. They are the thousands of applications of these methods.

What we will and will not cover: We will cover: Bayesian networks, Markov random fields, and Variational methods. We will not cover: Deep learning, Reinforcement learning, and Game theory.

Graphical models 101

Classes of graphical models: Directed Acyclic Graphs (DAGs) and Undirected Graphs (UGs).

A simple example of structure learning: Hill-climbing search on MRF using AIC. Example: Image classification.

Graph theory

Maximal cliques and minimal separators: Definitions and examples.

What are decomposable models: Decomposable models are a Markov Random Field for which the graph is chordal. Example: Bayesian network.

Properties of decomposable models: 1. Closed form for Z , 2. No \log operations, 3. Junction tree algorithm, 4. No \log operations, 5. Linear-time algorithm, 6. Interaction between BN and MRF.

Evaluation - Scoring

Decomposable models are essential for scalability, because... We need: 1. scalable scoring, 2. efficient search, 3. scalable belief propagation.

Bottom line: A score decomposable models is essential to: 1. AIC for MRFs, 2. A set of operations (Bayesian networks).

Most scores are scalable: Entropy [1], Mutual Information [2], Precision + recall, G-test statistic [3], Max. F1 score [4].

Break

Efficient search

Scoring in greedy search: In this case, we only need: Scoring of edge (0,1) to node 12,2. Data.

Clique graph (CG): Definition and properties.

Clique graph and greedy search: How to use the clique graph for scoring.

Search and statistical paradigm: Comparison of different search methods.

The nitty-gritty

Counting efficiently: Scoring for example with KL minimized when... What does it mean to compute $D(X||Y)$?

Counting efficiently (2): Many different ways to compute the same thing... What does it mean to compute $D(X||Y)$?

Memorization: How to store and retrieve information efficiently.

Addition of the same edge to different reference models: How to combine different models.

Use cases

Study of the elderly: 25 variables, 15,000 patients.

Insurance customer management: 93 variables, 6,000 customers.

Portfolio management: 500 variables, 20 years of trading.

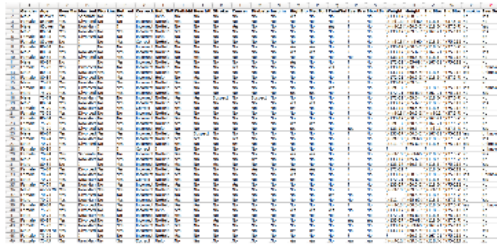
Wrapping up!

This tutorial in a nutshell: 1. Graphical models are everywhere, 2. Graphical models are everywhere, 3. Graphical models are everywhere, 4. Graphical models are everywhere.

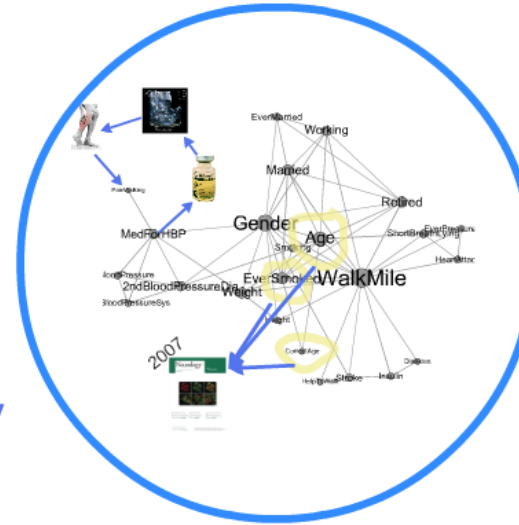
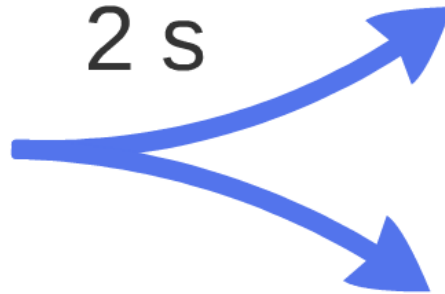
Open problems: 1. Efficient constrained search, 2. Better scores (eg on Directed acyclic graphs), 3. Efficient scoring of marginal "bits", 4. Efficient data structures for counting, 5. Leaving out of core, 6. Latent variables.

