An exploration of methods for verification of probability models based on Bayesian networks.

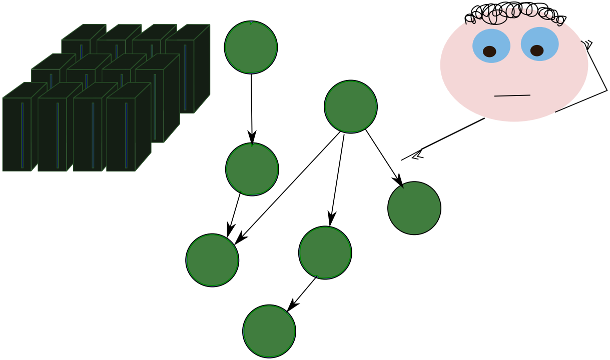
Pavel Novikov
Supervised by: Oleg Senko

Moscow, 2015

Table of contents

Bayesian networks

- Bayesian networks are good for encoding distributions.
- Problem: learning from scratch requires huge amounts of computational resources and doesn't guarantee good result.
- Alternative: expert knowledge (still no guarantee, though).



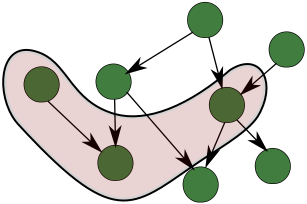
Verification

Problem:

- Learned network may be of low quality.
- Expert knowledge can be flawed.

Proposed verification procedure:

- Look at various marginal distributions.
- Use statistical testing to check if they fit to data.



Exploration procedure

1. Generate random Bayes net
2. Generate sample from this net
3. Extract subset distributions
4. Perform statistical testing

Repeat many times and examine false rejection portion.

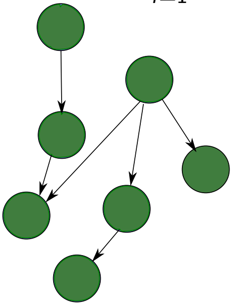
Table of contents

Where it all starts

$$P(X) = P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i | x_1, ..., x_{i-1}) =$$

$$= \prod_{i=1}^{n} P(x_i | Par(x_i))$$

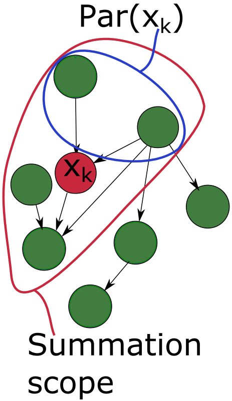
- Some scopes of each factor may be dummy.
- Graph is used to show which ones are.



Basic variable elimination method

We can simply sum variables out one by one.

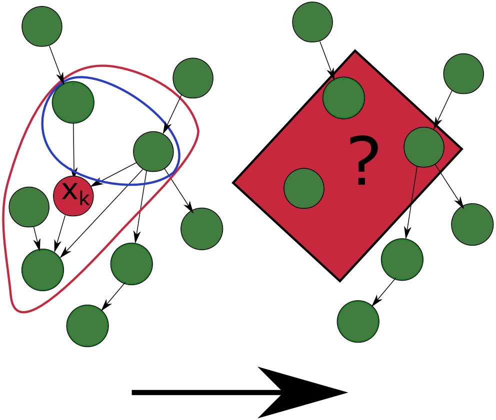
$$\sum_{x_k \in Dom(x_k)} \prod_i P(x_i|Par(x_i)) =$$
$$\left(\prod_{\{i:i\neq k, x_k \notin Par(x_i)\}} \left[P(x_i|Par(x_i)) \right] \right)$$
$$\left(\sum_{\{x_k \in Dom(x_k)\}} \left[P(x_k|Par(x_k)) \prod_{j:x_k \in Par(x_j)} (P(x_j|Par(x_j))) \right] \right)$$



Problems of variable elimination method

There are some drawbacks:

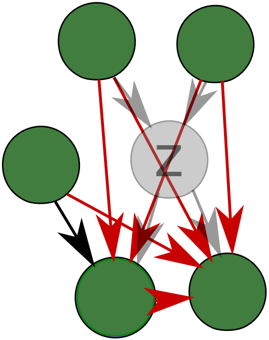
- it has exponential complexity
- we lose bayesian network



Recovering Bayesian network structure

Assertion

Distribution $P(X \setminus \{z\})$ factorizes over a graph G', produced from graph G by connecting every child c of removed vertex z with all vertices of the summation scope, preceding c in some fixed topological order.



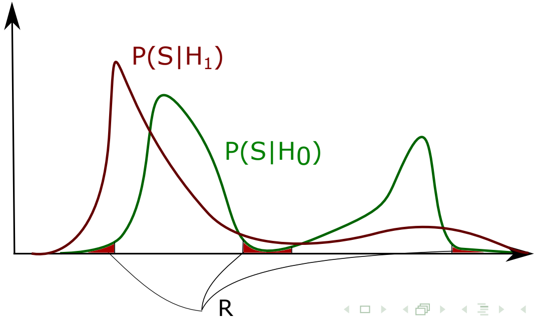
Algorithm for recomputing CPD

1. initialize $\mathcal{P}_0 \leftarrow P(z|Par(z))$, $i \leftarrow 1$
2. take next variable c in topological order from $Child(z)$
3. $\mathcal{P}_i = \mathcal{P}_{i-1} P(c|Par(c))$,
4. $P(x_c|\tilde{P}ar(x_c)) = \frac{\sum_z \mathcal{P}_i}{\sum_z \mathcal{P}_{i-1}}$.
5. $i \leftarrow i + 1$
6. if there are still variables in $Child(z)$ go to step 2;

Table of contents

The idea of hypothesis testing

H_0: zero hypothesis, *"default"*.
H_1: alternative.
S: statistic.
R: rejection region.
Idea: pick R so that $P(S \in R|H_0) < \alpha$(*significance level*).
p-value : minimal significance level that allows to reject particular hypothesis.



Pearson χ^2 and g-Test

P - probability given H_0
F - observed counts

	P	F
e_1	P_{e_1}	n_{e_1}
e_2	P_{e_2}	n_{e_2}
	...	
e_k	P_{e_k}	n_{e_k}
$\sum$	1	n

$$\chi^2 = \sum_{i=1}^{k} \frac{(n_{e_i} - nP_{e_i})^2}{nP_{e_i}}$$
$$G = 2\sum_{i=1}^{k} nP_{e_i} \log(\frac{nP_{e_i}}{n_{e_i}})$$

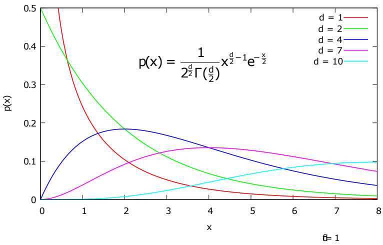
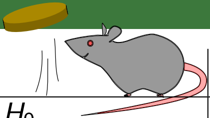


Figure: χ^2 *distribution*

Multiple hypothesis testing

Problem overview
Extracting subset distribution
Performing hypothesis testing
Generating random Bayesian net
Results



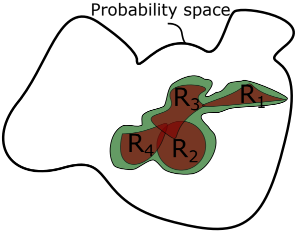
	Coin tossing	Food admixture
H_0	Coin is fair	Admixture affects mice
H_1	Coin is not fair	Admixture does not affect mice
Rejection	Equal results of all tosses	Specialized test procedure for some vital signs in 2 groups of mice
Procedure 1 OK	Throw N times.	Measure one type of vital signs
Procedure 2 Not OK	Make M people throw N times each	Measure M types

Multiple hypothesis testing really makes a difference!

