Learning Credal Networks

Serafín Moral

Dpto. Ciencias de la Computación

Universidad de Granada

Spain

Outline

- Introduction to Bayesian networks.
- Learning parameters.
- Learning interval probabilities.
- Learning the structure.
- Application of an imprecise score.

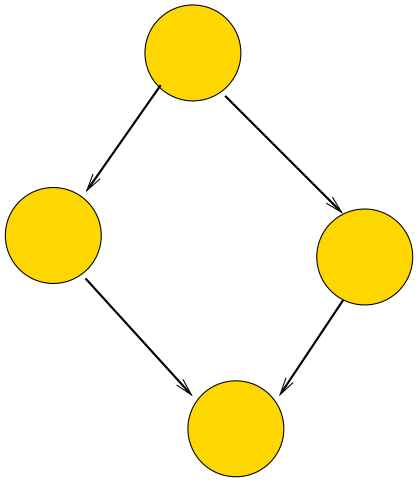
Bayesian Networks

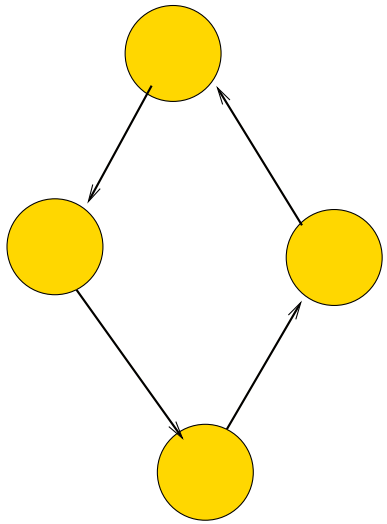
For problems in which we have a set of variables, $\mathbf{X} = \{X_1, \ldots, X_n\}$.

A Bayesian Network has two parts

- *Qualitative:* a directed acyclic graph
  - A node for each problem variable
  - A set of arcs non producing directed cycles.

YES

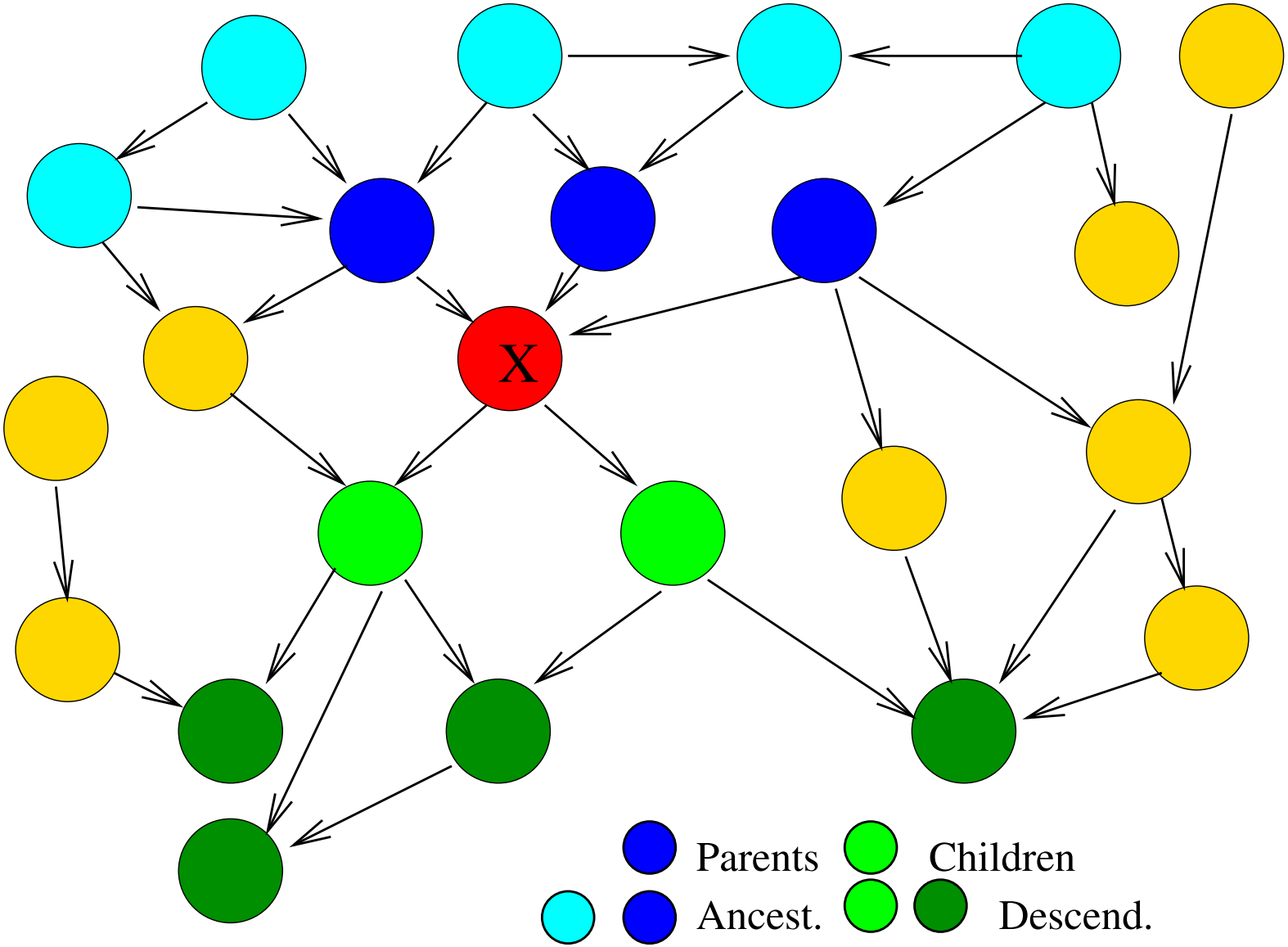




NO

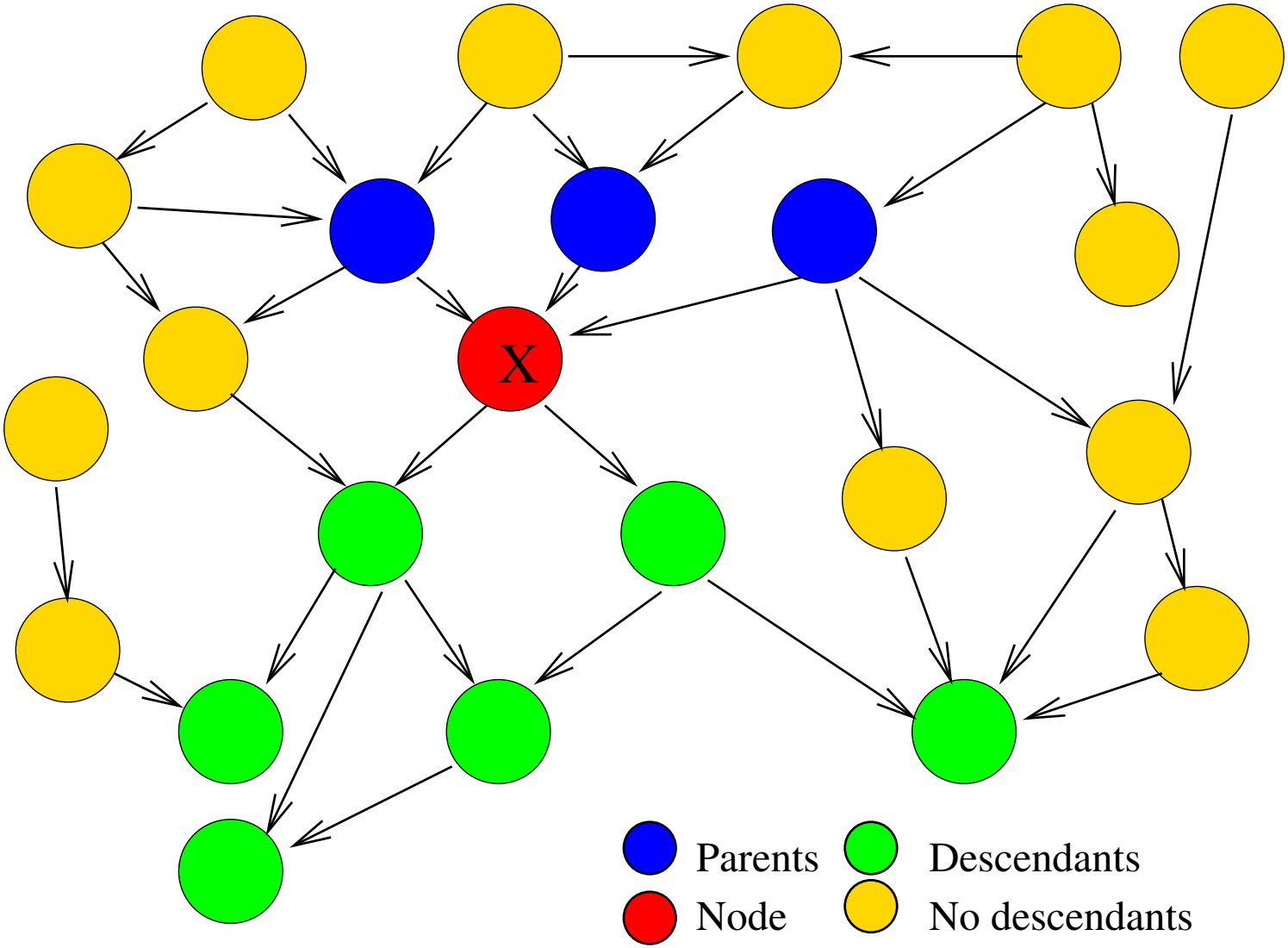
- *Quantitative:* a set of conditinal probability distributions (one for each variable conditioned to its parents).

Bayesian Networks



Basic Independences

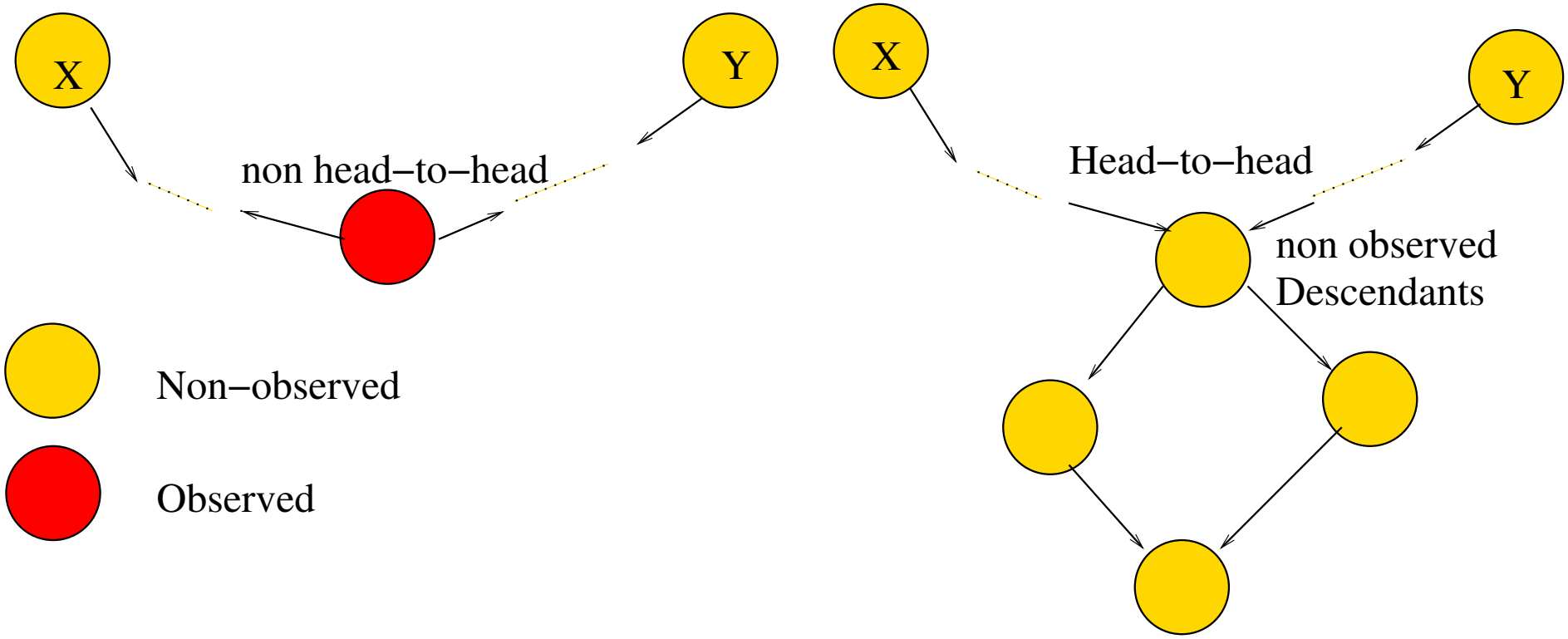
Each node is independent of its non-descendants given its parents.



Additional Independence: D-separation

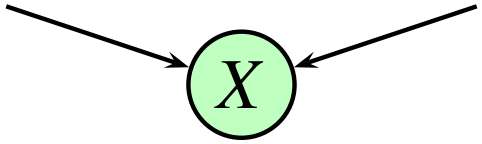
X is independent of Y given $Z_1, \ldots, Z_k$ if **any** path (using arcs in both directions) between X and Y is blocked in some node by observations $Z_1, \ldots, Z_k$.

Two ways of blocking:

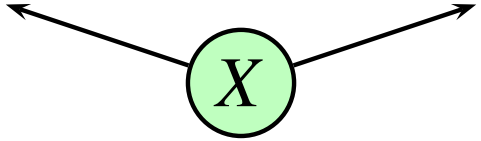


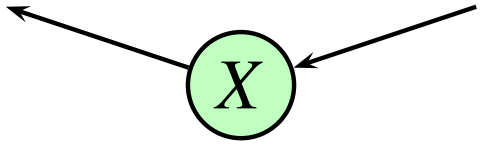
Head-to-head

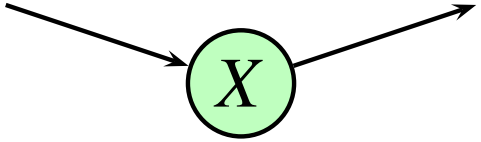
Head-to-head



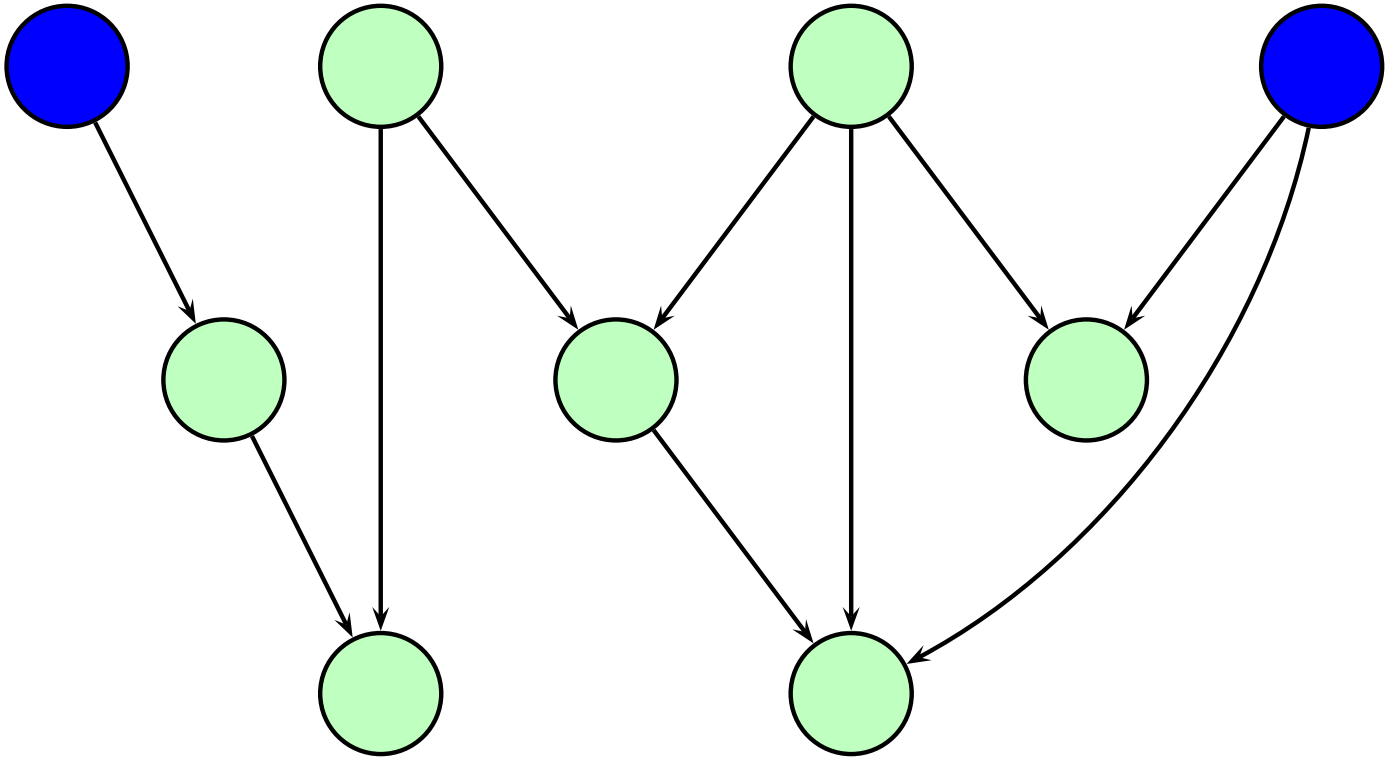
Non head-to-head

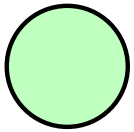




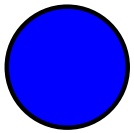


Independence Examples

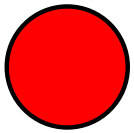




Rest of variables

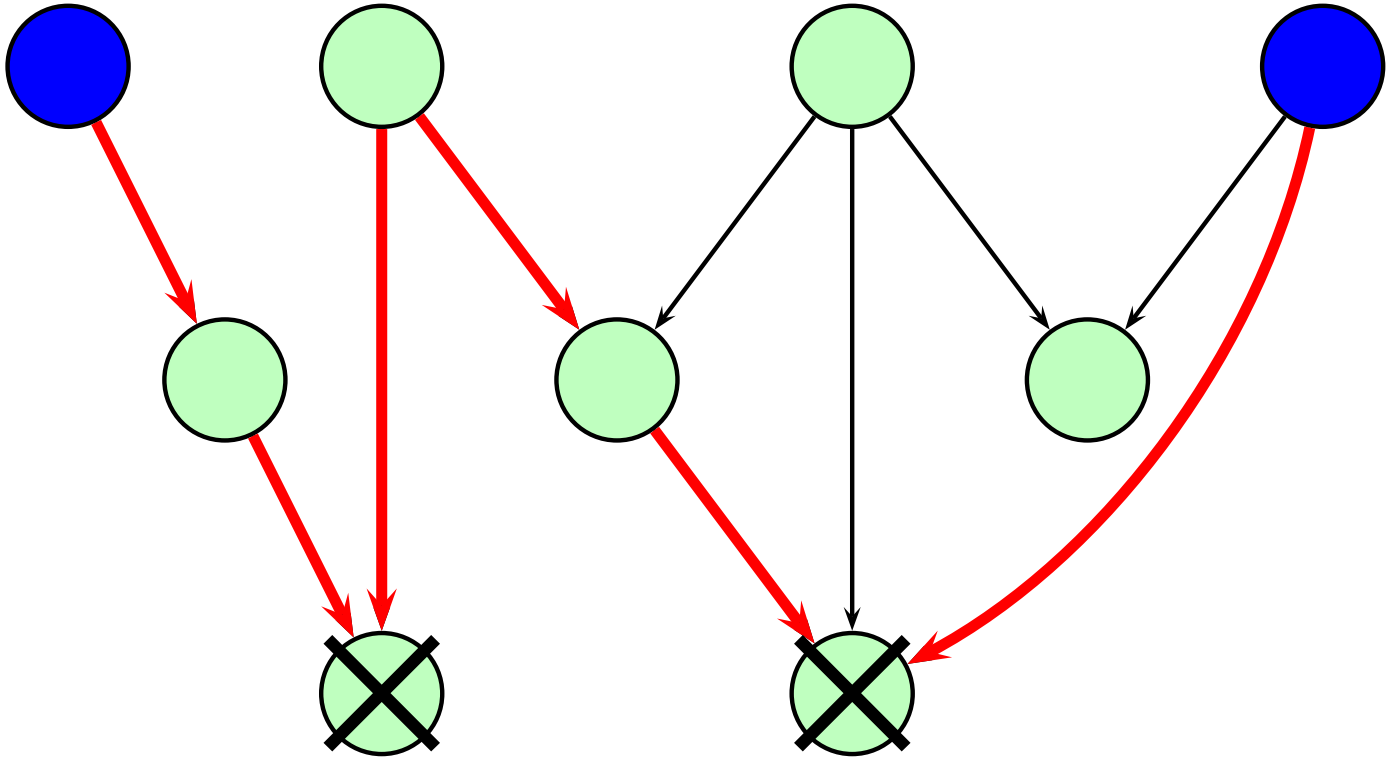


Examined variables



Observed Node

Independence Examples



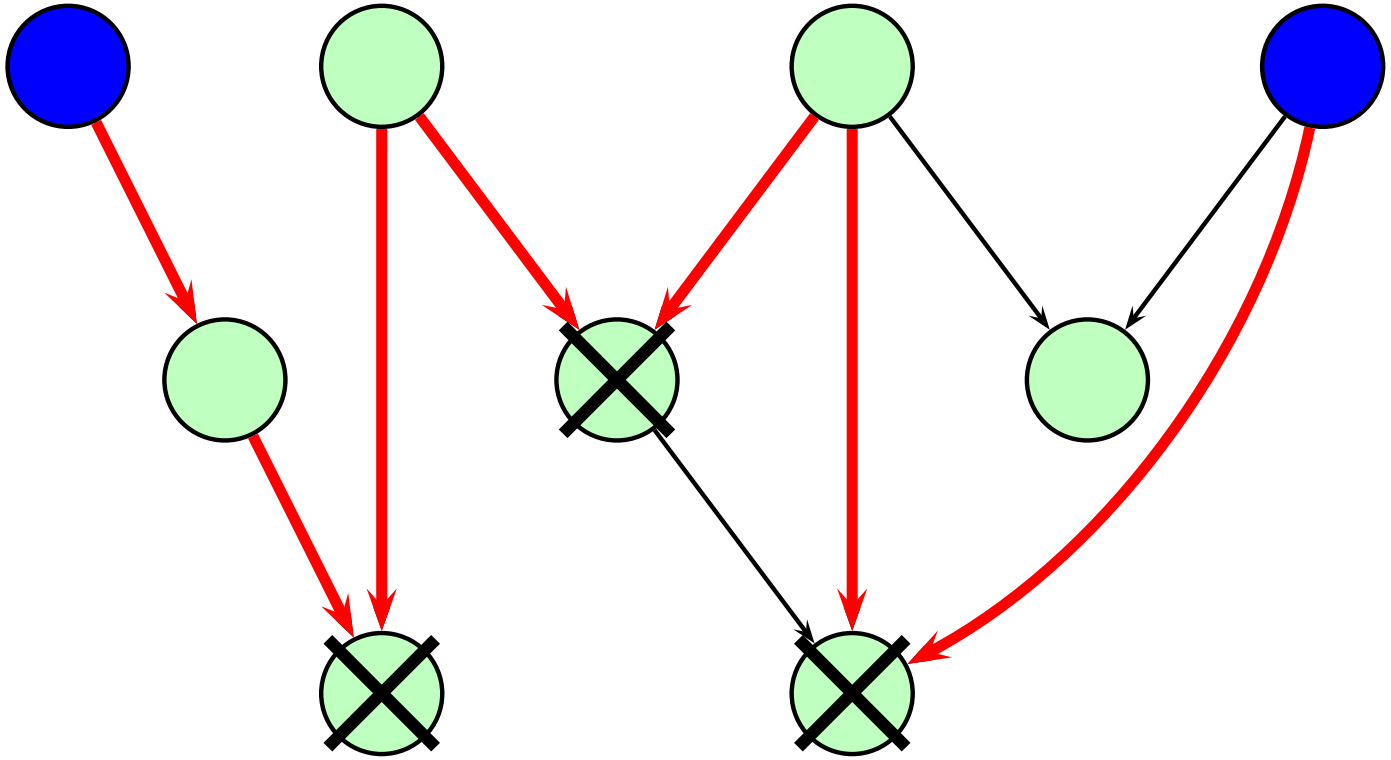
Rest of variables

Examined variables

Observed Node

First path blocked

Independence Examples



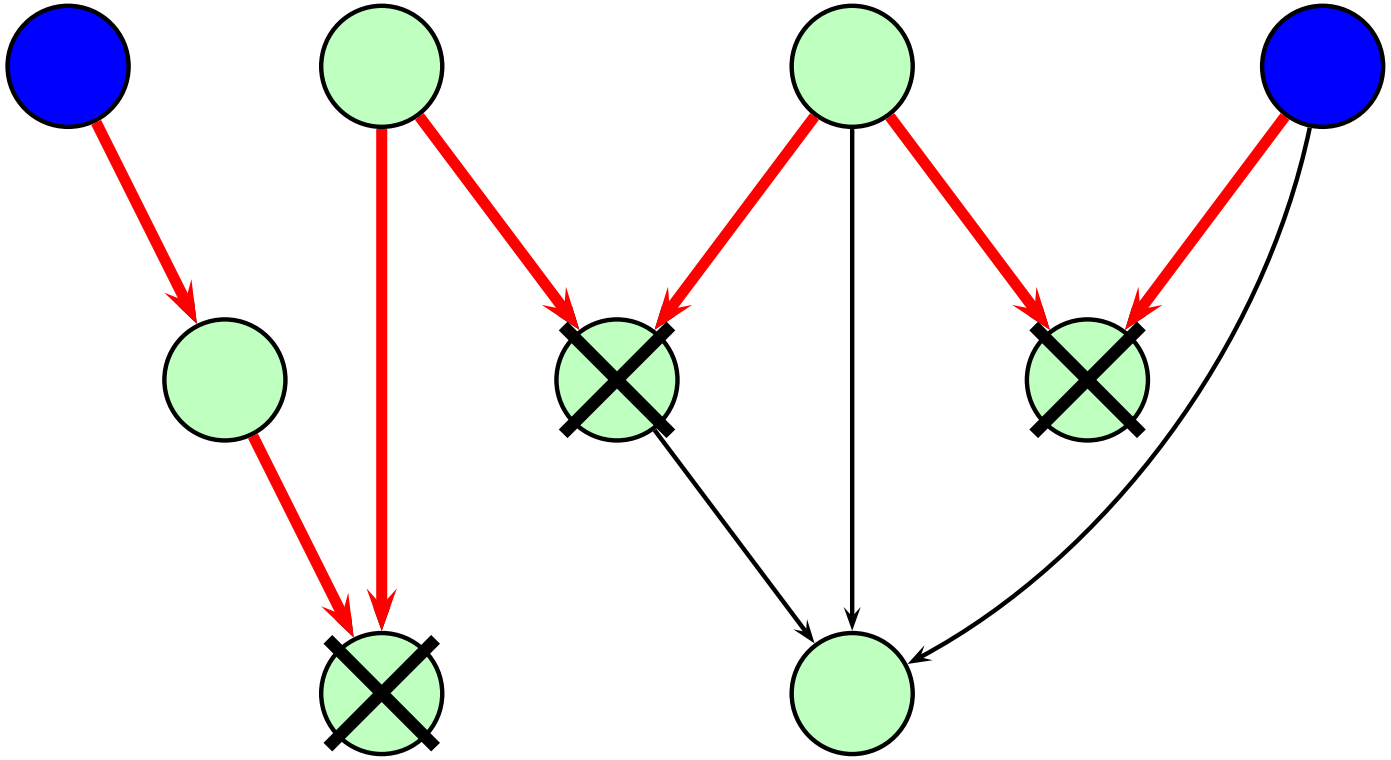
Rest of variables

Examined variables

Observed Node

Second path blocked

Independence Examples



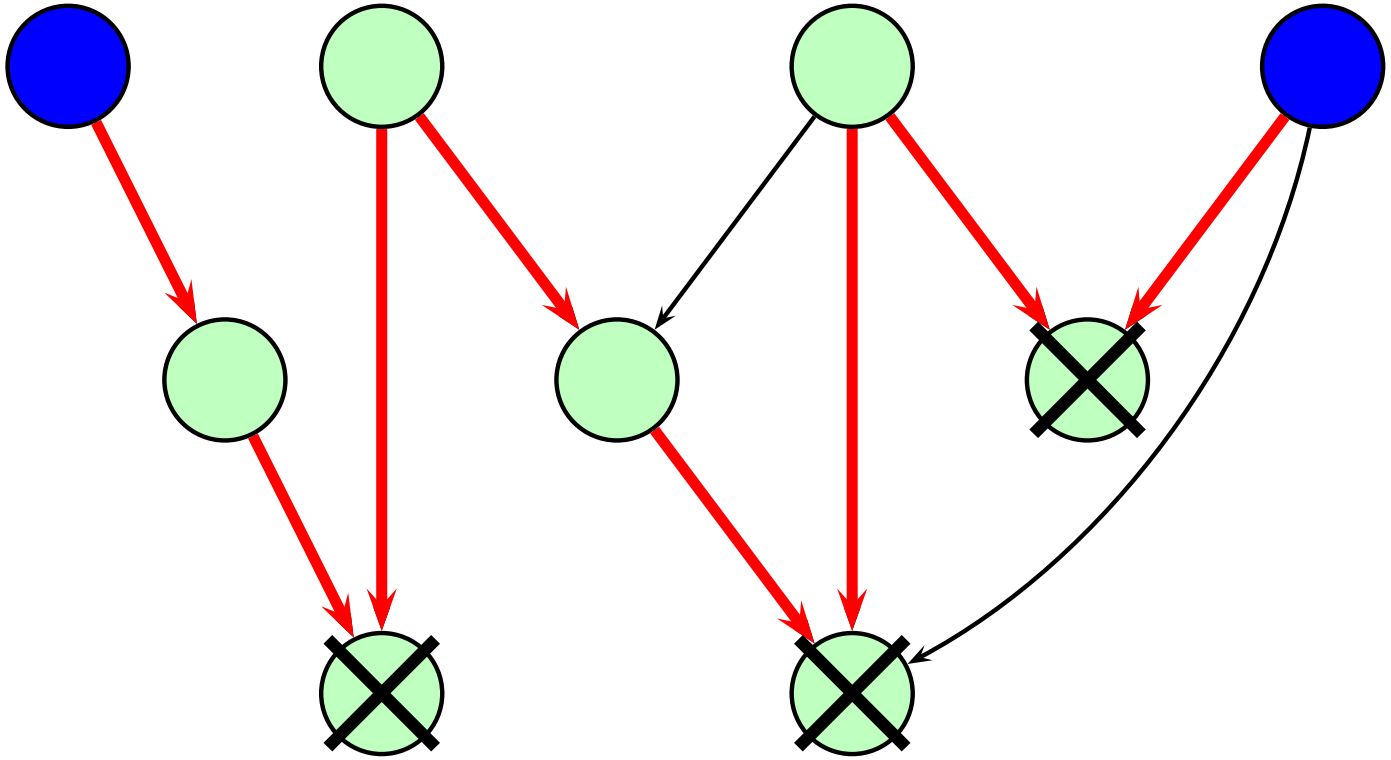
Rest of variables

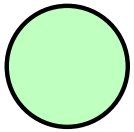
Examined variables

Observed Node

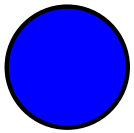
Third path blocked

Independence Examples