Study of the elderly

- 25 variables
- 15,000 patients



2 s



Belief propagation

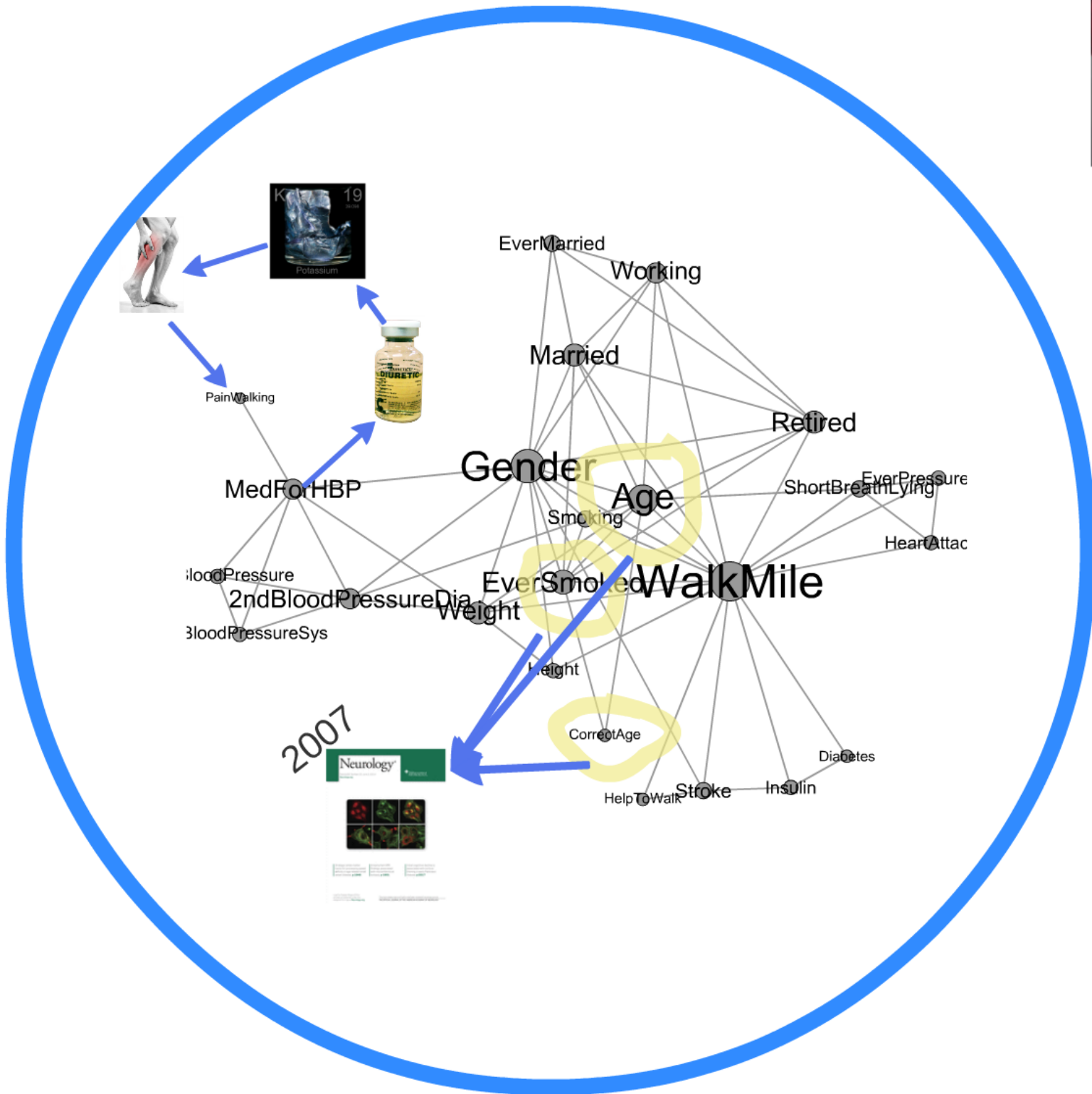
New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.



evidence	stroke	diabetes	heart attack
female under 70	5%	15%	10%
+ married	5%	15%	9%
+ smoking	7%	17%	12%
+ BP=17/10	8%	17%	13%
+ no help to walk	5%	16%	12%
+ quit smoking?	4%	14%	9%