Bonferroni correction

N hypotheses: $H_0^1, H_1^1; \ldots; H_0^N, H_1^N$
N significance levels: $\alpha^1; \ldots; \alpha^N$
N rejection regions: $R^1; \ldots; R^N$
N statistics: $S^1; \ldots; S^N$

$$P(\vee_{i=1}^{N}(S_i \in R_i) \leq \sum_{i=1}^{N} P((S_i \in R_i))$$



Idea: fix α_i to be equal α/N.

Stepwise correction procedures

- sort hypotheses by p-values p_k.
- compare p_k with $\frac{\alpha}{N+1-k}$
- reject all $H_0^i : i \leq r$, where r is:
- Holm step-down: $r = min(\{k : p_k > \frac{\alpha}{N+1-k}\}) - 1$
- Hochberg step-up: $r = max(\{k : p_k \leq \frac{\alpha}{N+1-k}\})$

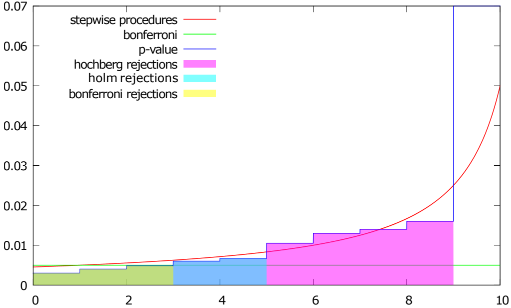


Figure: example of rejection procedures

Table of contents

Random graphs - Erdős–Rényi

Problem overview
Extracting subset distribution
Performing hypothesis testing
Generating random Bayesian net
Results

First formulation: each edge can be added to the graph independently of others with probability p.
Second formulation: random set of k edges is chosen uniformly at random.

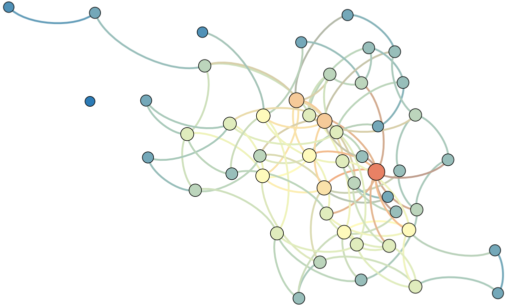


Figure: 50 vertices, 100 edges

Random graphs - Barabási–Albert

Problem overview
Extracting subset distribution
Performing hypothesis testing
Generating random Bayesian net
Results

Procedure starts with fully connected graph with n vertices.
Each new vertex added to the graph is connected to n old vertices.
Probability to chose a particular old vertex to connect to is proportional to its degree.

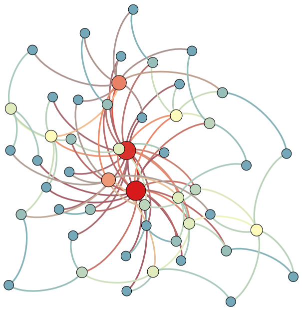


Figure: 50 vertices, 97 edges

Dirichlet and Beta distributions

Dirichlet distribution is a distribution over n-dimensional vectors of positive numbers that sum to one - tabular distributions.

$$Dir(x, \alpha) = \frac{1}{B(x)} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

2-dimensional case is referred to as Beta-distribution.