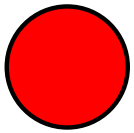




Rest of variables



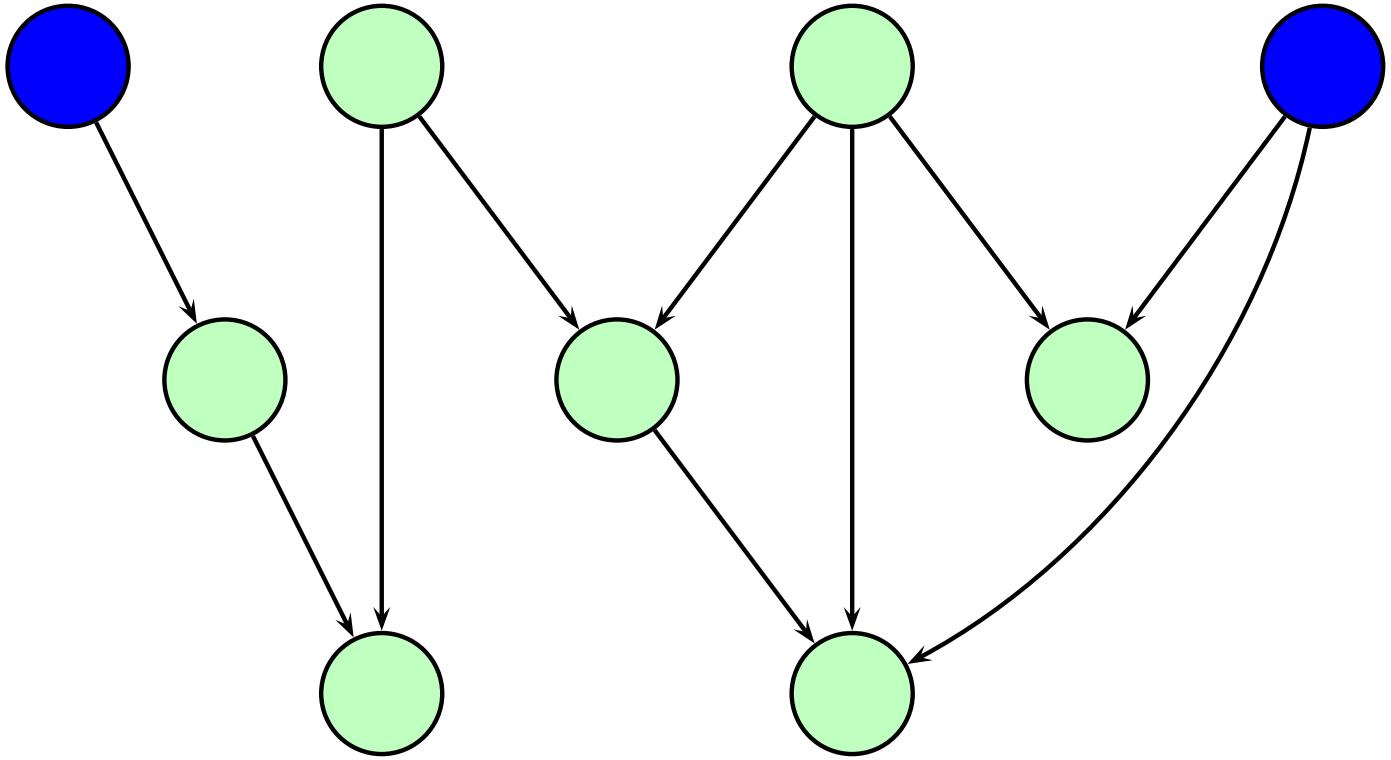
Examined variables



Observed Node

Forth path blocked

Independence Examples



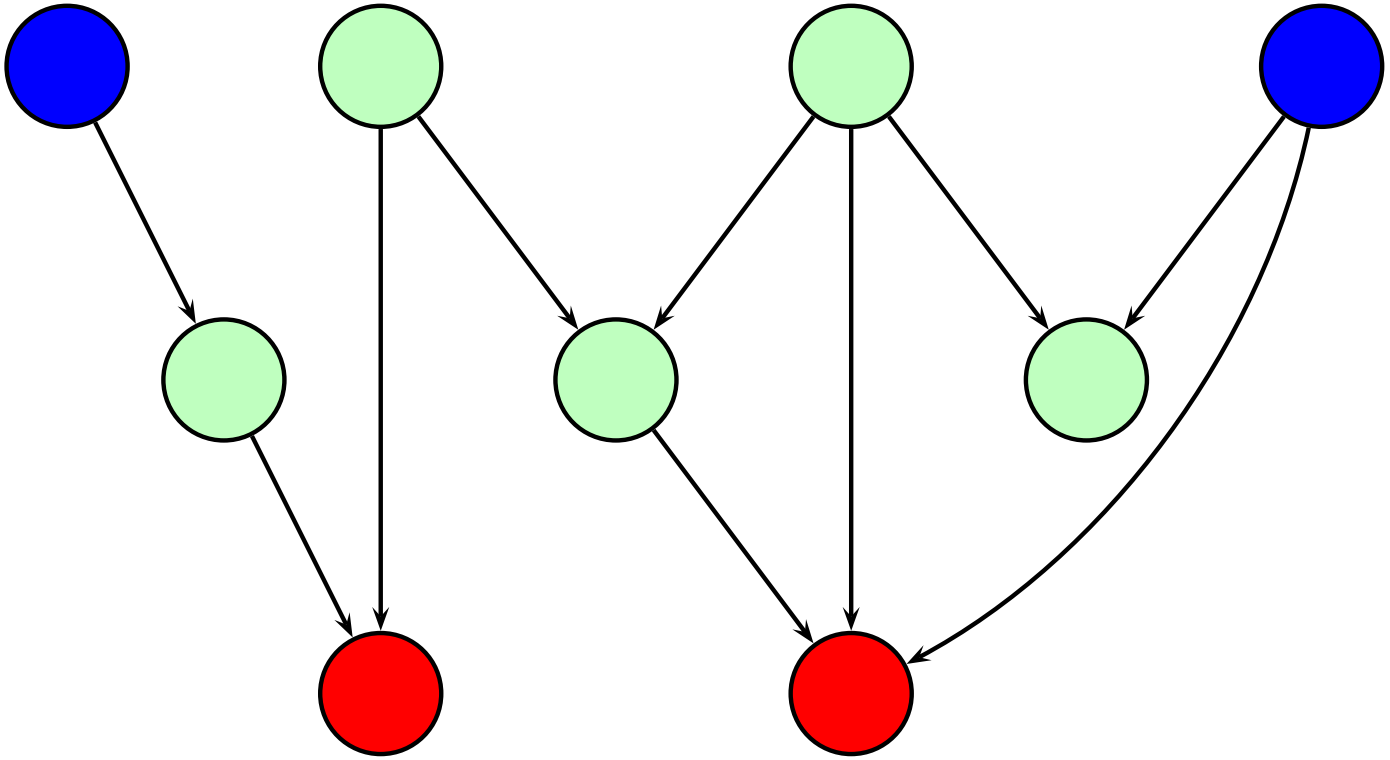
Rest of variables

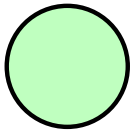
Examined variables

Observed Node

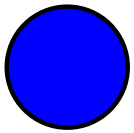
Independent Variables (Non obs.)

Independence Examples

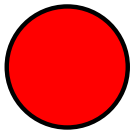




Rest of variables

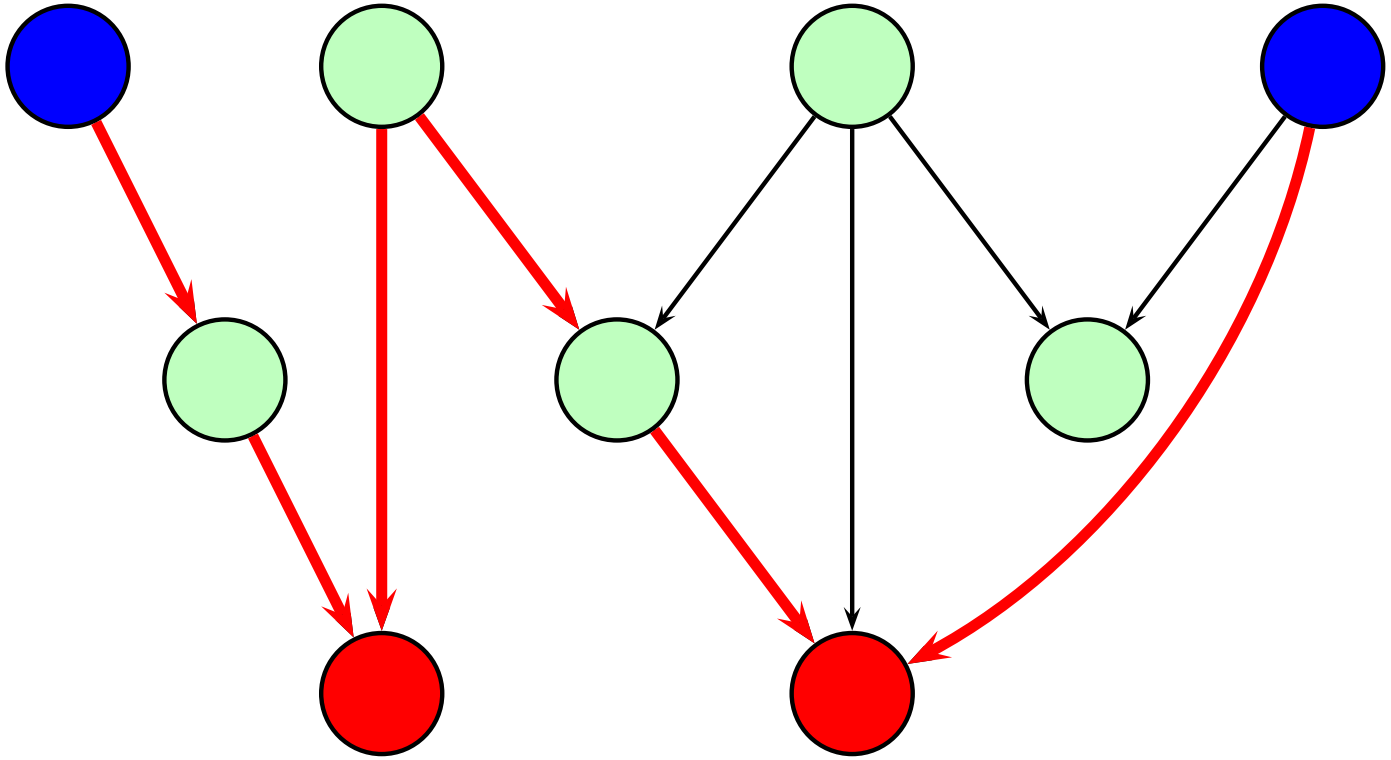


Tested variables



Observed Node

Independence Examples



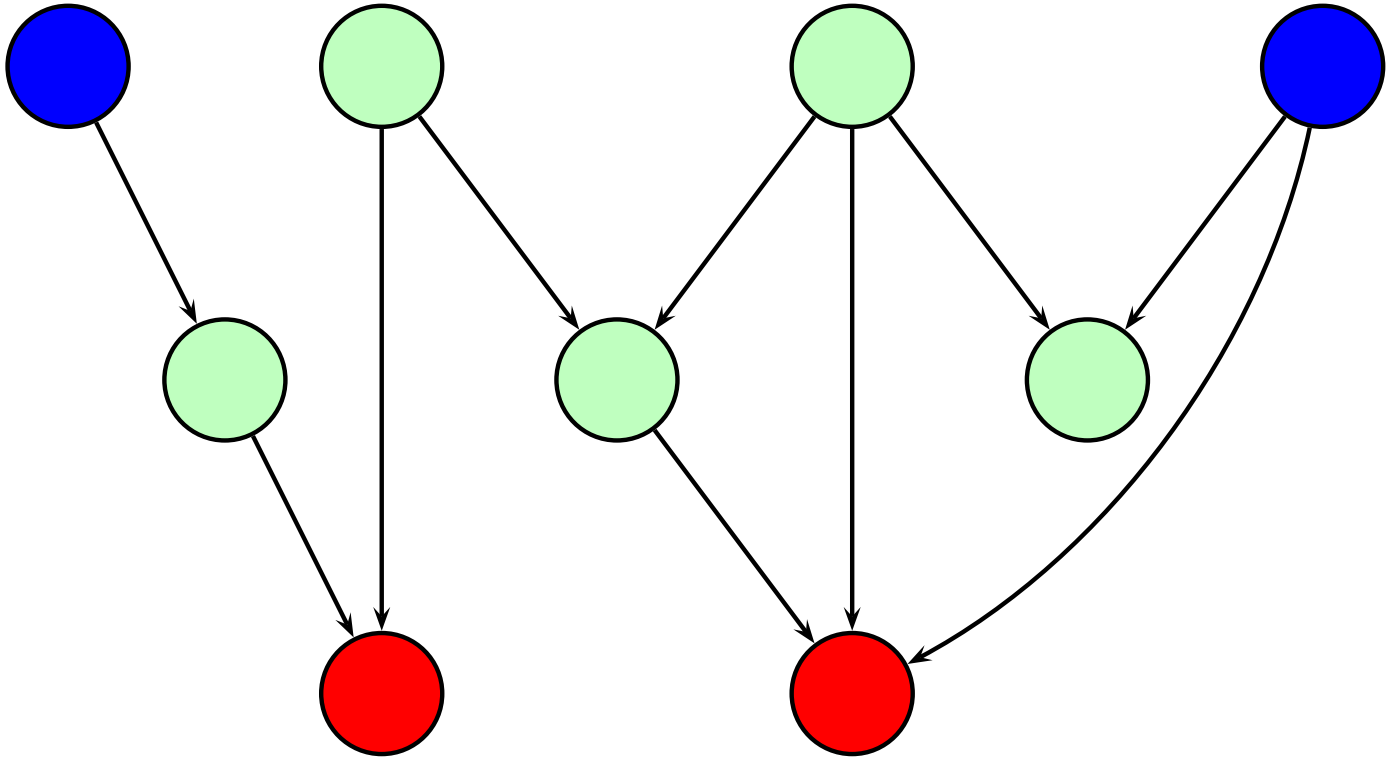
Rest of variables

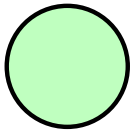
Tested variables

Observed Node

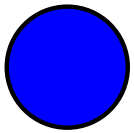
First path NON blocked

Independence Examples

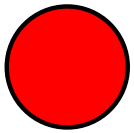




Rest of variables



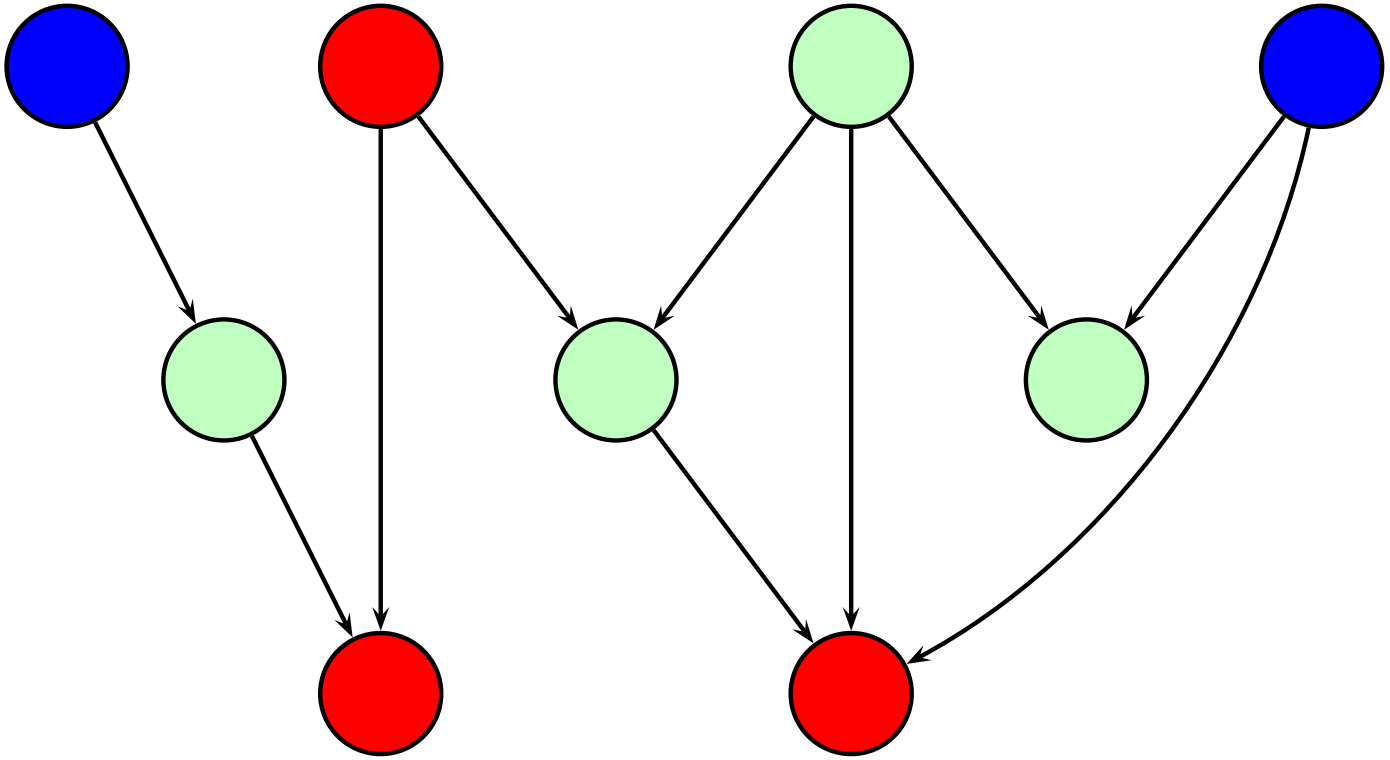
Tested variables



Observed Node

Dependent variables (Red obs.)