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Gender	Age	EverMar	Married	Working	Retired	Correct	HelpToV	WalkMil	HeartAt	Stroke	Cancer	Diabete	Insulin	HighBlo	MedFor	PainWa	EverPre	ShortBr	Weight	Height	2ndBLoc	2ndBLoc	Smokin	EverSmo
2	Male	85over	Yes	Separate	No	Yes	?	Help	No	No	No	No	No	No	Yes	No	No	?	No	\(133-17	\(60.5-65	?	?	No	Yes
3	Female	85over	Yes	Divorced	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(37.5-75	No	No
4	Male	85over	Yes	NowMarr	?	?	Incorrect	NoHelp	No	No	No	Yes	No	No	No	No	Yes	?	No	\(-inf-13:	\(60.5-65	\(118.5-1	\(37.5-75	No	Yes
5	Male	80-84	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(133-17	\(69.5-in	\(167-21	\(75-112	No	Yes
6	Female	80-84	Yes	Divorced	No	No	Incorrect	Help	No	No	No	No	No	No	No	No	No	?	No	?	?	?	No	No	
7	Female	85over	No	?	No	Yes	Correct	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(75-112	No	No
8	Female	80-84	No	?	No	No	Incorrect	NoHelp	No	No	No	No	No	No	Yes	Yes	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(37.5-75	No	No
9	Male	80-84	Yes	NowMarr	No	Yes	Incorrect	NoHelp	No	No	No	No	No	No	Yes	Yes	No	?	No	\(133-17	\(65-69.5	\(167-21	\(75-112	No	Yes
10	Female	80-84	Yes	Divorced	No	Yes	Incorrect	NoHelp	No	Yes	No	No	No	No	No	No	No	No	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
11	Male	80-84	No	?	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(172-21	\(65-69.5	\(167-21	\(75-112	No	No
12	Female	75-79	Yes	Divorced	No	No	Incorrect	NoHelp	No	No	No	Yes	No	No	No	No	Yes	?	No	\(133-17	\(60.5-65	?	?	Yes	Unknown
13	Male	80-84	Yes	NowMarr	Yes	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	Yes	Unknown
14	Female	80-84	Yes	Divorced	No	Yes	Incorrect	Help	No	Yes	No	No	No	No	No	No	No	?	No	\(172-21	\(60.5-65	\(118.5-1	\(37.5-75	No	No
15	Male	75-79	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	Yes	No	No	No	No	No	No	No	?	No	\(133-17	\(69.5-in	?	?	No	No
16	Female	80-84	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	Yes	?	No	\(-inf-13:	?	\(118.5-1	\(75-112	Yes	Unknown
17	Male	75-79	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(172-21	\(65-69.5	\(167-21	\(75-112	No	Yes
18	Male	75-79	Yes	NowMarr	Yes	No	Incorrect	NoHelp	Yes	Yes	Suspect	No	Suspect	No	No	No	No	?	No	\(172-21	\(69.5-in	\(118.5-1	\(37.5-75	No	No
19	Male	80-84	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(133-17	\(65-69.5	\(118.5-1	\(75-112	No	Yes
20	Female	75-79	Yes	NowMarr	No	Yes	Correct	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
21	Female	75-79	Yes	Divorced	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(60.5-65	\(-inf-111:	\(37.5-75	No	No
22	Female	80-84	Yes	NowMarr	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(167-21	\(37.5-75	No	No
23	Female	75-79	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(-inf-13:	\(60.5-65	\(167-21	\(37.5-75	No	No
24	Male	75-79	Yes	Divorced	No	Yes	Incorrect	Help	No	No	No	No	No	No	Yes	Yes	No	Yes	No	\(133-17	\(69.5-in	\(-inf-111:	\(37.5-75	No	Yes
25	Male	80-84	Yes	Divorced	Yes	Yes	Incorrect	NoHelp	Yes	Suspect	No	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(167-21	\(75-112	Yes	Unknown
26	Female	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	Yes	No	Yes	No	No	No	No	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(37.5-75	No	Yes
27	Male	70-74	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	Yes	No	No	No	No	No	?	No	\(211-inf	\(65-69.5	\(118.5-1	\(75-112	Yes	Unknown
28	Female	80-84	?	?	?	?	Correct	?	No	No	No	No	No	No	No	No	No	?	No	?	?	?	?	?	Unknown
29	Female	70-74	Yes	Separate	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	Yes	No	?	No	\(-inf-13:	\(-inf-60.	\(118.5-1	\(37.5-75	No	Yes
30	Male	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	Yes	No	No	No	No	?	No	\(133-17	\(65-69.5	\(118.5-1	\(37.5-75	No	No
31	Male	80-84	No	?	No	Yes	Incorrect	NoHelp	Yes	No	Yes	No	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	Yes
32	Female	70-74	Yes	Divorced	Yes	Yes	Incorrect	NoHelp	Yes	No	No	No	Yes	No	Yes	Yes	No	?	No	\(172-21	?	\(118.5-1	\(75-112	No	No
33	Female	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	Yes	Unknown
34	Male	70-74	Yes	NowMarr	No	Yes	Correct	NoHelp	Yes	No	No	No	No	No	Yes	No	No	?	No	\(172-21	\(65-69.5	\(118.5-1	\(37.5-75	No	Yes
35	Female	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	Yes	No	No	No	No	No	?	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
36	Male	under70	Yes	NowMarr	Yes	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(69.5-in	\(-inf-111:	\(37.5-75	No	No
37	Female	70-74	Yes	Divorced	No	No	Incorrect	NoHelp	No	No	No	No	No	No	No	No	Yes	No	No	?	\(60.5-65	\(118.5-1	\(37.5-75	No	No
38	Male	70-74	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(211-inf	\(65-69.5	\(118.5-1	\(37.5-75	No	Yes
39	Female	80-84	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	Yes	No	No	No	No	No	No	No	?	No	\(-inf-13:	\(60.5-65	\(118.5-1	\(37.5-75	Yes	Unknown
40	Female	75-79	Yes	Divorced	No	No	Incorrect	Help	Yes	No	No	No	No	No	Yes	Yes	Yes	No	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	No
41	Male	70-74	Yes	NowMarr	Yes	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	\(172-21	\(65-69.5	\(118.5-1	\(75-112	No	No
42	Female	80-84	Yes	Divorced	No	No	Incorrect	NoHelp	No	No	Yes	No	No	No	No	No	No	Yes	Yes	\(133-17	\(65-69.5	\(118.5-1	\(37.5-75	No	No
43	Female	75-79	Yes	NowMarr	No	Yes	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	No	No	\(133-17	\(-inf-60.	\(167-21	\(75-112	No	No
44	Male	under70	Yes	Divorced	No	Yes	Incorrect	NoHelp	Yes	No	No	Yes	No	Yes	Yes	Yes	No	No	No	\(133-17	\(60.5-65	\(118.5-1	\(75-112	No	Yes
45	Female	75-79	No	?	No	No	Incorrect	NoHelp	Yes	No	No	No	No	No	No	No	No	?	No	?	\(60.5-65	\(167-21	\(75-112	No	No
46	Female	75-79	Yes	NowMarr	No	Yes	Correct	Help	No	Yes	No	No	No	No	Yes	Yes	No	Yes	Yes	\(-inf-13:	\(60.5-65	\(118.5-1	\(75-112	No	No



Belief propagation

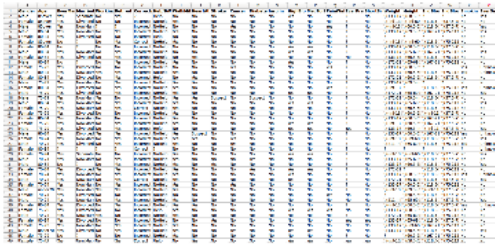
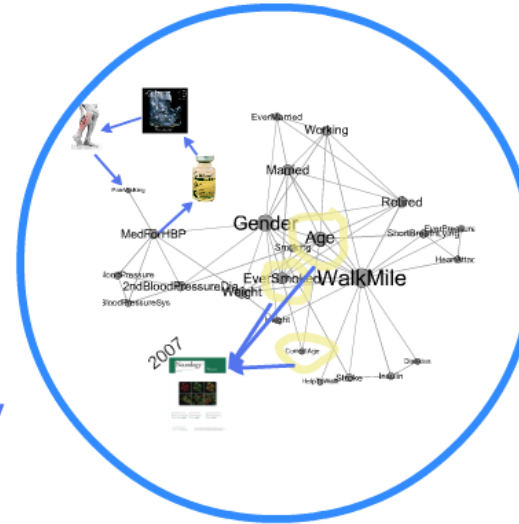
New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.