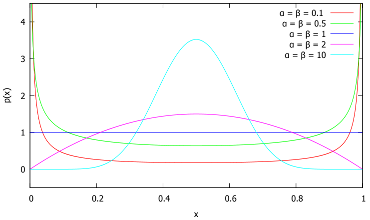
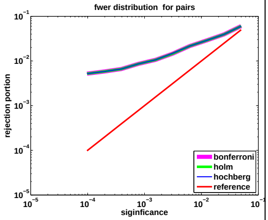
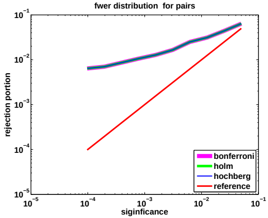
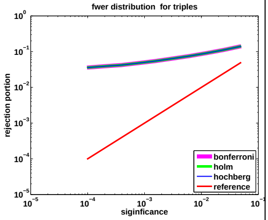
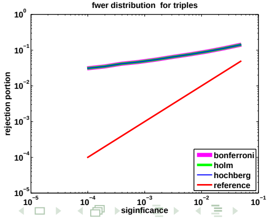


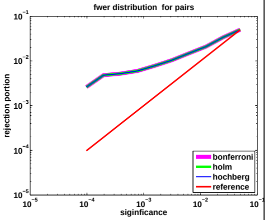
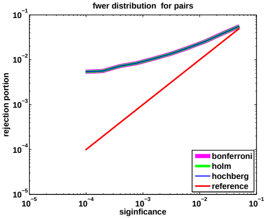
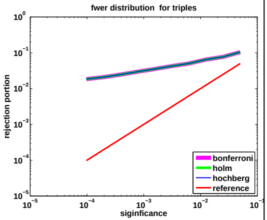
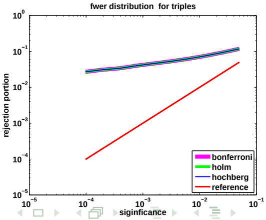
Figure: Beta distribution

Table of contents

χ^2, 10 vertices, 9 edges, $Dir(x, 1)$

	Erdős–Rényi	Barabási–Albert, direct order
Pairs		
Triples		

χ^2, 10 vertices, 17-18 edges, $Dir(x, 1)$

	Erdős–Rényi	Barabási–Albert, inverse order
Pairs		
Triples		

Simple hypothesis testing experiment - 100 000 experiments, 100 samples per experiment

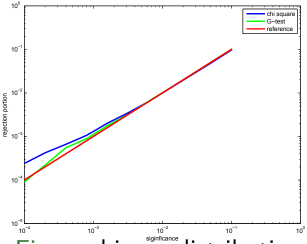


Figure: binary distribution

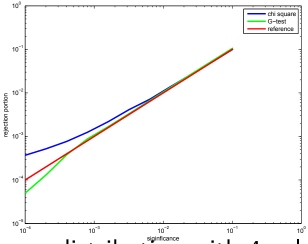


Figure: distribution with 4 values

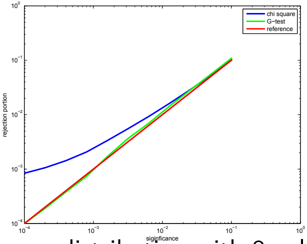


Figure: distribution with 8 values

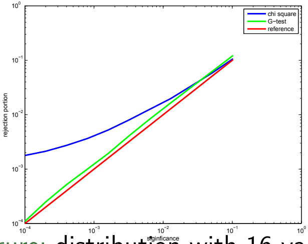
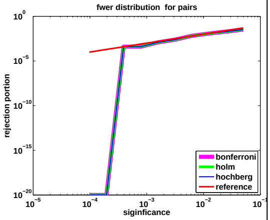
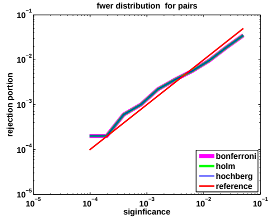
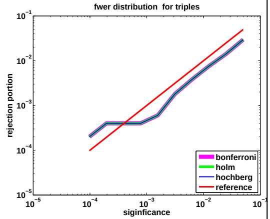
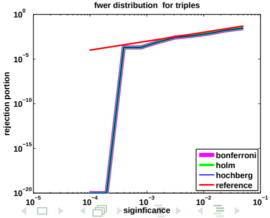