Independence Examples

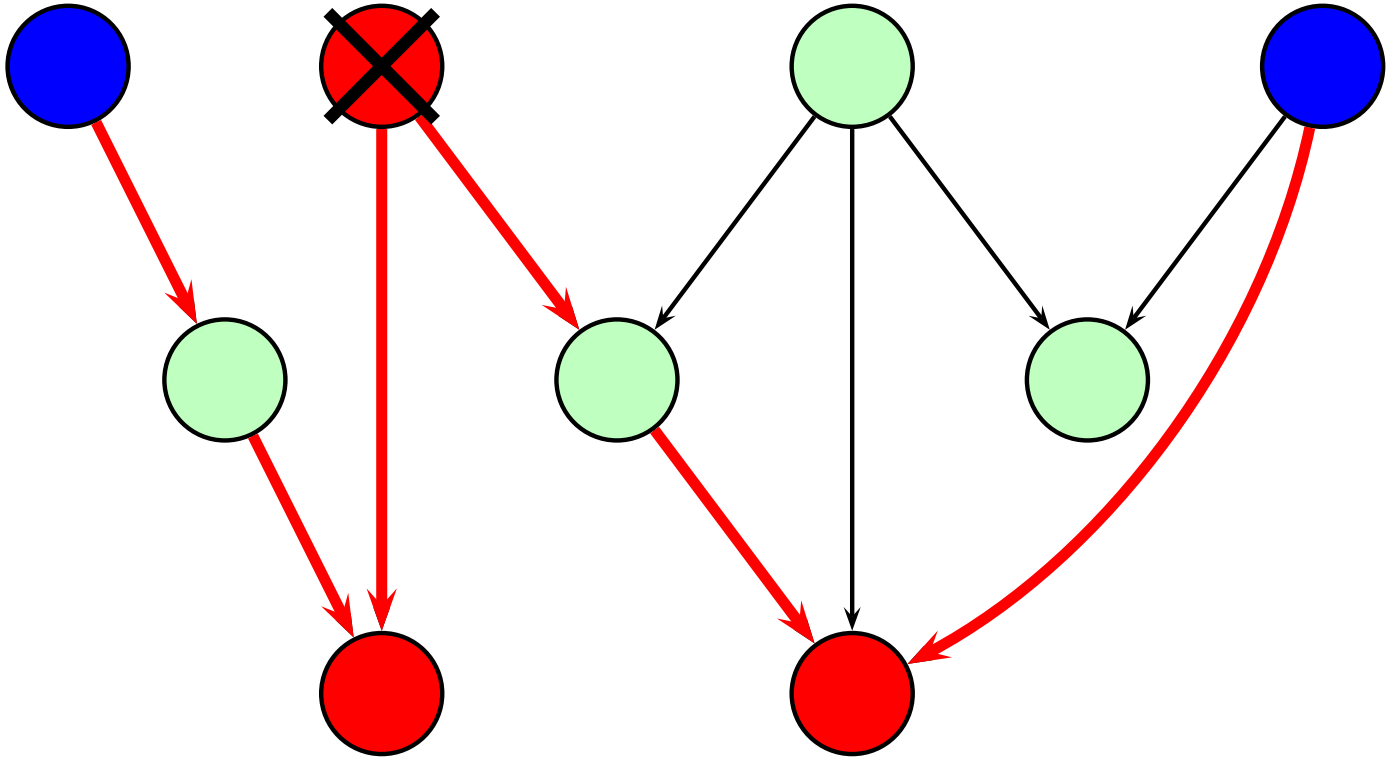


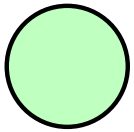
Rest of variables

Tested variables

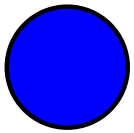
Observed Node

Independence Examples

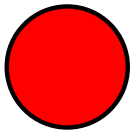




Rest of variables



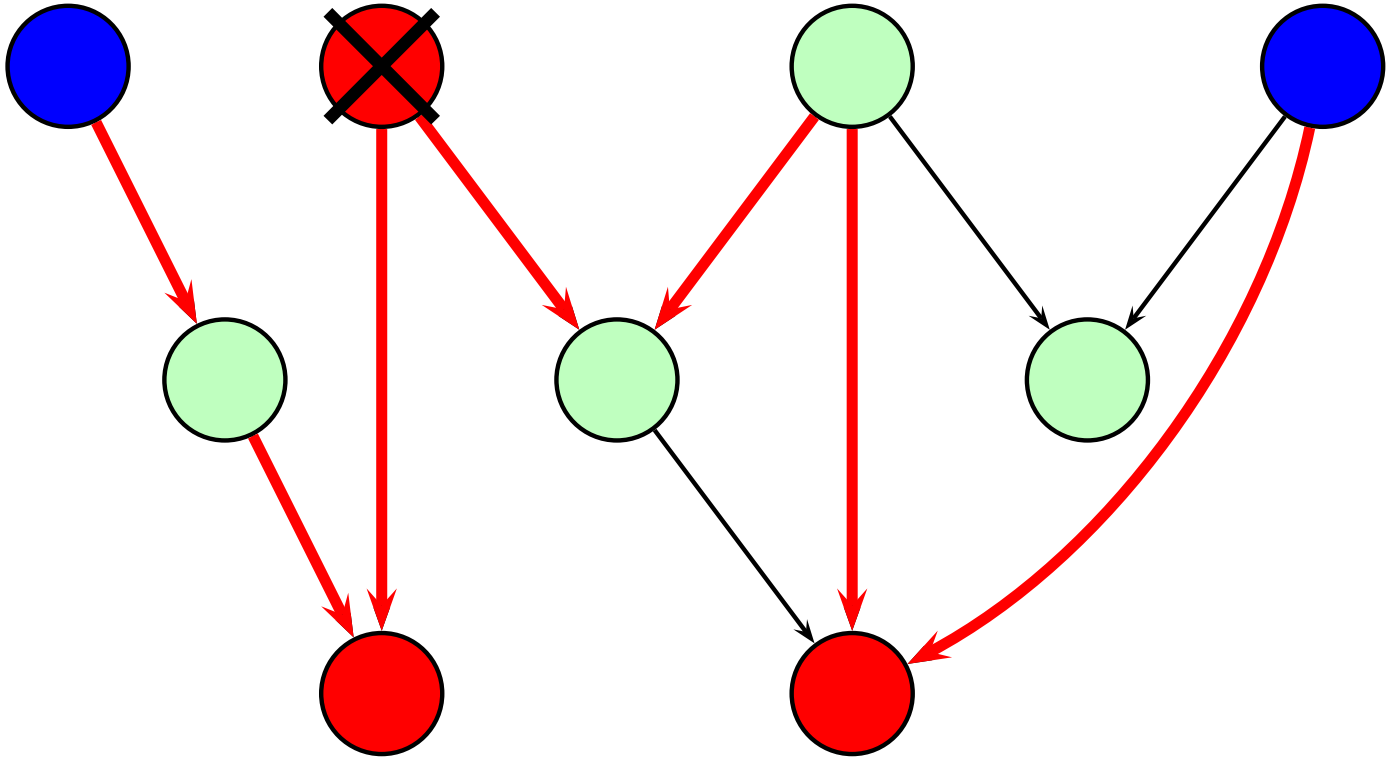
Tested variables

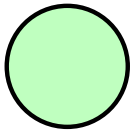


Observed Node

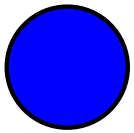
First path blocked

Independence Examples

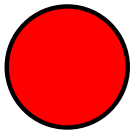




Rest of variables



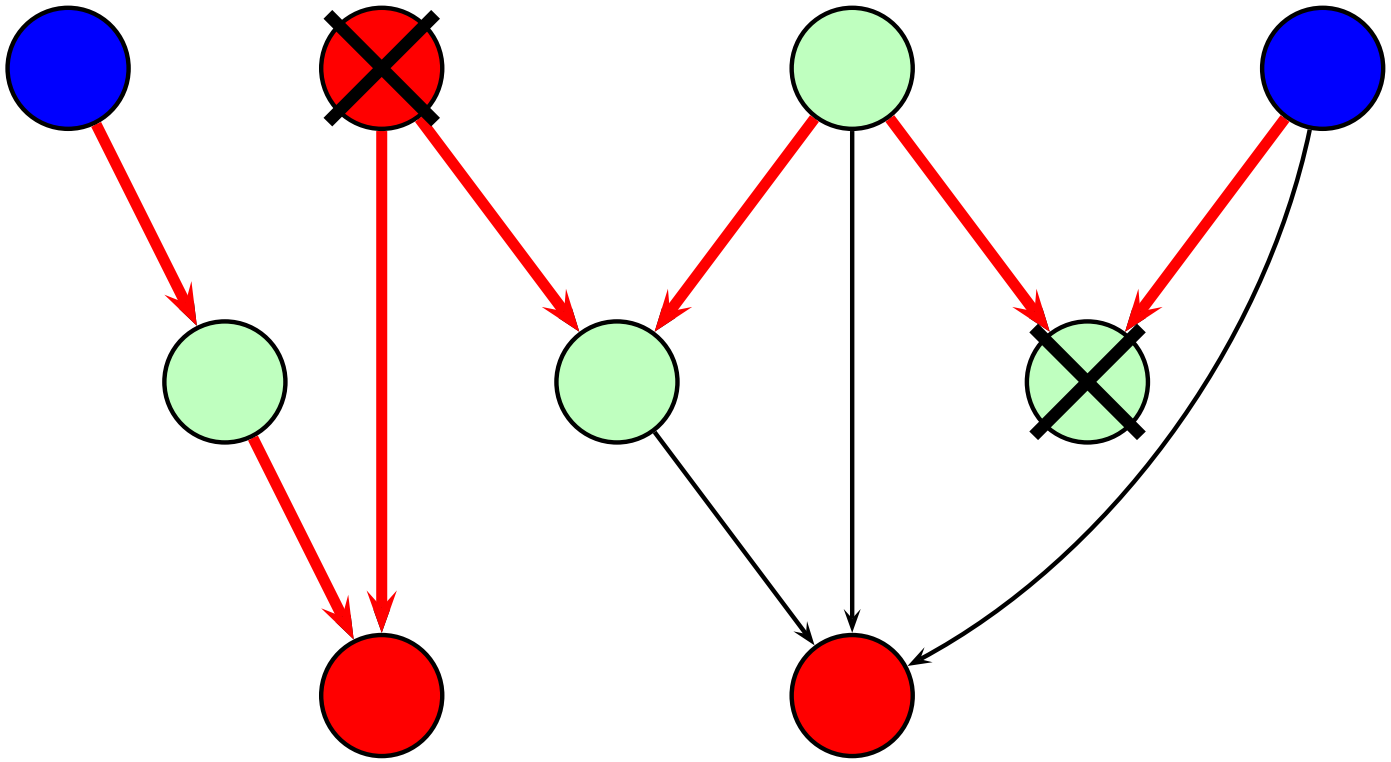
Tested variables

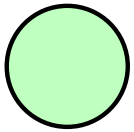


Observed Node

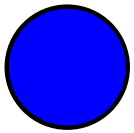
Second path blocked

Independence Examples

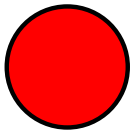




Rest of variables



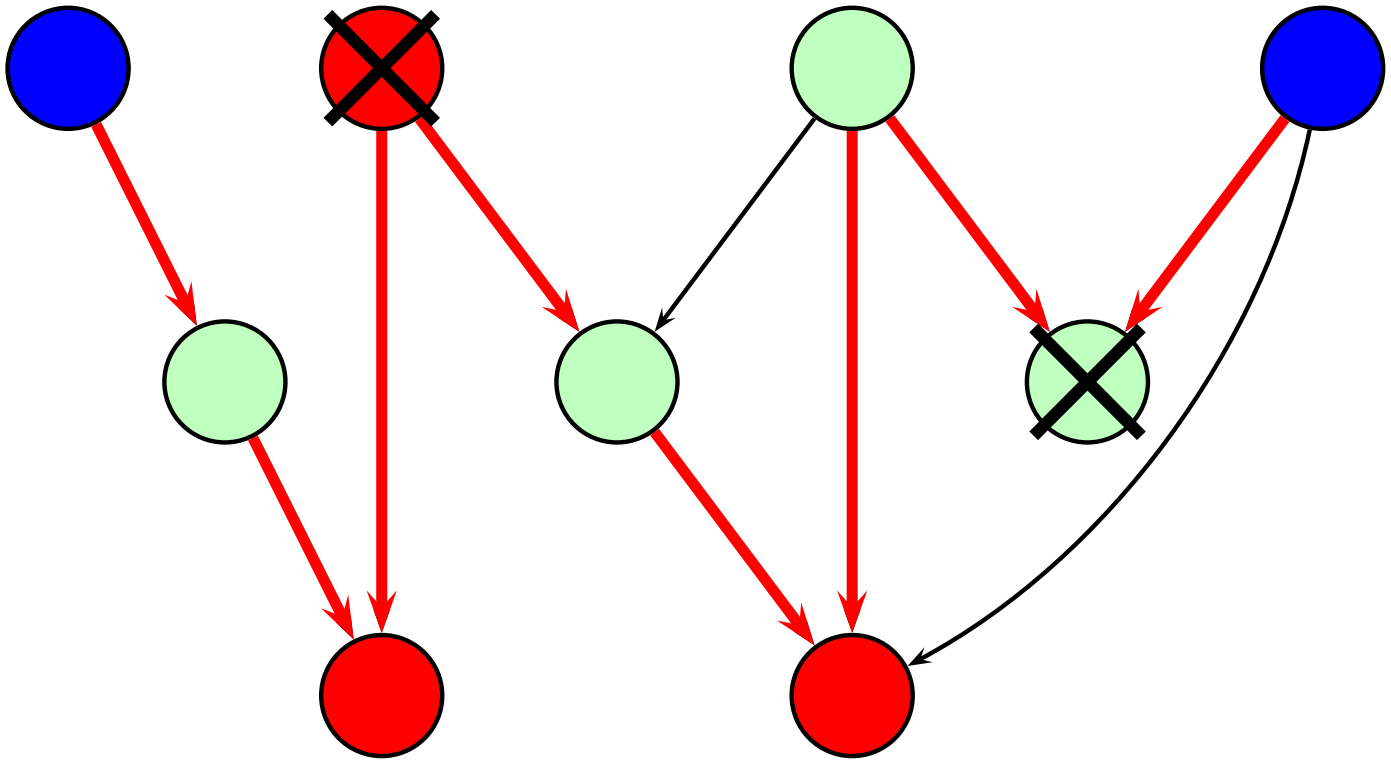
Tested variables

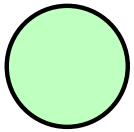


Observed Node

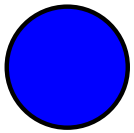
Third path blocked

Independence Examples

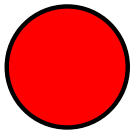




Rest of variables



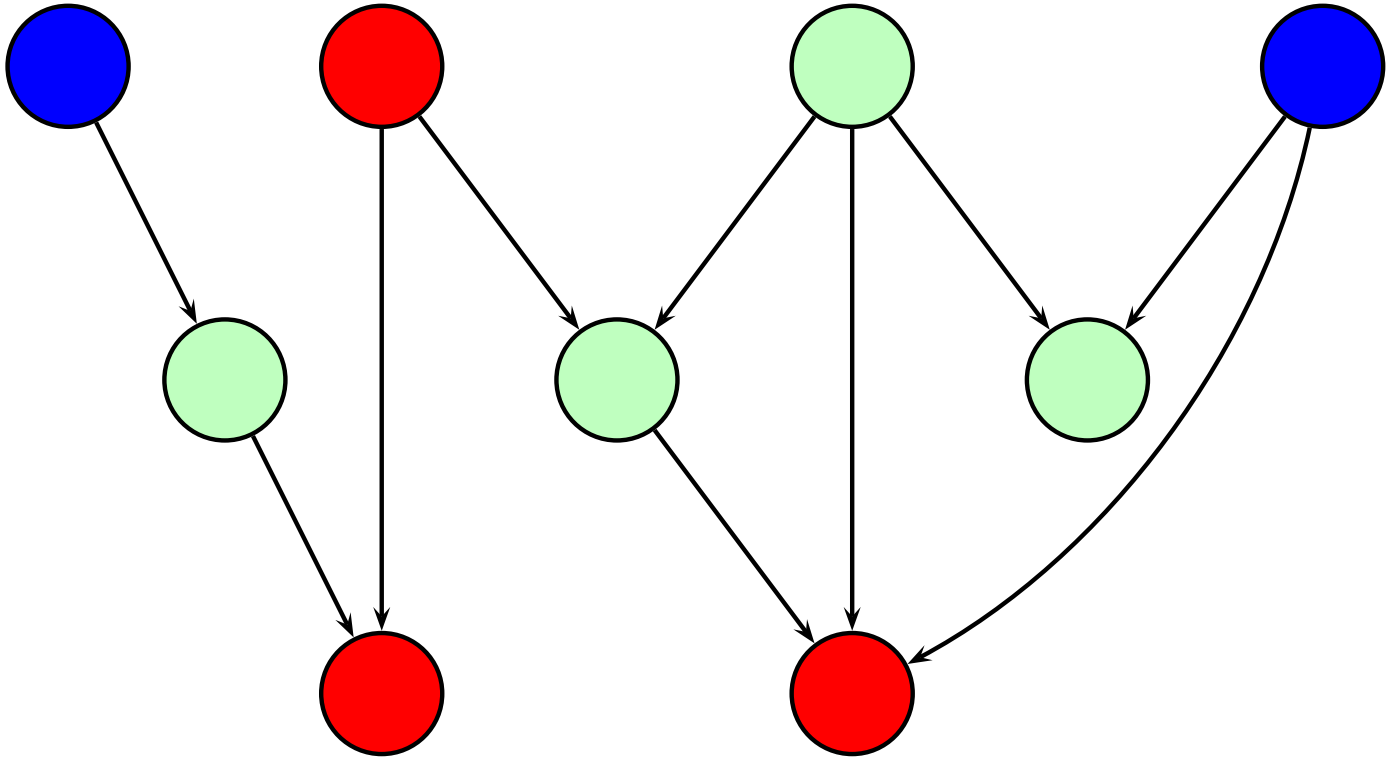
Tested variables

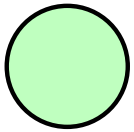


Observed Node

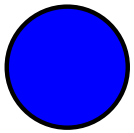
Forth path blocked

Independence Examples

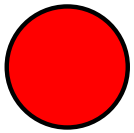




Rest of variables



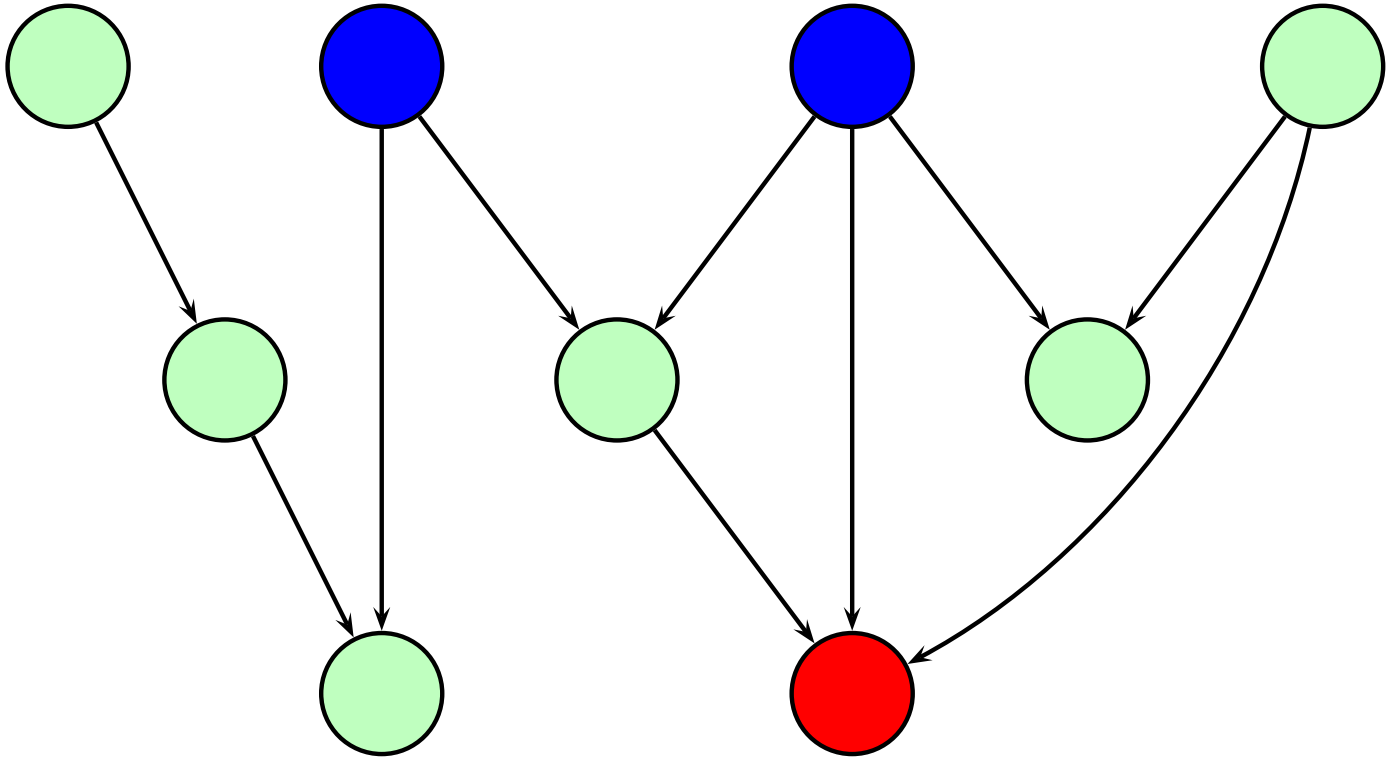
Tested variables



Observed Node

Independent Variables (Red obs.)