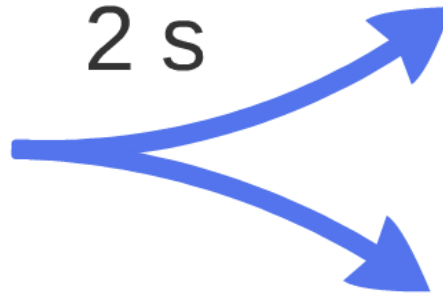
evidence	stroke	diabetes	heart attack
<i>female under 70</i>	5%	15%	10%
<i>+ married</i>	5%	15%	9%
<i>+ smoking</i>	7%	17%	12%
<i>+ BP=17/10</i>	8%	17%	13%
<i>+ no help to walk</i>	5%	16%	12%
<i>+ quit smoking?</i>	4%	14%	9%

Study of the elderly

- 25 variables
- 15,000 patients



2 s



Belief propagation

New patient, Lan, is visiting her new GP; the GP wants to check her risk of getting a few diseases: stroke, diabetes, heart attack.



evidence	stroke	diabetes	heart attack
female under 70	5%	15%	10%
+ married	5%	15%	9%
+ smoking	7%	17%	12%
+ BP=17/10	8%	17%	13%
+ no help to walk	5%	16%	12%
+ quit smoking?	4%	14%	9%

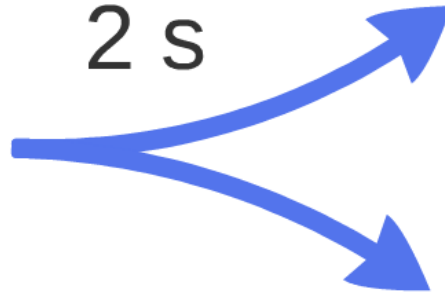


Insurance customer management

- 80 variables
- 6,000 customers

A large, dense grid of data points, likely representing a dataset with 80 variables and 6,000 customers. The grid is composed of many small, light-colored squares arranged in a regular pattern.

2 s



Belief propagation

New customer, Mat, is visiting his new branch; the customer representative takes the opportunity to check potential for new insurance policies.



evidence	fire	van	life

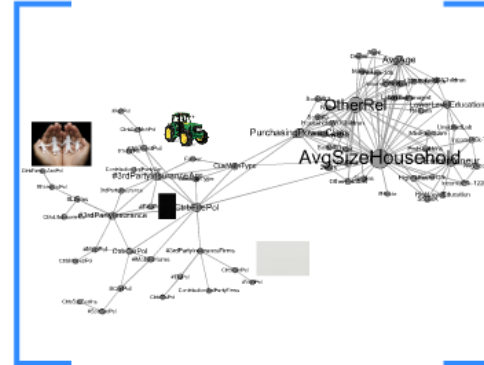
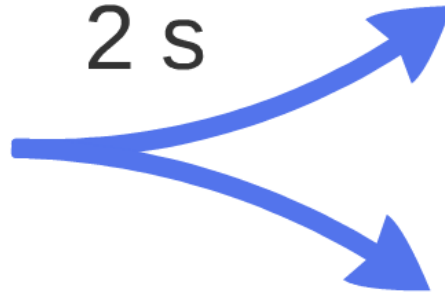
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
Customer	Number of	Avg size household	Avg age	Customer main type	Roman catholic	Protestant	Other religior	No religior	Married	Living togethe	Other rela	Singles	Household without childre	Household with childre	High level edu
33	1	3	2	8	0	5	1	3	7	0	2	1	2	6	6
37	1	2	2	8	1	4	1	4	6	2	2	0	4	5	5
37	1	2	2	8	0	4	2	4	3	2	4	4	4	2	2
9	1	3	3	3	2	3	2	4	5	2	2	2	3	4	4
40	1	4	2	10	1	4	1	4	7	1	2	2	4	4	4
23	1	2	1	5	0	5	0	5	0	6	3	3	5	2	2
39	2	3	2	9	2	2	0	5	7	2	0	0	3	6	6
33	1	2	3	8	0	7	0	2	7	2	0	0	5	4	4
33	1	2	4	8	0	1	3	6	6	0	3	3	3	3	3
11	2	3	3	3	3	5	0	2	7	0	2	2	2	6	6
10	1	4	3	3	1	4	1	4	7	1	2	0	3	6	6
9	1	3	3	3	1	3	2	4	7	1	2	2	3	5	5
33	1	2	3	8	1	4	1	4	6	2	3	3	4	3	3
41	1	3	3	10	0	5	0	4	7	1	1	1	4	5	5
23	1	1	2	5	0	6	1	2	1	2	6	5	3	1	1
33	1	2	3	8	0	7	0	2	7	2	0	0	5	4	4
38	1	2	3	9	0	6	0	3	7	0	2	0	6	3	3
22	2	3	3	5	0	5	0	4	7	0	2	0	2	7	7
13	1	4	2	3	2	4	0	3	7	0	2	1	3	6	6
31	1	2	4	7	0	2	0	7	9	0	0	0	6	3	3
33	1	4	3	8	0	6	0	3	9	0	0	0	3	6	6
33	2	3	3	8	0	4	2	3	7	0	2	0	2	7	7
13	1	3	2	3	1	7	0	2	7	0	2	1	3	6	6
34	2	3	2	8	0	7	0	2	7	2	0	0	4	5	5
13	2	4	3	3	0	4	2	4	8	1	1	1	3	6	6
33	1	3	3	8	0	6	1	2	6	0	3	2	3	5	5
37	1	3	3	8	0	5	0	4	7	2	0	0	3	6	6
40	1	3	3	10	0	3	0	6	9	0	0	0	4	5	5
31	1	4	2	7	0	9	0	0	5	0	4	0	0	9	9
33	2	2	3	8	0	7	1	2	5	1	4	4	1	5	5
24	2	2	2	5	1	3	2	4	2	4	3	3	3	3	3
23	1	2	2	5	2	4	2	4	7	1	2	6	0	3	3
33	1	4	3	8	1	4	1	5	8	1	1	1	2	7	7
38	1	2	3	9	0	5	2	2	4	2	4	4	4	3	3
13	2	4	3	3	0	4	2	4	8	1	1	1	3	6	6
8	1	3	2	2	2	4	1	4	6	1	3	1	4	5	5
7	1	3	2	2	0	7	2	0	7	2	0	0	6	3	3
41	1	3	3	10	0	7	1	2	8	1	1	1	5	3	3
39	1	3	2	9	0	6	0	3	6	0	3	0	0	9	9
33	2	3	3	8	0	2	3	5	6	3	0	0	3	6	6
24	1	3	3	5	1	5	1	3	6	1	2	0	0	9	9
11	1	3	3	3	2	7	0	0	9	0	0	2	3	4	4
8	1	3	3	2	1	4	1	4	6	1	2	2	3	5	5
33	1	2	4	8	0	5	0	4	8	0	1	1	7	2	2