Figure: distribution with 16 values

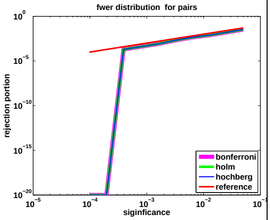
$G - test$, 10 vertices, 9 edges, $Dir(x, 1)$

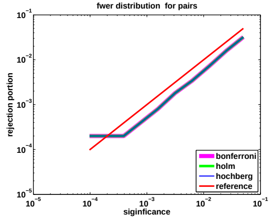
<table>
<tr><th></th><th>Erdős–Rényi</th><th>Barabási–Albert, random order</th></tr>
<tr><td>Pairs</td><td>
</td><td>
</td></tr>
<tr><td>Triples</td><td>
</td><td>
</td></tr>
</table>

$G - test$, 10 vertices, 17-18 edges, $Dir(x, 1)$

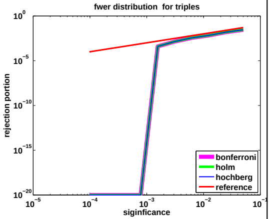
	Erdős–Rényi	Barabási–Albert, random order

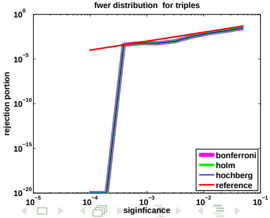
Pairs



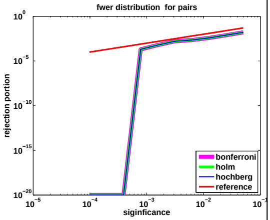
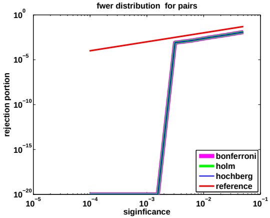
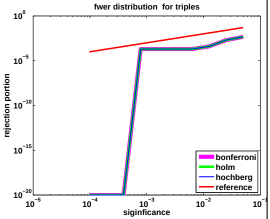
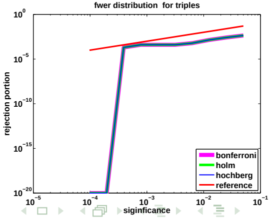


Triples

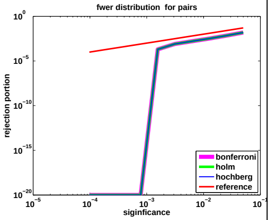
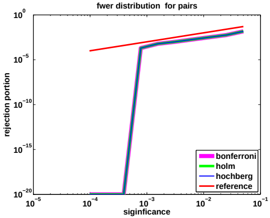
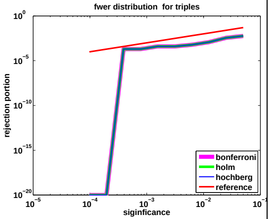
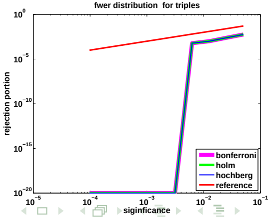




$G - test$, 10 vertices, 9 edges, $Dir(x, 0.2)$

<table>
<tr><th></th><th>Erdős–Rényi</th><th>Barabási–Albert, random order</th></tr>
<tr><td>Pairs</td><td>
</td><td>
</td></tr>
<tr><td>Triples</td><td>
</td><td>
</td></tr>
</table>

$G - test$, 10 vertices, 17-18 edges, $Dir(x, 0.2)$

<table>
<tr><th></th><th>Erdős–Rényi</th><th>Barabási–Albert, random order</th></tr>
<tr><td>Pairs</td><td>
</td><td>
</td></tr>
<tr><td>Triples</td><td>
</td><td>
</td></tr>
</table>

Conclusions

- It's much better to use g-Test than χ^2 test.
- It does not matter which correction procedure we choose.
- Graph does not play a big role.
- Strength of variable interdependence does play an important role.

The End