Independence Examples

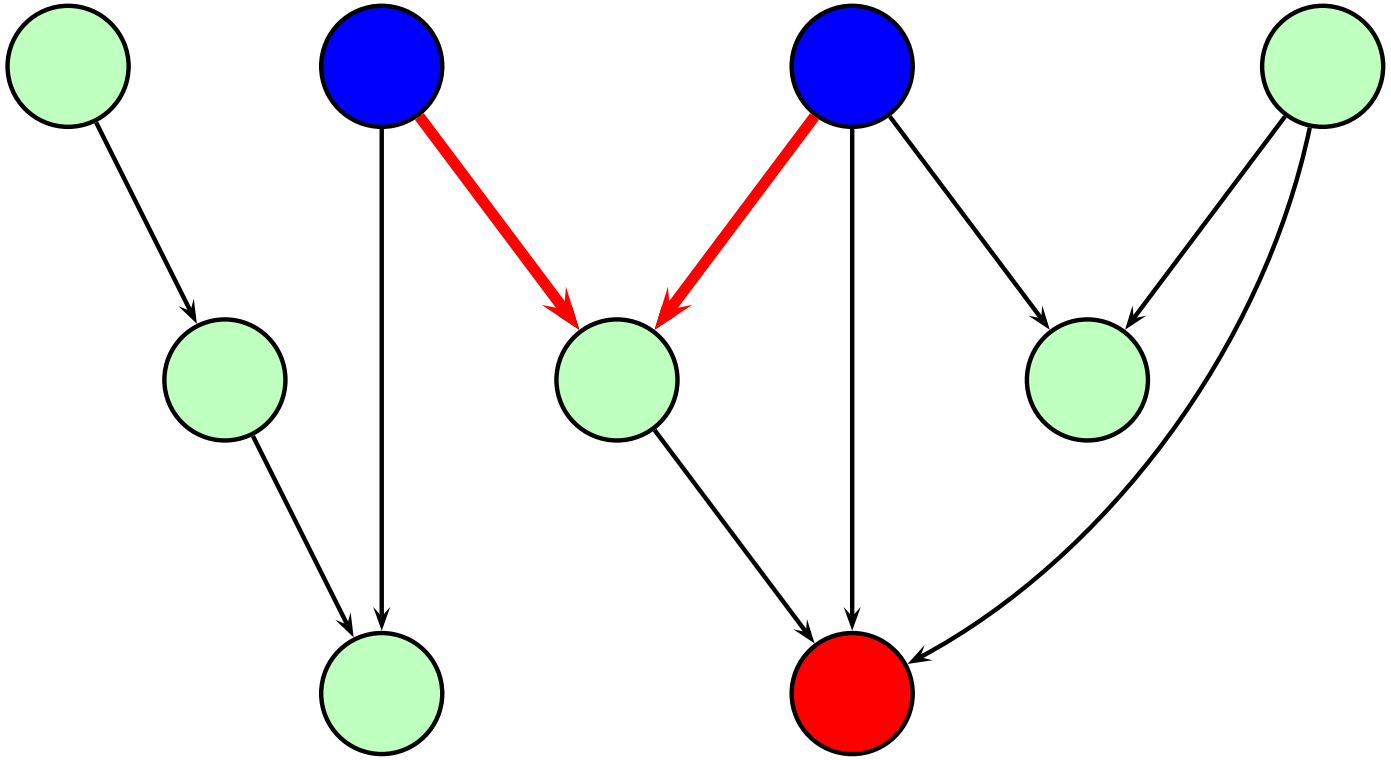


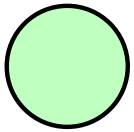
Rest of variables

Tested variables

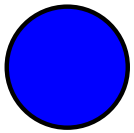
Observed Node

Independence Examples

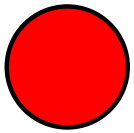




Rest of variables



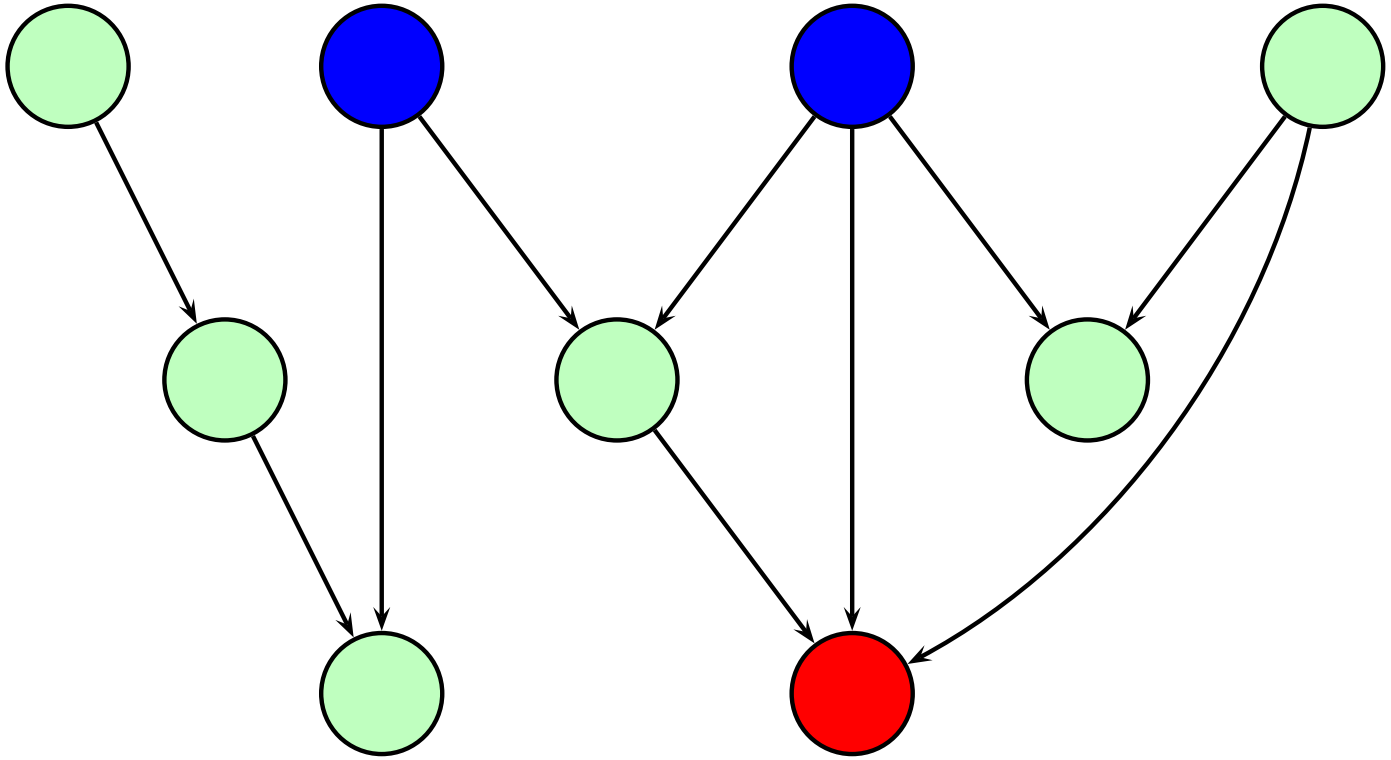
Tested variables

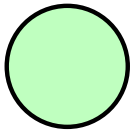


Observed Node

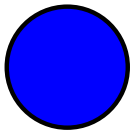
First path NON blocked

Independence Examples

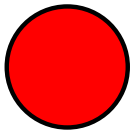




Rest of variables



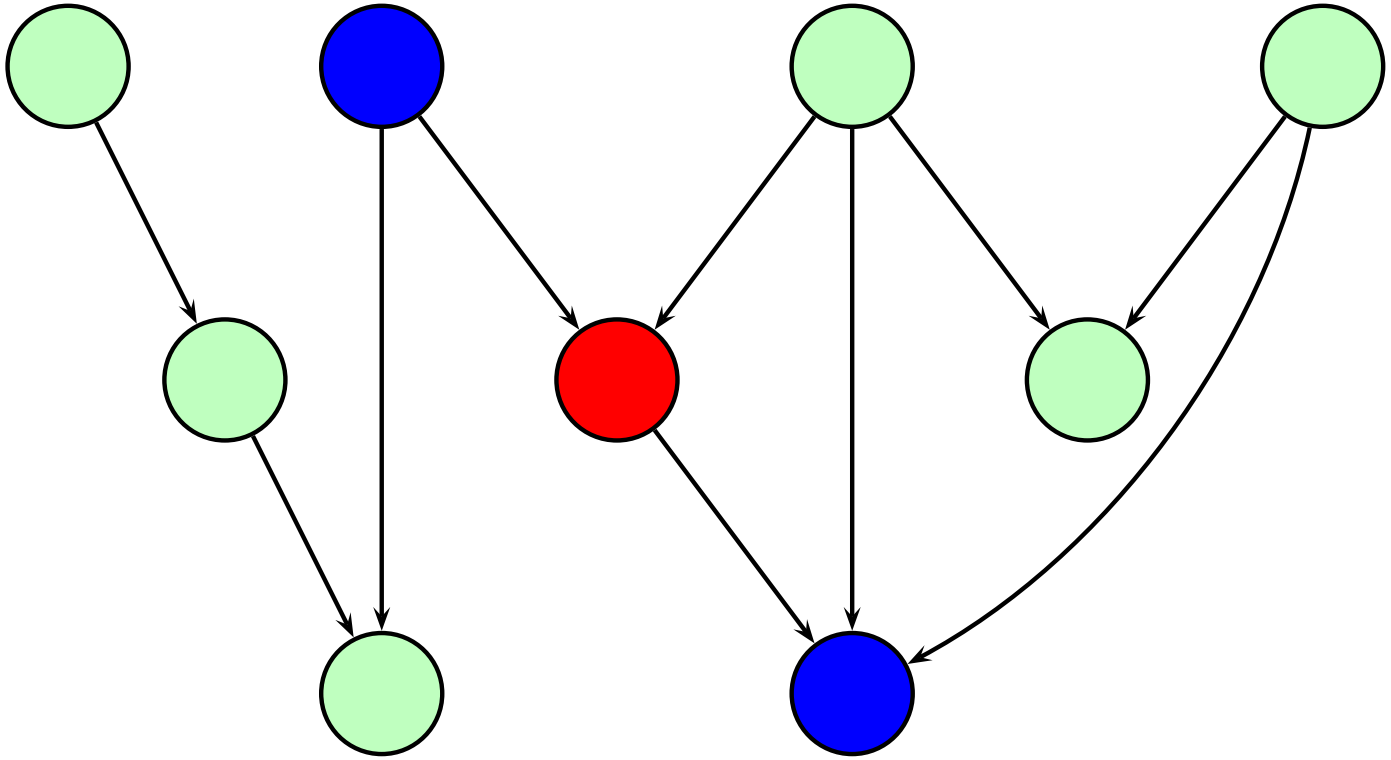
Tested variables

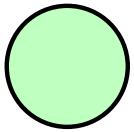


Observed Node

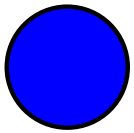
Dependent variables (Red obs.)