Insurance customer management

- 80 variables
- 6,000 customers

A screenshot of a data table with approximately 80 columns and 6,000 rows, representing the raw data for insurance customer management.

2 s



Belief propagation

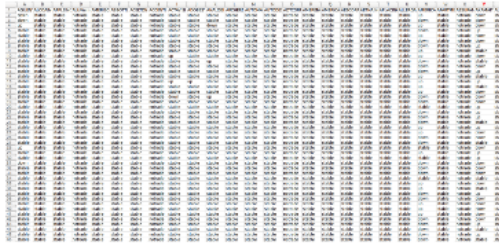
New customer, Mat, is visiting his new branch; the customer representative takes the opportunity to check potential for new insurance policies.



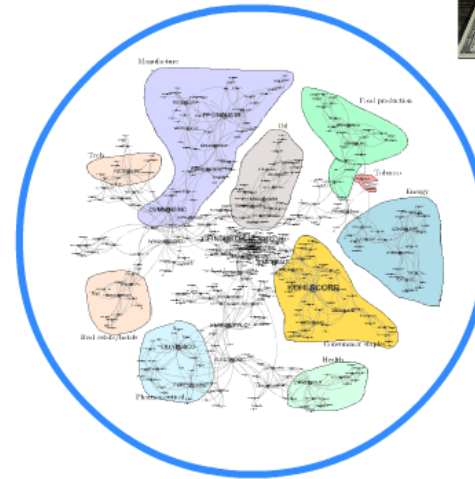
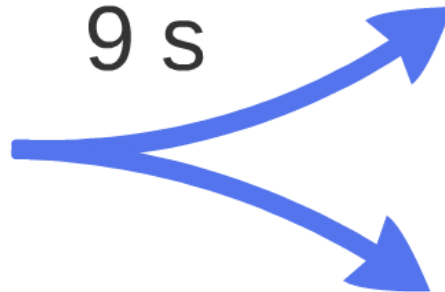
evidence	fire	van	life

Portfolio management

- 500 variables
- 20 years of trading



9 s



Belief propagation

Financial adviser wants to see how the market might behave given a few speculations over stocks.

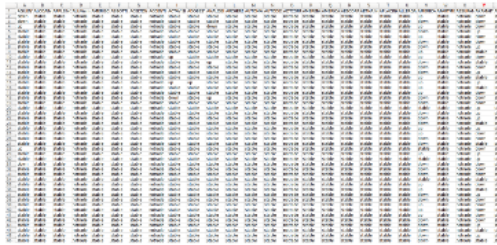


evidence		NETFLIX	MERCK	RALPH LAUREN	
prior		19%-19%	23%-22%	12%-13%	23%-25%
Netflix	↑	21%-21%	27%-30%	13%-13%	—
J&J	↓	21%-20%	—	34%-15%	24%-25%
Apple	↓	26%-20%	34%-26%	—	23%-25%
Amazon	↑	—	—	—	25%-37%

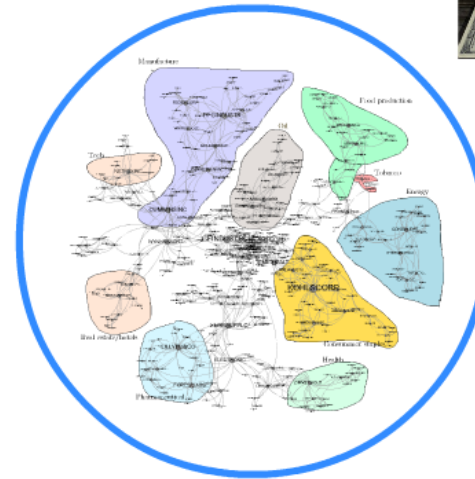
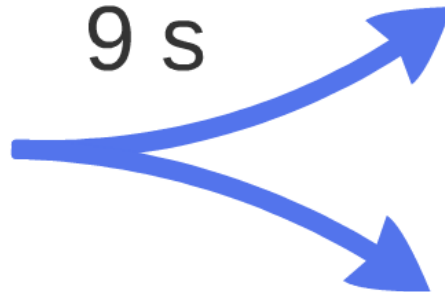
- Netflix needs drives?
- Merck and J&J are in the same cluster
- <http://www.buyupside.com> says AMAZN and RL have 0.99 correlation coefficient
- External factor? Sales of Ralph Lauren on Amazon.com?

Portfolio management

- 500 variables
- 20 years of trading



9 s



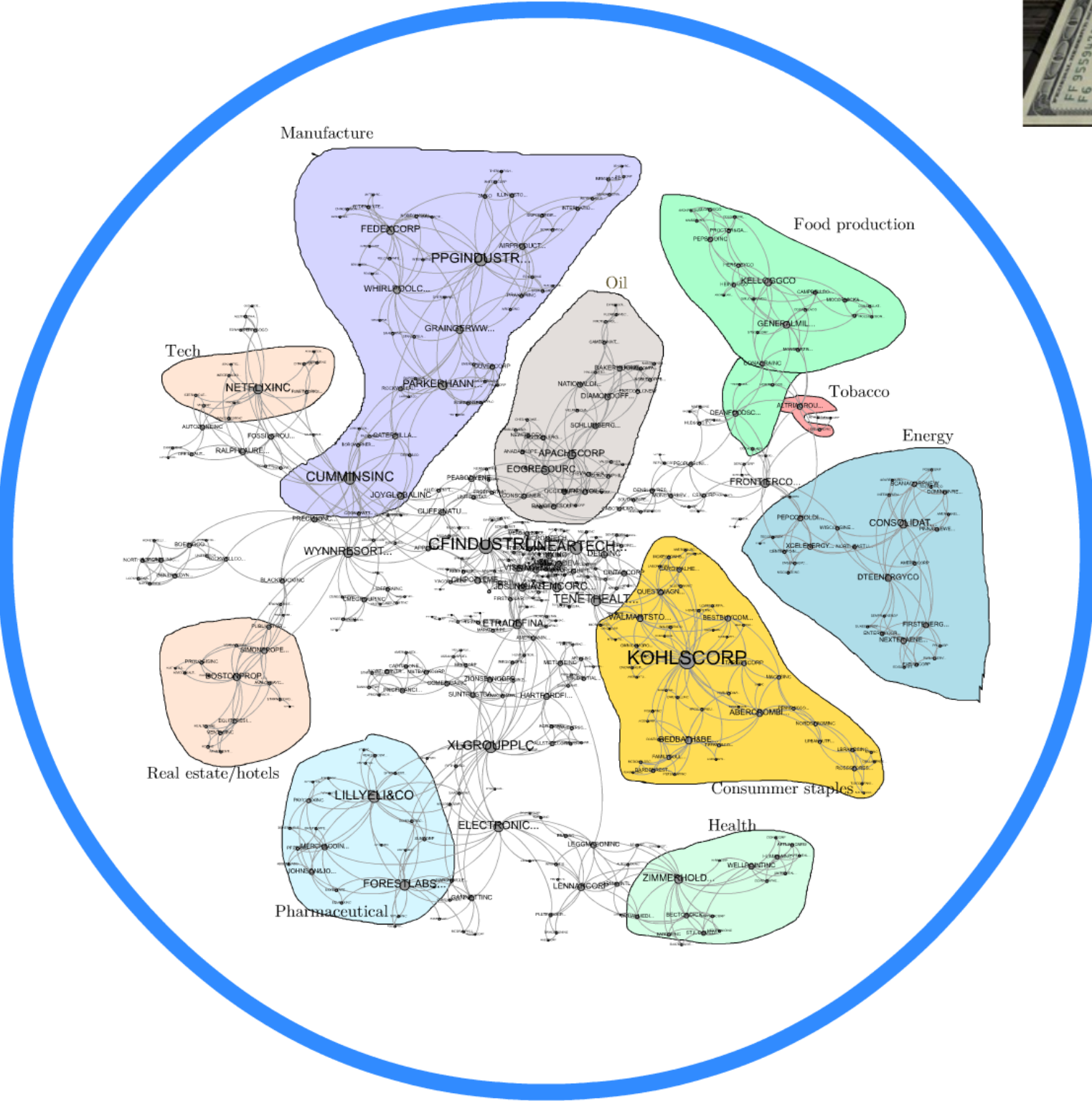
Belief propagation

Financial adviser wants to see how the market might behave given a few speculations over stocks.



evidence		NETFLIX	MERCK	RALPH LAUREN
prior	19%-19%	23%-22%	12%-13%	23%-25%
Netflix	21%-21%	27%-30%	13%-13%	—
J&J	21%-20%	—	34%-15%	24%-25%
Apple	26%-20%	34%-26%	—	23%-25%
Amazon	—	—	—	25%-37%




















- Netflix needs drives?
- Merck and J&J are in the same cluster
- <http://www.buyupside.com> says AMAZN and RL have 0.99 correlation coefficient
- External factor? Sales of Ralph Lauren on Amazon.com?