Independence Examples

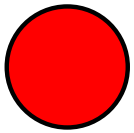




Rest of variables

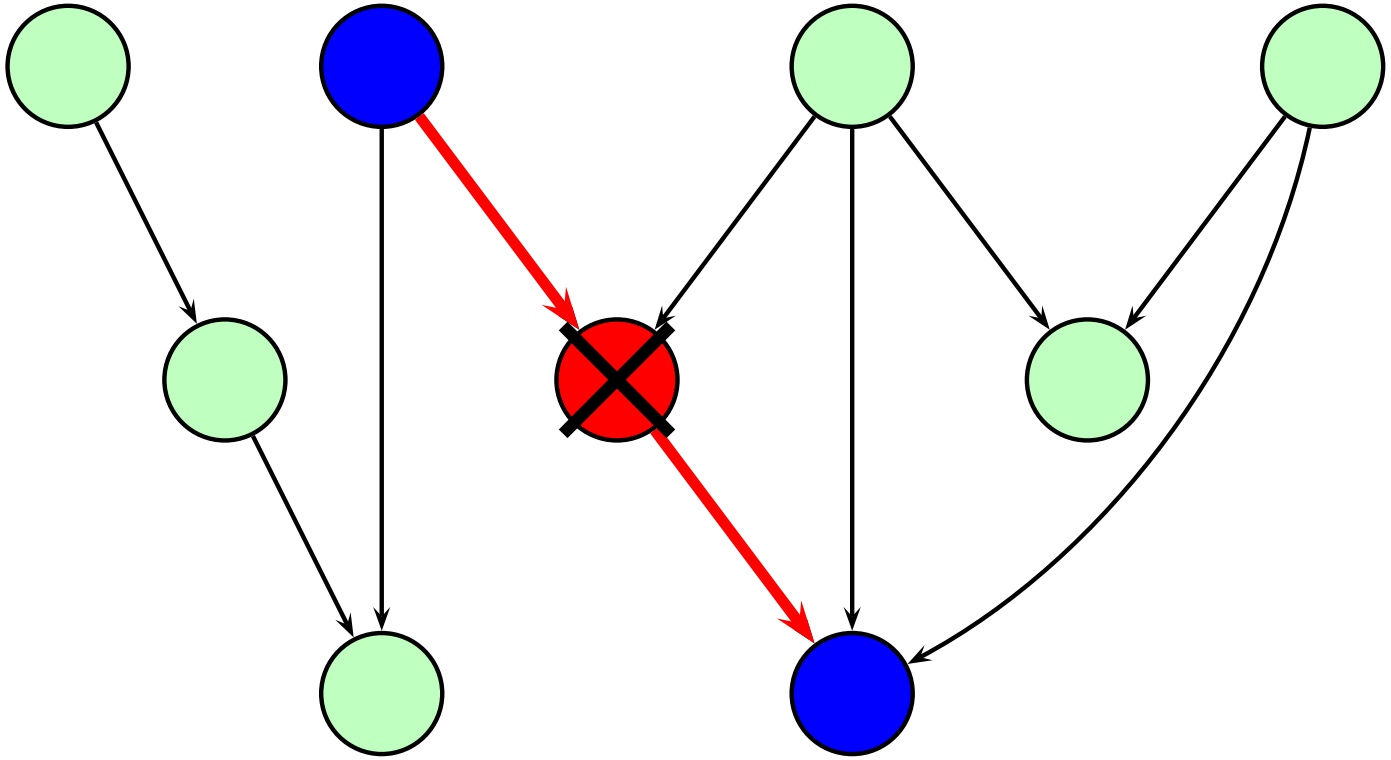


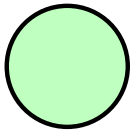
Tested variables



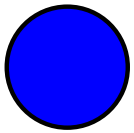
Observed Node

Independence Examples

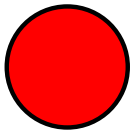




Rest of variables



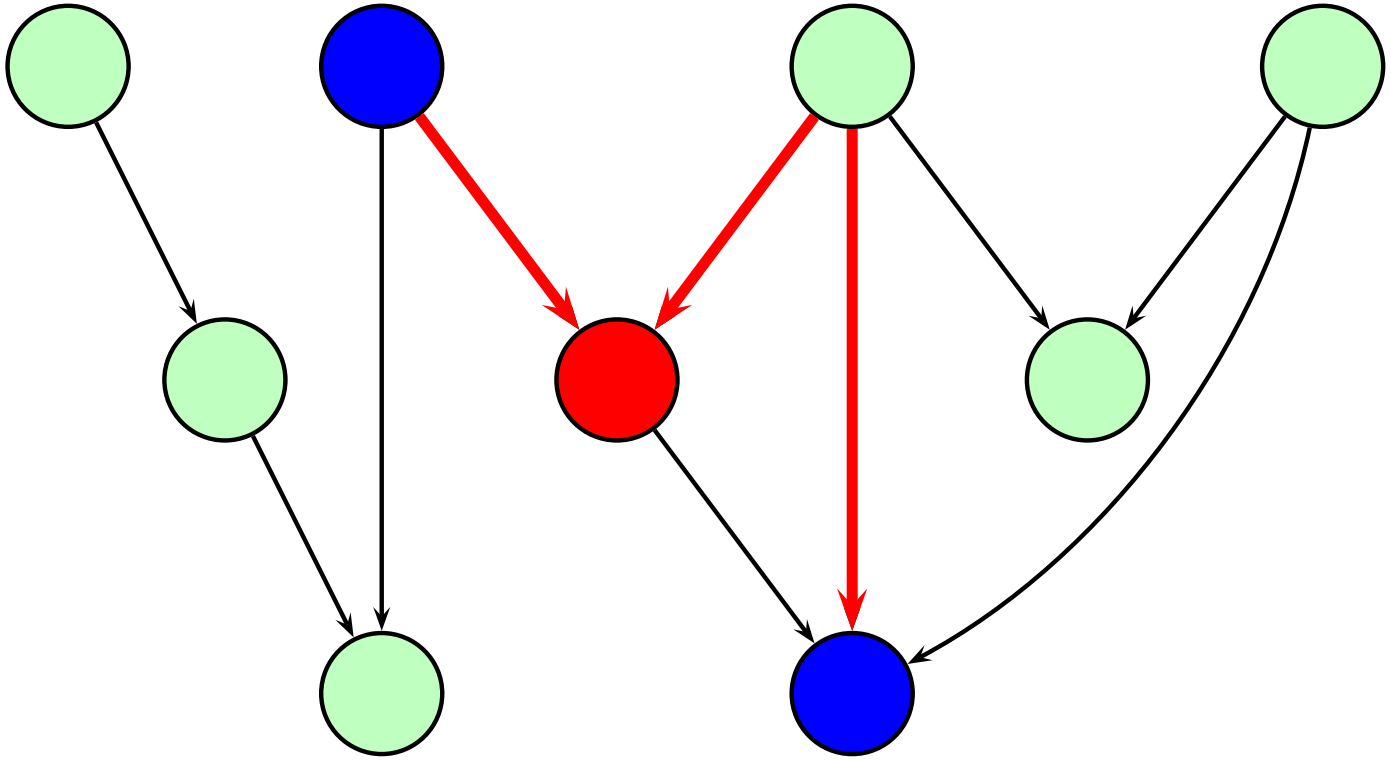
Tested variables



Observed Node

First path blocked

Independence Examples



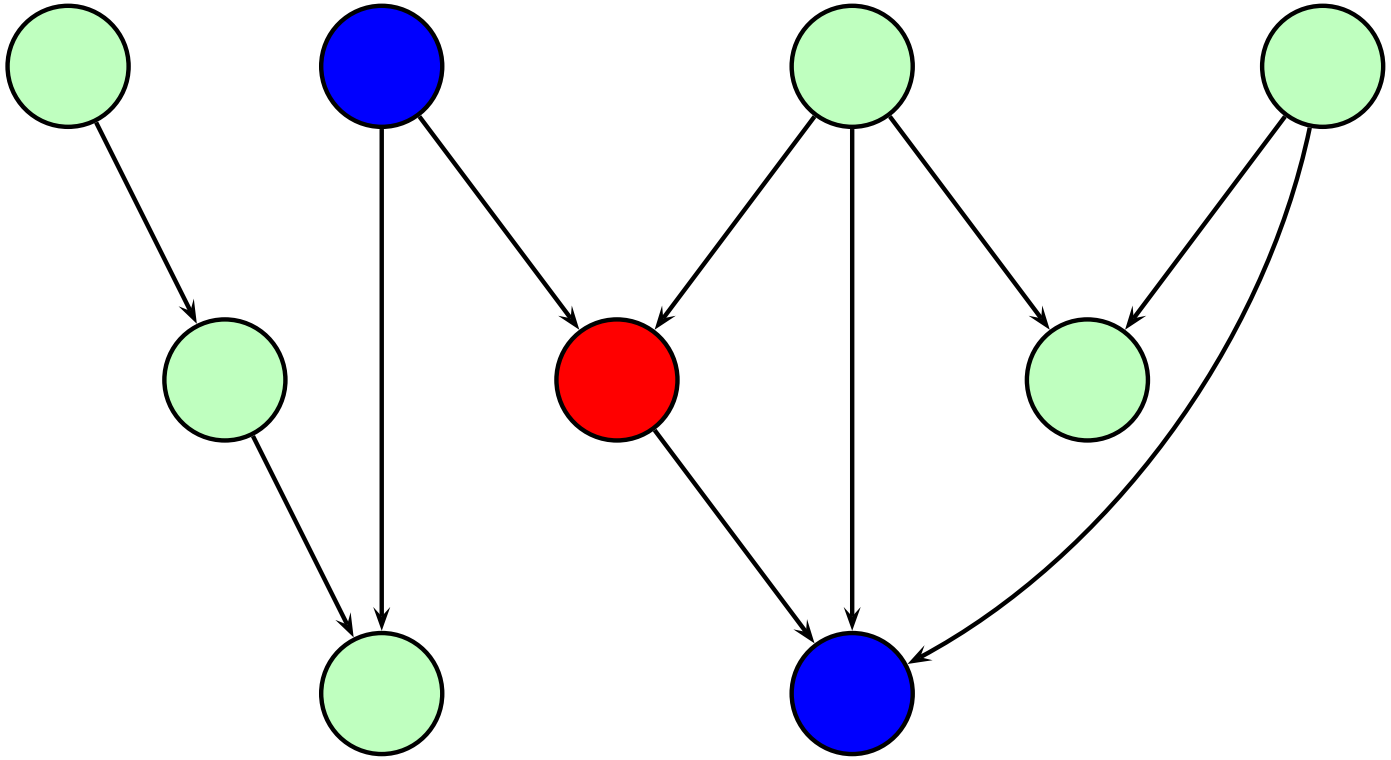
Rest of variables

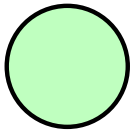
Tested variables

Observed Node

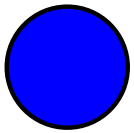
Second path NO blocked

Independence Examples

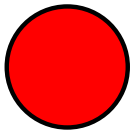




Rest of variables



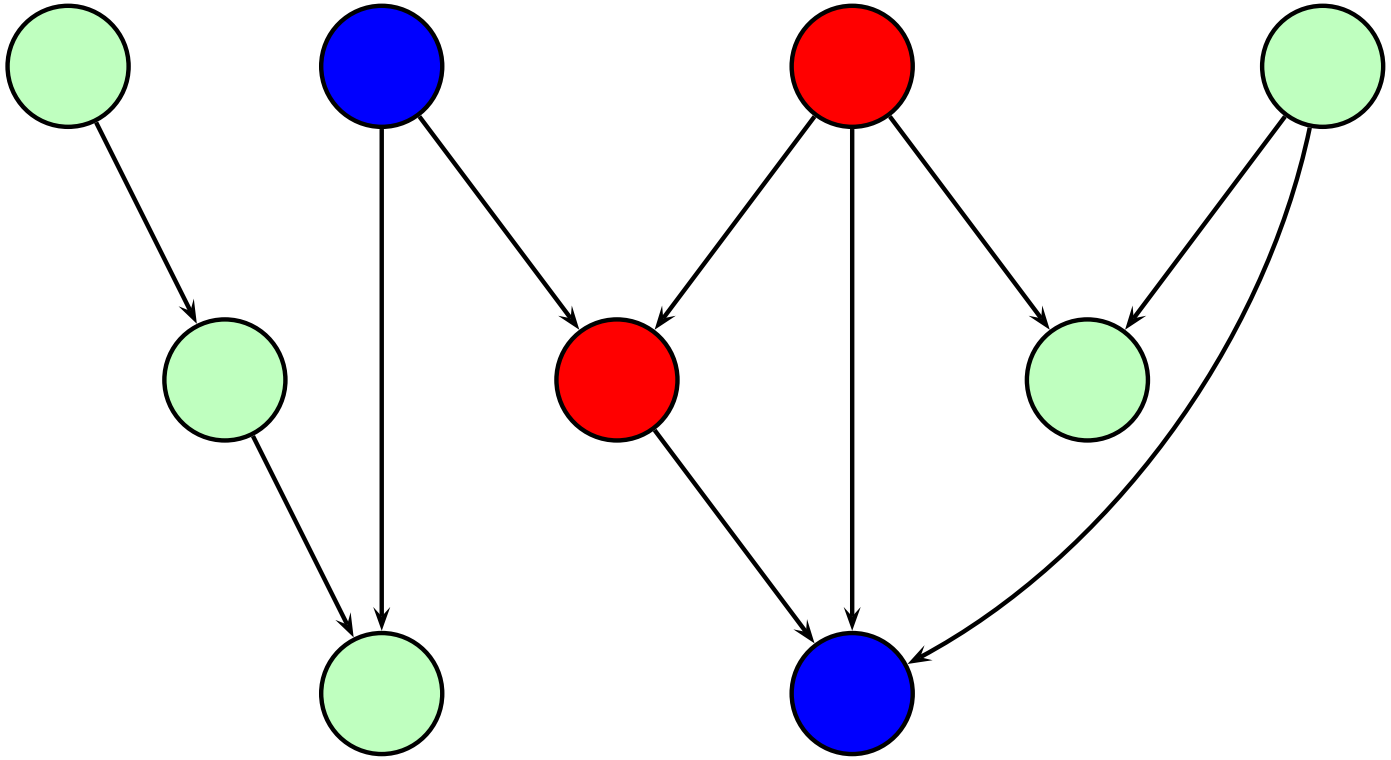
Tested variables

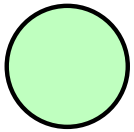


Observed Node

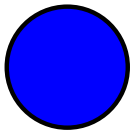
Dependent variables (Red obs.)