Belief propagation

Financial adviser wants to see how the market might behave given a few speculations over stocks.



evidence				
	 	 	 	 
prior	19%-19%	23%-22%	12%-13%	23%-25%
Seagate  	21%-21%	27%-30%	13%-13%	—
Johnson & Johnson 	21%-20%	—	34%-15%	24%-25%
Apple  	26%-20%	34%-26%	—	23%-25%
Amazon  	—	—	—	25%-37%

- Netflix needs drives?
- Merck and J&J are in the same cluster
- <http://www.buyupside.com/> says AMAZN and RL have 0.89 correlation coefficient
 - External factor? Sales of Ralph Lauren on Amazon.com?

Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probability theory + Graph Theory
 Probability
 Quantifying uncertainty
 Not a trade-off in classical agent systems
 Aids Complexity representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 In: Feature selection, Feature extraction, Feature engineering, Feature construction, Feature interaction, Feature fusion, Feature selection, Feature extraction, Feature engineering, Feature construction, Feature interaction, Feature fusion
 Out: Feature selection, Feature extraction, Feature engineering, Feature construction, Feature interaction, Feature fusion

Graphical models 101

Classes of graphical models
 Undirected graphical models (UGM)
 Directed graphical models (DGM)

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data
 Scoring
 Search

Graph theory

Maximal cliques and minimal separators
 Definition 1: A clique is a subset of nodes in a graph such that every two nodes in the subset are adjacent to each other.
 Definition 2: A minimal separator is a set of nodes that separates the graph into two components.

What are decomposable models
 Decomposable models are a special class of models for which the graph is chordal.

Properties of decomposable models
 1. Closed form for the partition function
 2. No global optimization
 3. Junction tree algorithm
 4. No global optimization
 5. Linear-time algorithm
 6. Interaction between IJ and MRF [2]

Useful algorithms
 Junction tree algorithm
 Variable elimination

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:
 1. scalable scoring
 2. efficient search
 3. scalable belief propagation
 = all the results we will show here

Bottom line
 A score decomposable model is essential to:
 - AIC for MRFs
 - A set of operations to speed up inference

Most scores are scalable
 Energy [1]
 Submodular L1 [2] Because it is submodular when energy is L1
 Global statistics [3]
 Max. FIM [4]

Break

Efficient search

Scoring in greedy search
 In this case, we only need:
 Scoring of edge (i,j) in G
 Data

Clique graph (CG)
 Definition 1: A clique graph is a graph in which the nodes are cliques of the original graph.
 Definition 2: A clique graph is a graph in which the nodes are cliques of the original graph.

Clique graph and greedy search
 We can extend the greedy search algorithm to the clique graph.

Search and statistical paradigm
 Search
 Statistical paradigm

The nitty-gritty

Counting efficiently
 Scoring for example with KL minimized when...
 What does it mean to compute $D(X||Y)$?
 Efficient counting built on efficient counting

Counting efficiently (2)
 Many different counting problems require counting the number of configurations of a graphical model.

Memorization
 From the high multiplicity of configurations to the low multiplicity of configurations.

Addition of the same edge to different reference models
 What we have seen so far
 Counting the addition of edges into graphs
 Corollary
 How often does that happen?
 How can we use this information?

How fast can we get?
 Bar chart showing performance metrics.

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell
 1. Graphical models are everywhere!
 2. We can learn graphical models from data with 1,000+ variables
 3. It is possible to do inference on graphs in the library that we are exploring in our selected class
 4. There is still so much work to be done!

Open problems
 1. Efficient constrained search
 2. Better scores (eg on Directed scoring on Ising)
 3. Efficient scoring of marginal "bits"
 4. Efficient data structures for counting
 5. Learning out of core
 6. Latent variables

Open problems (2)
 7. How to handle numerical variables
 8. How to handle missing values?
 9. Learning accurate parameters in large tables
 10. Many problems are non-hanging (that you just need to push them)

We hope that you enjoyed our tutorial on...
 Scalable learning of graphical models
 Feature selection and graph theory
 Feature selection and graph theory
 Feature selection and graph theory

This tutorial in a nutshell

1. Graphical models are extremely useful:
 - Extracting knowledge from data
 - Compact representation of high-order multivariate distributions
 - Making omnidirectional predictions



2. We can learn graphical models from datasets with 1,000+ variables <https://github.com/fpetitjean/Chordalysis/>

3. *Chordalysis* is the name we gave to the library that can do everything we have talked about

4. There is still so much work to be done!



Open problems

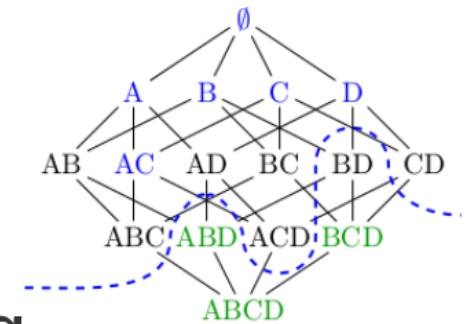
NOW IT'S YOUR TURN.

1. Efficient randomized search

→ Comparison with exact search when <25 variables?

2. Better scores (eg no Dirichlet scoring so far!)

3. Efficient storing of marginal "data"



4. Efficient data structures for counting

→ on large datasets, 99% of the CPU is used for counting

5. Learning out of core



6. Latent variables

Open problems (2)

Your community needs

7. How to handle numerical variables?

8. How to handle missing values?

9. Learning accurate parameters in large tables



Many problems are low-hanging fruit; you just need to pick them!