Independence Examples

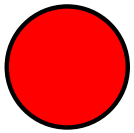




Rest of variables

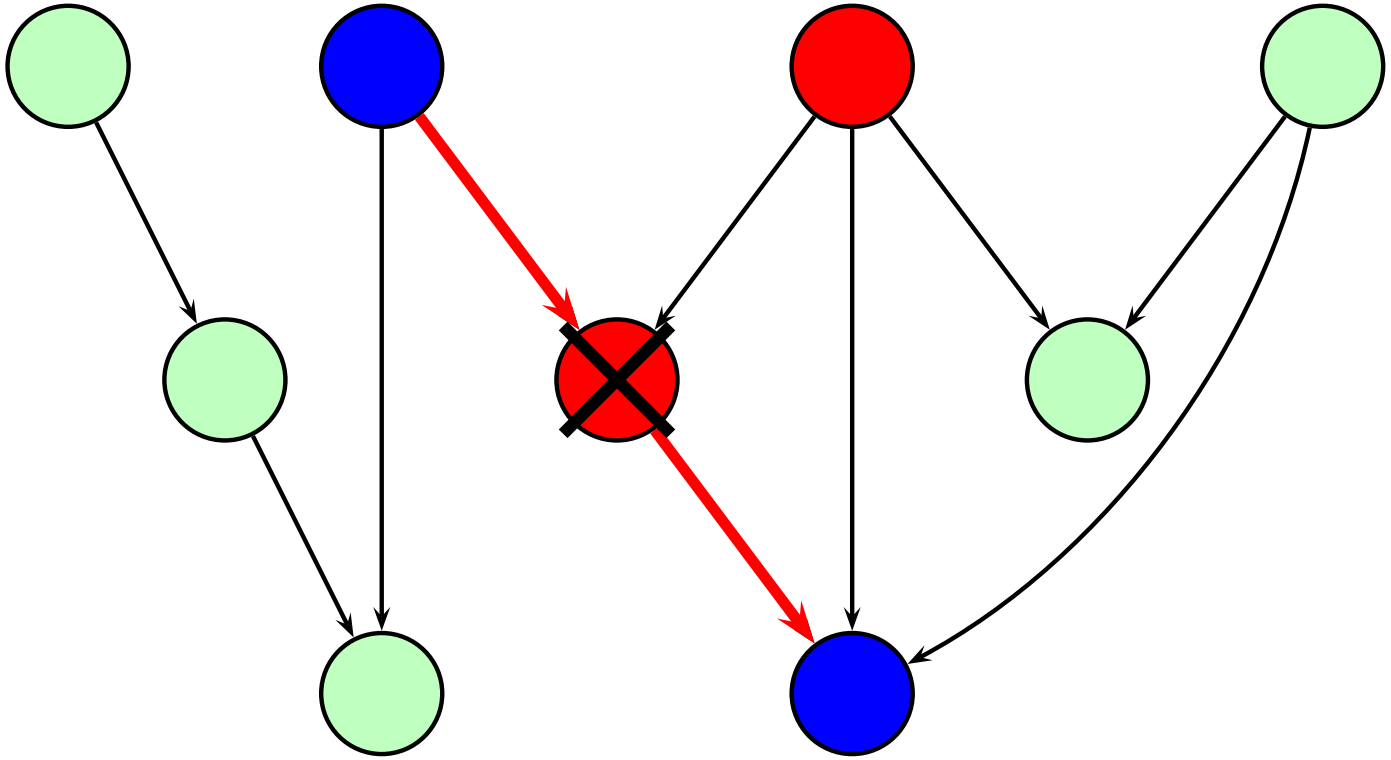


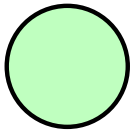
Tested variables



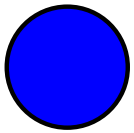
Observed Node

Independence Examples

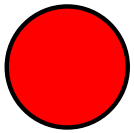




Rest of variables



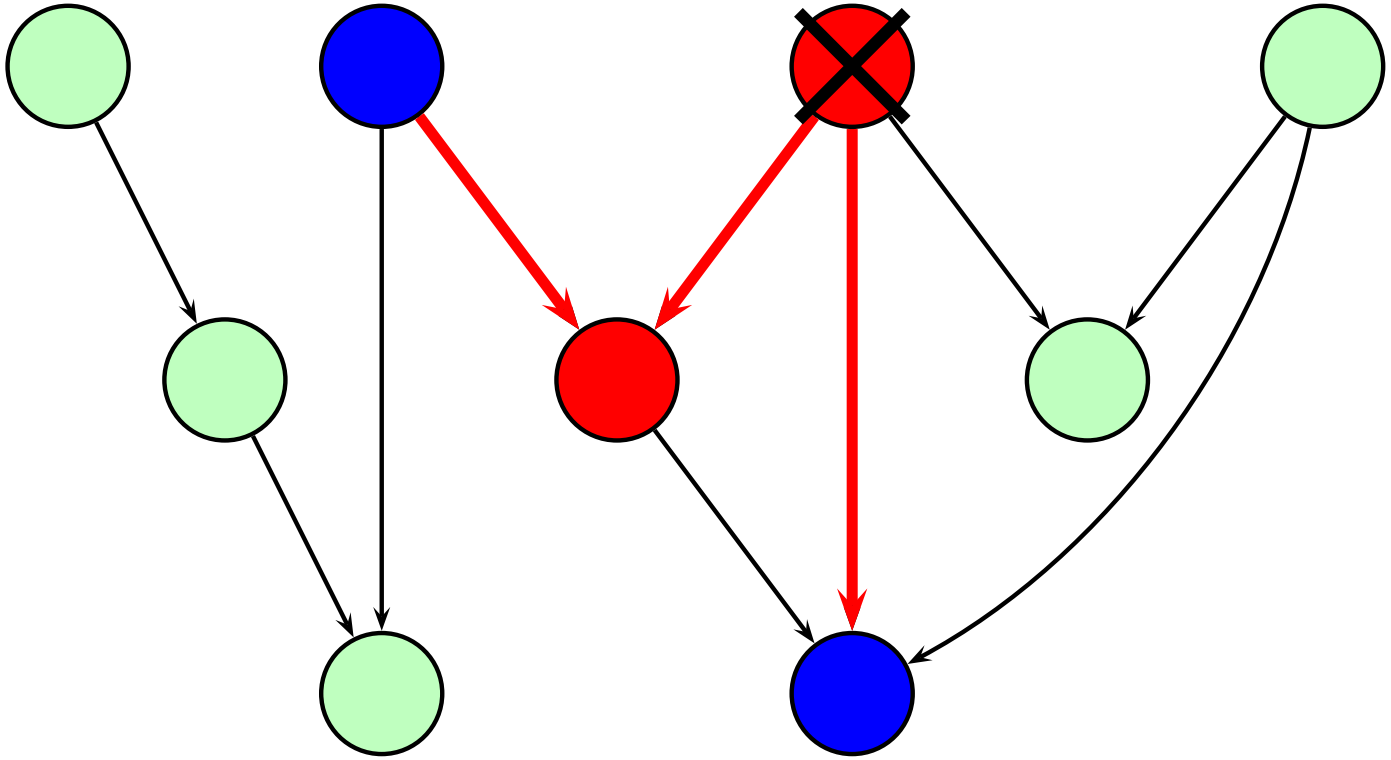
Tested variables



Observed Node

First path blocked

Independence Examples



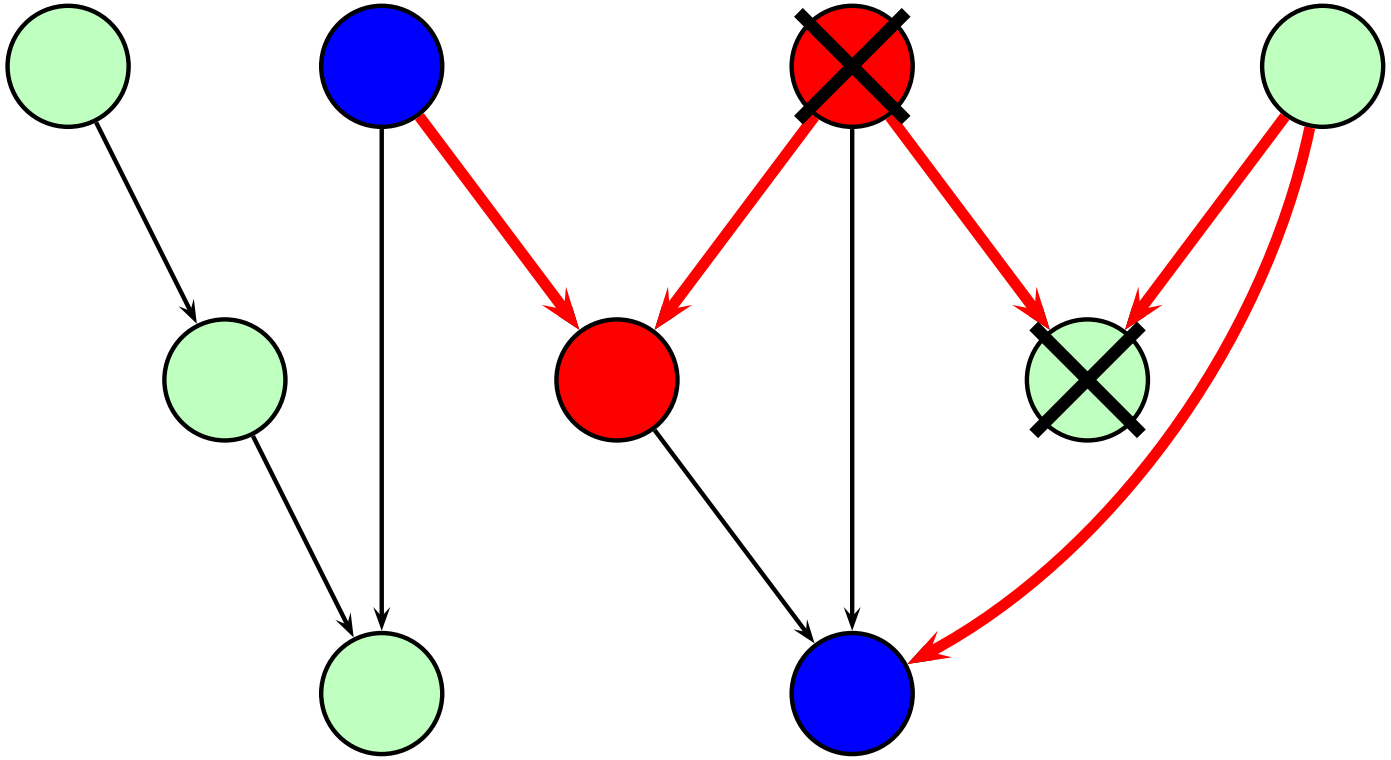
Rest of variables

Tested variables

Observed Node

Second path blocked

Independence Examples



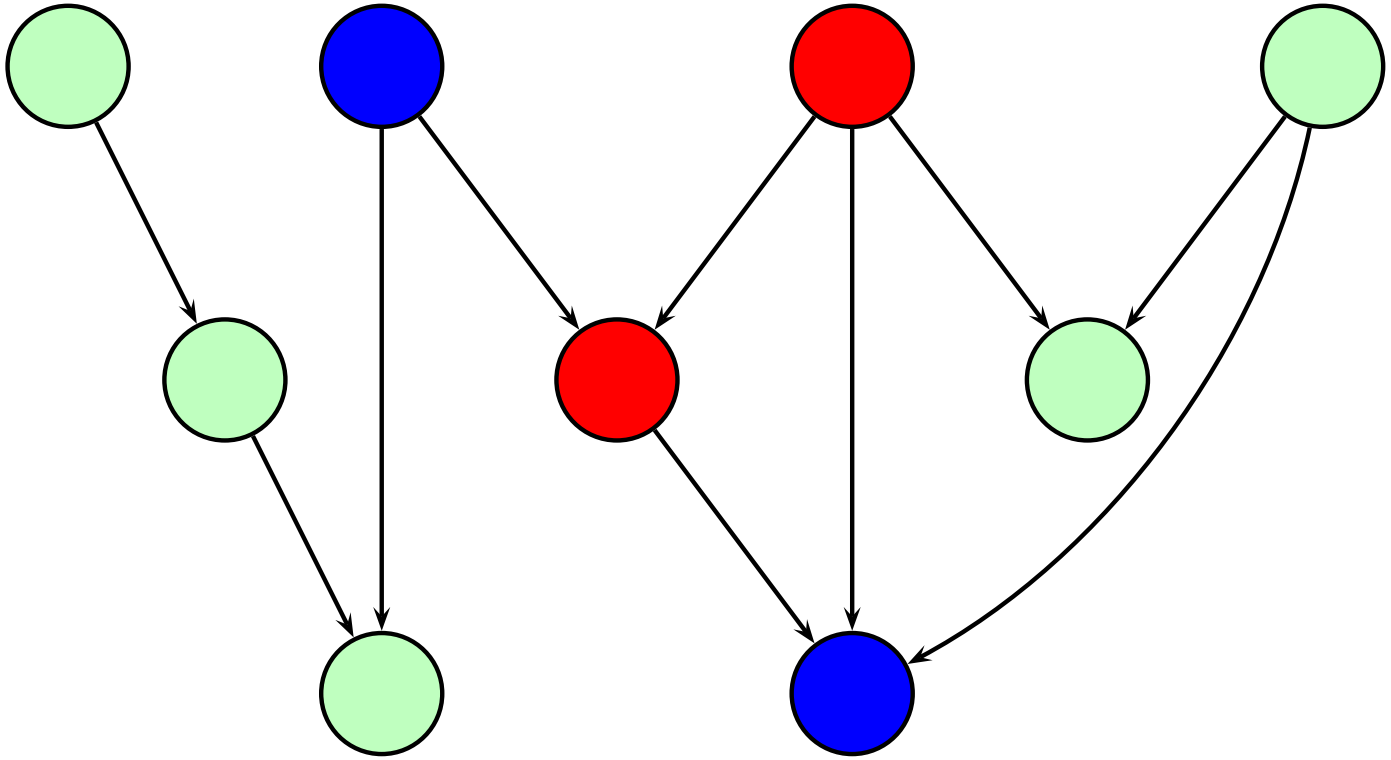
Rest of variables

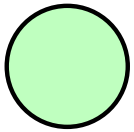
Tested variables

Observed Node

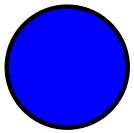
Third path blocked

Independence Examples

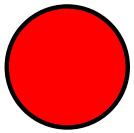




Rest of variables



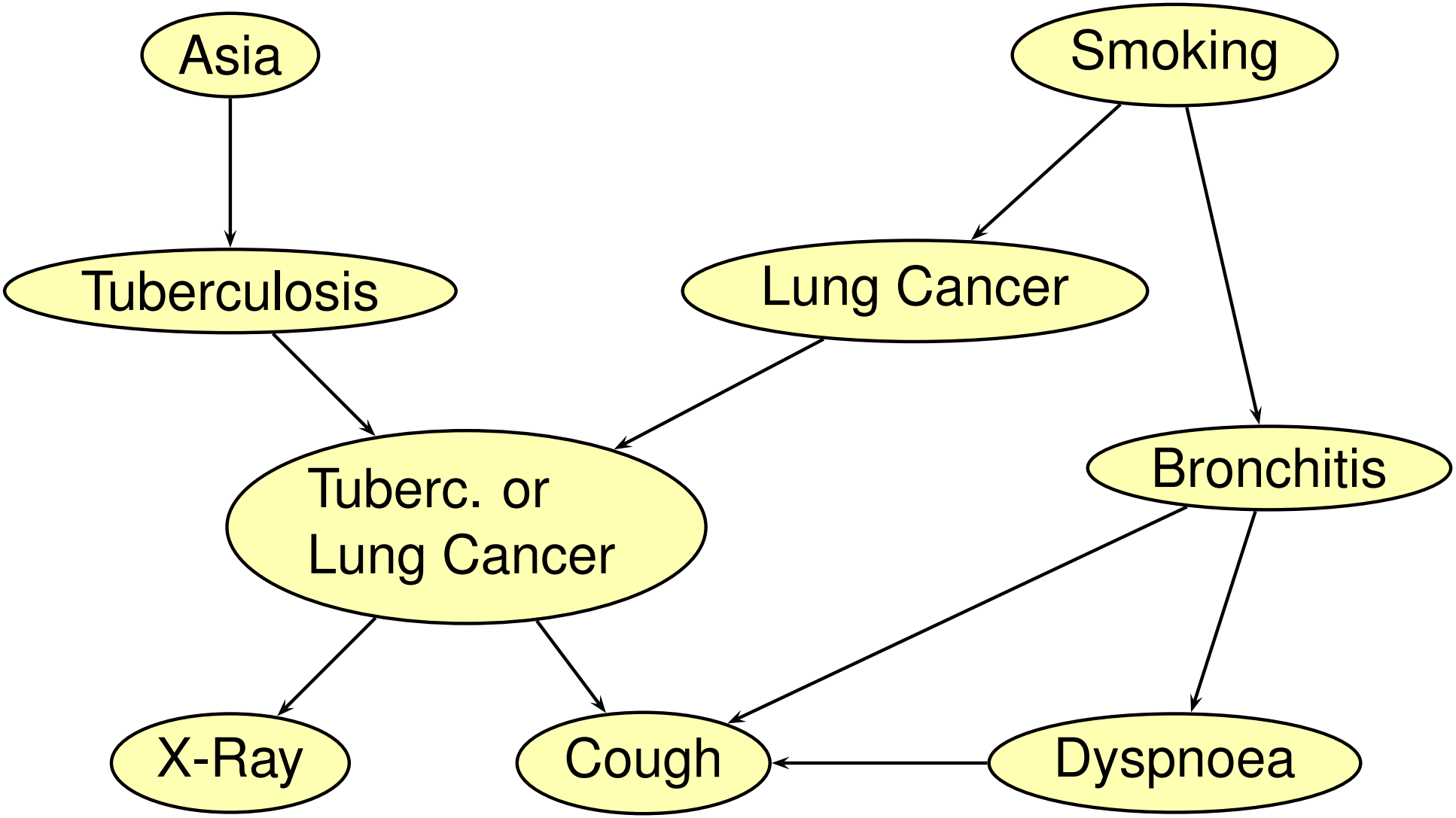
Tested variables



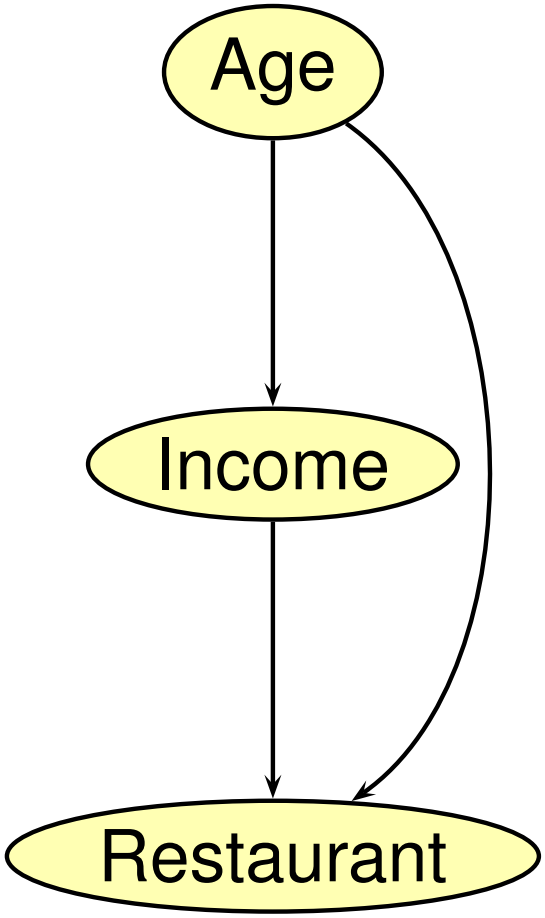
Observed Node

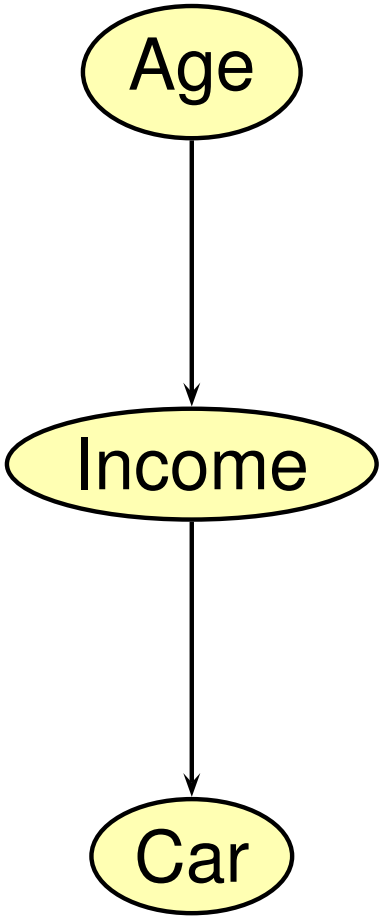
Independent Variables (Red obs.)

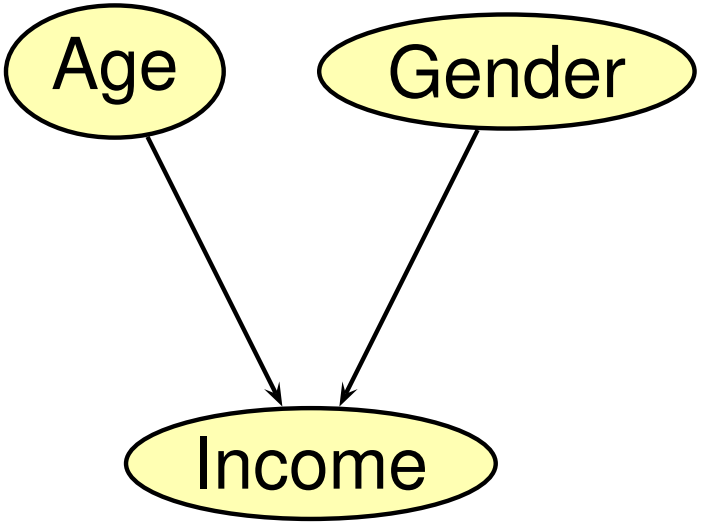
Asia Network



3 Variables Examples







Decomposition Theorem

Given a Bayesian network with variables $\mathbf{X}$, then we can have the following decomposition of the joint probability distribution:

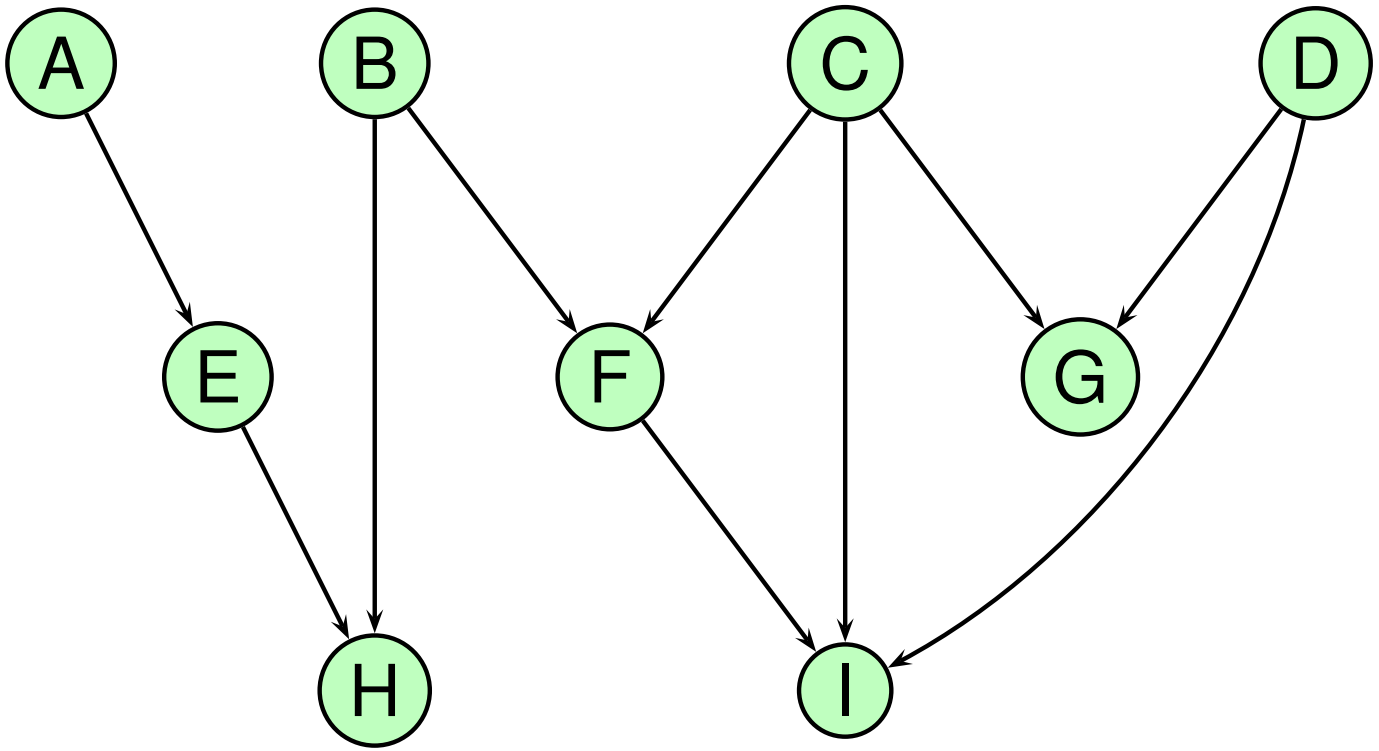
$$p(\mathbf{x}) = \prod_{\mathbf{y} \in \mathbf{x}} \mathbf{p}(\mathbf{y}|\mathbf{pa}(\mathbf{y}))$$

where $pa(Y)$ is the set of parents of variable Y.

Consequence: To specify a joint probability distribution we only have to give a conditional probability distribution for each node given its parents.

In the case of a root node, we give its marginal distribution.

Example



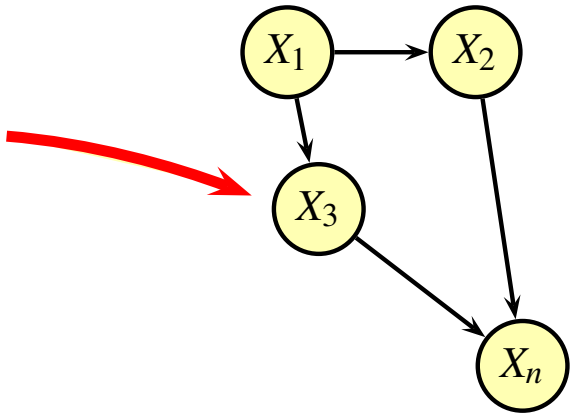
$$p(a,b,c,d,e,f,g,h,i) =$$

$$p(a).p(b).p(c).p(d).p(e|a).p(f|b,c).p(g|c,d).p(h|b,e).p(i|c,d,f)$$

Learning

Learning in Bayesian networks can be defined as the process of inducing a model from a database.

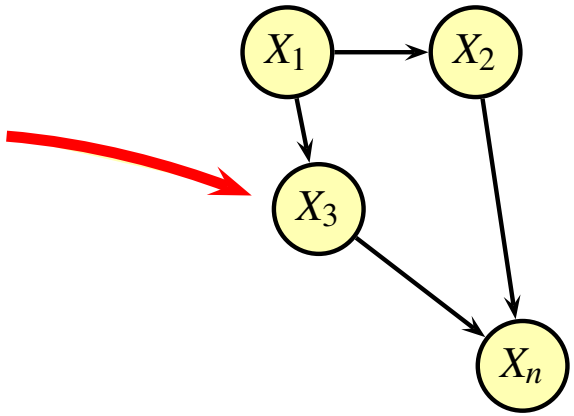
X_1	X_2	$\ldots$	X_n
x_1^1	x_2^1	$\ldots$	x_n^1
x_1^2	x_2^2	$\ldots$	x_n^2
x_1^3	x_2^3	$\ldots$	x_n^3
x_1^4	x_2^4	$\ldots$	x_n^4



Learning

Learning in Bayesian networks can be defined as the process of inducing a model from a database.