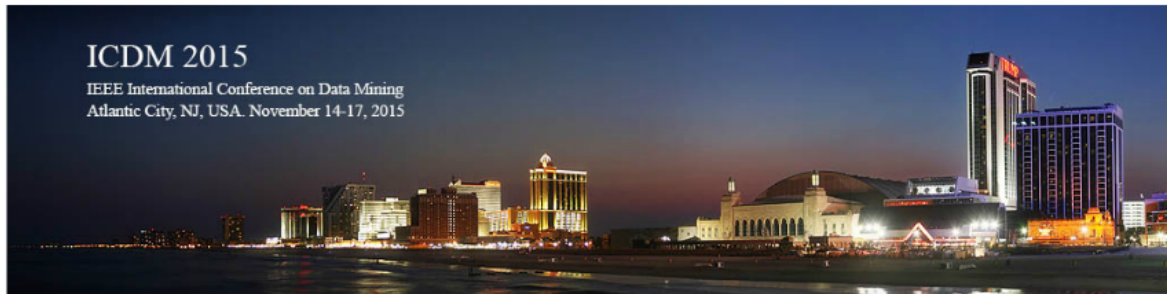
We hope that you enjoyed our tutorial on ...

Scalable learning of graphical models

François Petitjean and Geoff Webb



<http://www.francois-petitjean.com>



Scalable learning of graphical models

Introduction - Motivation

What is a probabilistic graphical model?
 Probability theory + Graph Theory
 probability
 Quantifying uncertainty
 Not a trade-off in classical agent systems
 Aids Complexity representing probability distributions

What are graphical models useful for?
 - the thousands of applications of these methods...

What we will and will not cover
 What we will cover: In, Out
 What we will not cover: Out

Graphical models 101

Classes of graphical models
 Undirected graphical models
 Directed graphical models

A simple example of structure learning
 Hill-climbing search on MRF using AIC

Learning a model from data
 Scoring
 Search

Graph theory

Maximal cliques and minimal separators
 Definition 1: A maximal clique is a set of nodes in a graph such that every two nodes in the set are adjacent, and no other node in the graph is adjacent to every node in the set.
 Definition 2: A minimal separator is a set of nodes in a graph such that every two nodes in the set are adjacent, and every node in the set is adjacent to at least one node in each of two disjoint maximal cliques.

What are decomposable models
 Decomposable models are a Markov Random Field for which the graph is chordal or triangulated.

Properties of decomposable models
 1. Closed form for the partition function
 2. No global optimization
 3. Junction tree algorithm
 4. No 2ⁿ global search
 5. Linear-time algorithm for inference
 6. Interaction between IJ and MRF [2]

Useful algorithms
 Junction tree algorithm
 Variable elimination

Evaluation - Scoring

Decomposable models are essential for scalability, because...
 ... we need:
 1. scalable scoring
 2. efficient search
 3. scalable belief propagation
 = all the results we will show here

Bottom line
 A score decomposable model is essential to:
 - AIC for MRFs
 - A set of operations to support inference
 - Any scoring function that has been identified for MRFs in the literature to be used for decomposable models
 - MRF-based applications
 - MRF-based graphical models that are not decomposable
 - The scale of n

Most scores are scalable
 Energy [1]
 Subtree Ladder [2] Because it is unrolled when energy is used
 Global tables [3]
 Max. FMS [4,5]

Break

Efficient search

Scoring in greedy search
 In this case, we only need:
 Scoring of edge (i,j) in G
 Data

Clique graph (CG)
 Definition 1: A clique graph is a graph in which the nodes are maximal cliques of a graph G, and two nodes are adjacent if and only if their corresponding cliques in G share at least one node.
 Definition 2: A clique graph is chordal if and only if the original graph G is chordal.
 Definition 3: A chordal graph is a graph in which every cycle of length greater than 3 has a chord.
 Definition 4: A chordal graph is a graph in which every cycle of length greater than 3 has a chord.
 Definition 5: A chordal graph is a graph in which every cycle of length greater than 3 has a chord.

Clique graph and greedy search
 We can reduce the search space of the clique graph to the search space of the original graph.

Search and statistical paradigm
 Search paradigm
 Statistical paradigm

The nitty-gritty

Counting efficiently
 Scoring for example with KL minimized when...
 What does it mean to compute $\sum_{X \in \mathcal{X}} P(X)$?
 We can count efficiently by using the junction tree algorithm.

Counting efficiently (2)
 Many different ways to count efficiently require making use of the junction tree algorithm.

Memorization
 From the high multiplicity of the nodes in the junction tree, we can reduce the search space of the clique graph to the search space of the original graph.

Addition of the same edge to different reference models
 What we have seen so far:
 Counting the addition of an edge into a graph
 Corollary:
 How often does that happen?
 How can we use this information?

How fast can we get?
 Comparison of different algorithms

Use cases

Study of the elderly
 - 25 variables
 - 15,000 patients

Insurance customer management
 - 93 variables
 - 6,000 customers

Portfolio management
 - 500 variables
 - 20 years of trading

Wrapping up!

This tutorial in a nutshell
 1. Graphical models are everywhere!
 2. We can learn graphical models from data with 1,000+ variables
 3. It is possible to do inference on graphs in the library that are exponentially larger than the data
 4. There is still so much work to be done!

Open problems
 1. Efficient constraint-based search
 2. Better scores (eg on Directed scoring on Ising)
 3. Efficient coding of marginal "bits"
 4. Efficient data structures for counting
 5. Learning out of core
 6. Latent variables

Open problems (2)
 7. How to handle numerical variables
 8. How to handle missing values?
 9. Learning accurate parameters in large tables
 10. Many problems are non-hanging (that you just need to push them)

We hope that you enjoyed our tutorial on...
 Scalable learning of graphical models
 From the lecture and your study
 Many thanks to the participants
 Many thanks to the participants