X_1	X_2	$\ldots$	X_n
x_1^1	x_2^1	$\ldots$	x_n^1
x_1^2	x_2^2	$\ldots$	x_n^2
x_1^3	x_2^3	$\ldots$	x_n^3
x_1^4	x_2^4	$\ldots$	x_n^4



Learning = Inducing a graph + Estimating parameters

Motivation

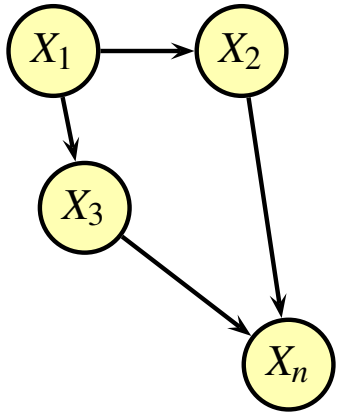
- Building probabilistic networks from experts can be difficult
  - Difficulties of experts to know the meaning of probability and dependence graphs
  - Very large domains. Only groups of experts can cover it.
  - Expensive and time consuming.
  - No expert knowledge available: we want to discover
- Learning with Bayesian networks offers a wide range of possibilities
  - It can integrate expert knowledge when available
  - It can be classification oriented
  - It is possible to discover causality
  - It is cheap

Modalities

- Known structure

X_1	X_2	$\ldots$	X_n
x_1^1	x_2^1	$\ldots$	x_n^1
x_1^2	x_2^2	$\ldots$	x_n^2
x_1^3	x_2^3	$\ldots$	x_n^3
x_1^4	x_2^4	$\ldots$	x_n^4

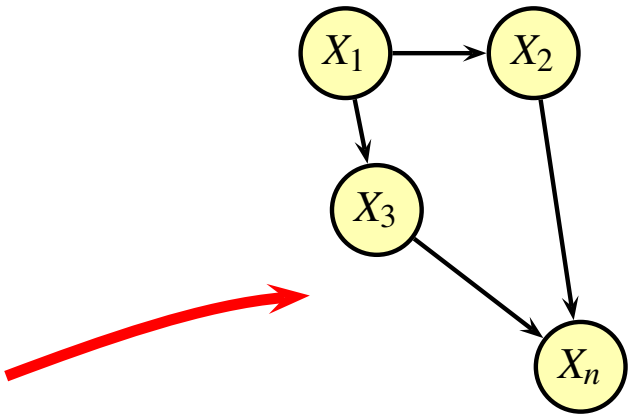
$$p(x_2|x_1), p(x_3|x_1), p(x_n|x_2, x_3)$$



Modalities

- Unknown structure

X_1	X_2	$\ldots$	X_n
x_1^1	x_2^1	$\ldots$	x_n^1
x_1^2	x_2^2	$\ldots$	x_n^2
x_1^3	x_2^3	$\ldots$	x_n^3
x_1^4	x_2^4	$\ldots$	x_n^4

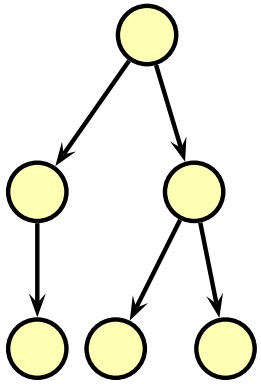


$$p(x_2|x_1), p(x_3|x_1), p(x_n|x_2, x_3)$$

Modalities: Partial Knowledge

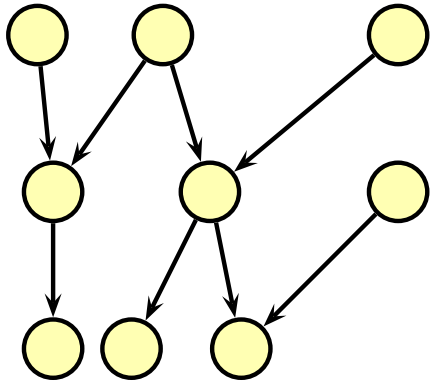
- We know the presence (absence) of links
- We know a partial order relation: X_1 goes before X_n or X_1 can not be a descendant of X_n.
- We know some independence relationships (or that some independence relationships are not true).
- We know an 'a priori' probability distribution on the set of possible graphs.

Restricted Set of Structures



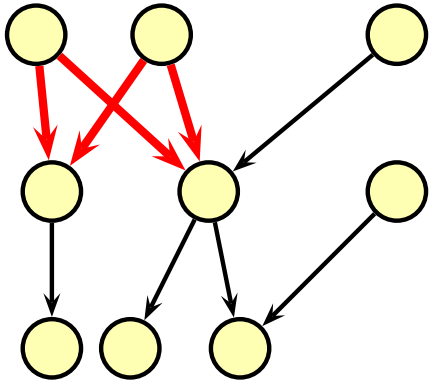
Tree

One node, at most, one parent.



Hipertree

No undirected cycles (loops).



Simple Graphs

Every loop has, at least to *head to head* nodes.

Incomplete Data

- Incomplete data when using a Bayesian network is not a problem (*Think in classical models when data are missing*).
- Incomplete data when learning makes everything more difficult (*there is an important hypothesis that is not verified in that case*), even with Missing at Random hypothesis.
  We can have the two modalities: estimating parameters or estimating parameters and structure.

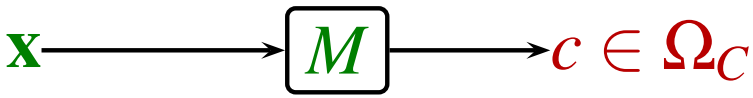
X_1	X_2	$\ldots$	X_n
?	x_2^1	$\ldots$	x_n^1
x_1^2	?	$\ldots$	?
x_1^3	?	$\ldots$	x_n^3
x_1^4	x_2^4	$\ldots$	x_n^4

The Supervised Classification Problem

We assume a set of variables or attributes $\mathbf{X} = (X_1, \ldots, X_n)$.
Each variable X_i will take values on a finite set U_{X_i}.
We have a class variable C, with values in U_C.
We have a database of values for these variables:

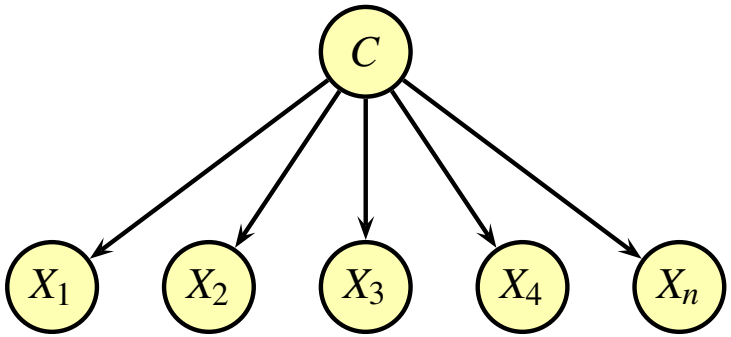
X_1	X_2	$\ldots$	X_n	C
x_1^1	x_2^1	$\ldots$	x_n^1	c_1
x_1^2	x_2^2	$\ldots$	x_n^2	c_2
x_1^3	x_2^3	$\ldots$	x_n^3	c_3
x_1^4	x_2^4	$\ldots$	x_n^4	c_4

We want to induce a model M such that if $\mathbf{x}$ is a value of $\mathbf{X}$.



Naïve Bayes Classifiers

Predictive variables are conditionally independent given the variable class

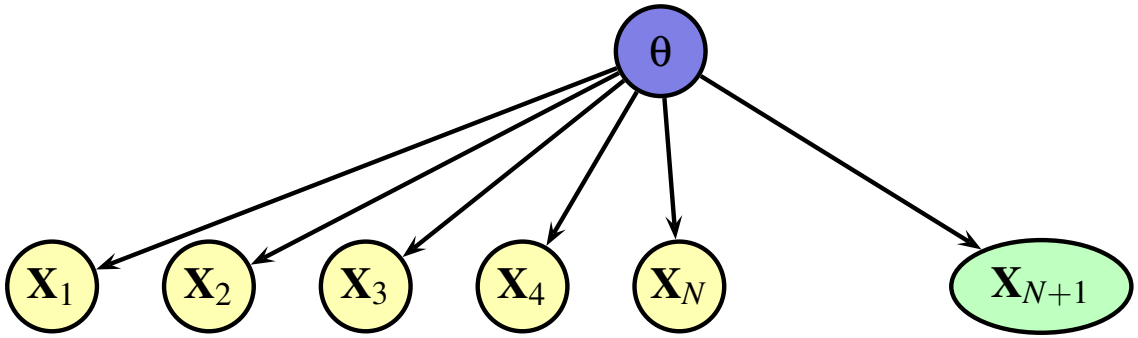


The joint probability factorizes as:

$$P(C = c).\prod_{i=1}^{n} P(X_i = x_i | C = c)$$

Estimating Parameters: Bayesian Approach

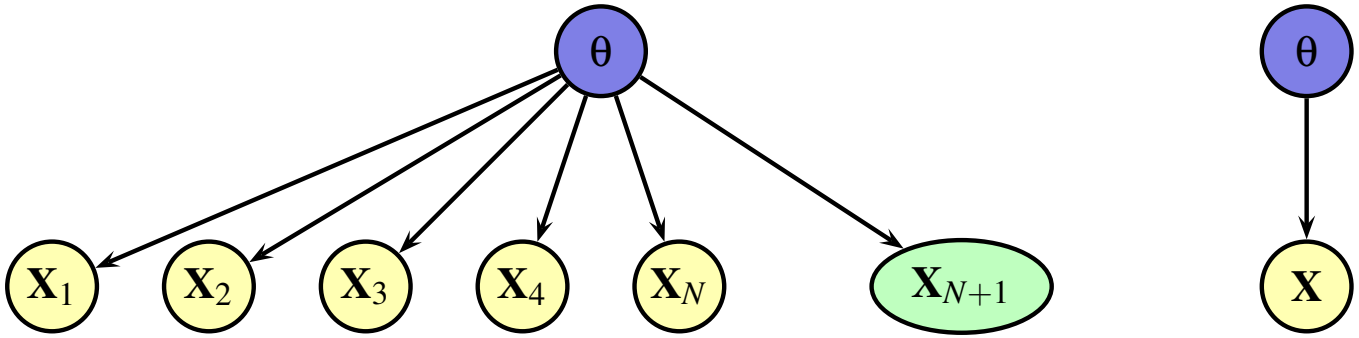
Bayesian Network Representation



We have an 'a priori' information about the parameter.
All the conditional distributions are the same.
We assume that we have observed $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and we want to compute the conditional information about $\mathbf{X}_{N+1}$.

Estimating Parameters: Bayesian Approach

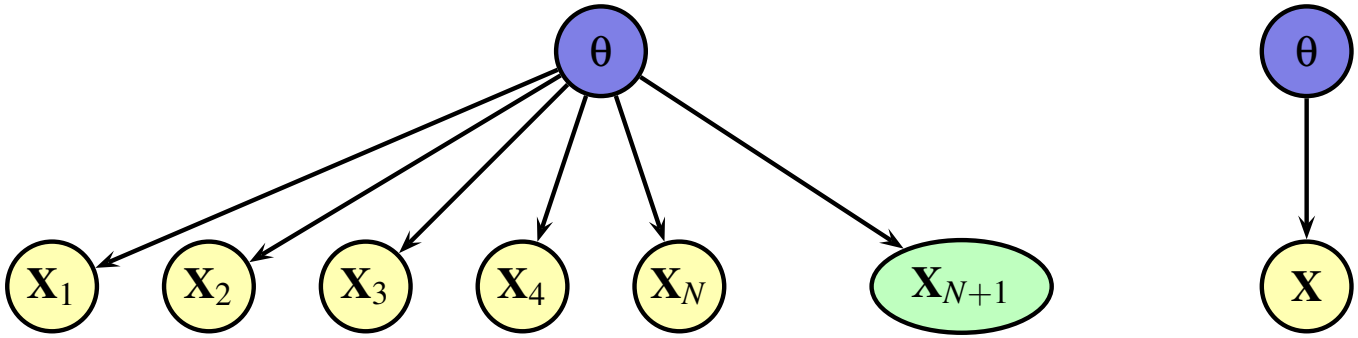
Bayesian Network Representation



We have an 'a priori' information about the parameter.
All the conditional distributions are the same.
We assume that we have observed $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and we want to compute the conditional information about $\mathbf{X}_{N+1}$.

Estimating Parameters: Bayesian Approach

Bayesian Network Representation



We have an 'a priori' information about the parameter.
All the conditional distributions are the same.
We assume that we have observed $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and we want to compute the conditional information about $\mathbf{X}_{N+1}$.
We assume that θ follows a Dirichlet distribution, and then we compute the 'a posteriori' expectation of the parameter given the sample

Example

Imagine that we have an urn with balls of different colors: red (R), blue (B), and green (G); but on an unknown quantity.

Assume that we picked up balls with replacement, with the following sequence: (B, B, R, R, B).

If we assume a Dirichlet 'a priori' distribution with parameters: $D(1,1,1)$, then the estimated frequencies for red, blue, and green are:
$(3/8, 4/8, 1/8)$

Observe, as green has a positive probability, even it never appears in the sequence.

A Bayesian Network

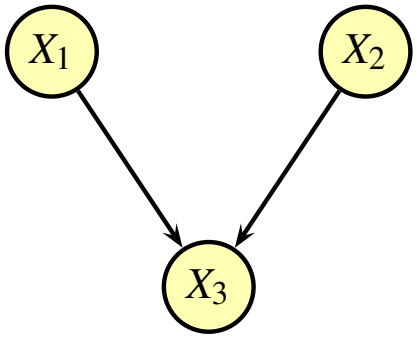
- The basic approach is to apply the Bayesian approach with Dirichlet 'a priori' distributions for each conditional probability distribution.
- In each case, we should only consider the part of the database that is compatible with the values of the parents to which we are conditioning.
- We have to be conscious of the basic hypothesis allowing us to do it, and when it does not make sense to apply them.
- We can have problems when selecting the equivalent sample size of the Dirichlet distributions.

Parametrization

- For each variable X_i let $x_i^1, \ldots, x_i^{r_i}$ the set of possible values where r_i is the number of possible values.
- The number of configurations for the parents of X_i will be denoted by q_i. The configuration number j will be denoted by pa_j^i.
- The parameters necessary to specify a Bayesian network are
$$\theta_{ijk} = P(x_i^k | pa_j^i), \quad i = 1, \ldots, n, \quad j = 1, \ldots, q_i, \quad k = 1, \ldots, r_i$$
- θ_{ij} will denote the vector of multinomial probabilities $(\theta_{ij1}, \ldots, \theta_{ijr_i})$.

Example

Assume the following network where all the variables are binary

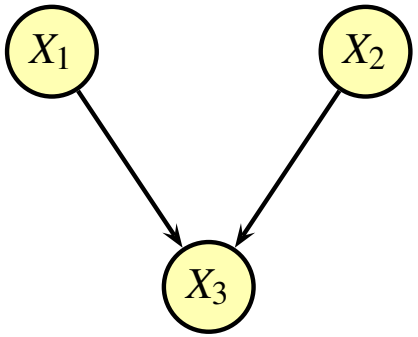


Example

Assume the following network where all the variables are binary

$\theta_{111} = P(X_1 = 0)$

$\theta_{112} = P(X_1 = 1)$



$\theta_{211} = P(X_2 = 0)$

$\theta_{212} = P(X_2 = 1)$

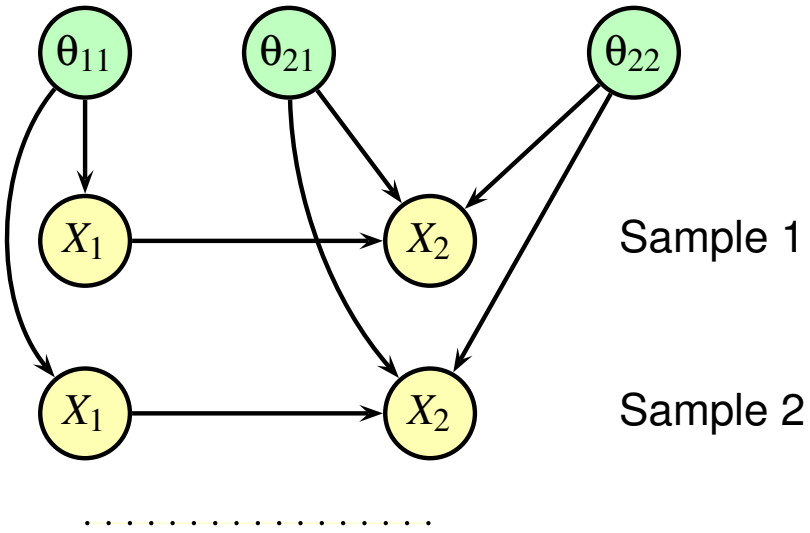
$$\theta_{311} = P(X_3 = 0|0,0), \quad \theta_{312} = P(X_3 = 1|0,0)$$
$$\theta_{321} = P(X_3 = 0|0,1), \quad \theta_{322} = P(X_3 = 1|0,1)$$
$$\theta_{331} = P(X_3 = 0|1,0), \quad \theta_{332} = P(X_3 = 1|1,0)$$
$$\theta_{341} = P(X_3 = 0|1,1), \quad \theta_{342} = P(X_3 = 1|1,1)$$

Parameter Independence

A basic hypothesis that is convenient and in some situations real is that the parameters distributions are independent.

$$p(\theta) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} p(\theta_{ij})$$

where θ denotes the vector of all the parameters.
Graphically with two binary variables:



Main Result

If we observe all the values of the variables in the sample, then the 'a posteriori' distributions of the parameters θ_{ij} are also independent.

$$P(\theta|D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} P(\theta_{ij}|D)$$

The Consequence:

We can apply the Dirichlet model to each conditional probability on an independent way and update each of them in an independent way.

The Dirichlet Updating

- Under conditions of independence
- If the distribution of the parameters θ_{ij} is a Dirichlet $D(\alpha_{ij1}, \ldots, \alpha_{ijr_i})$
- If in the database, there are N_{ijk} cases in which the variable X_i takes the value x_k^i, and the parents of this variable are in configuration pa_j^i
- Then the 'a posteriori' distributions for the parameters θ_{ij} are

$$D(\alpha_{ij1} + N_{ij1}, \ldots, \alpha_{ijr_i} + N_{ijr_i})$$

Example

Assume two variables:

X_1: Smoking; values 1 (yes) 2 (no)

X_2: Lung Cancer; values 1 (yes) 2 (no)

Imagine the following sequence of observations and network:

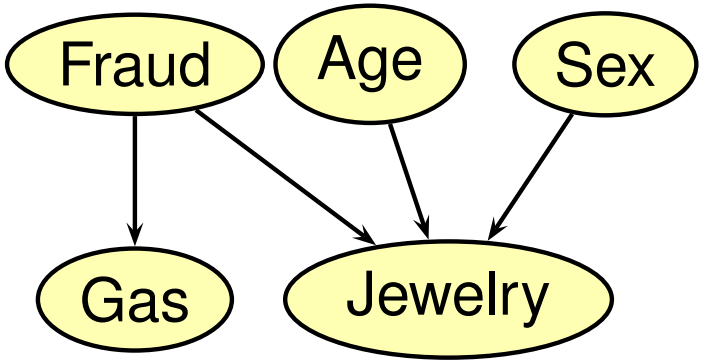
	$c.1$	$c.2$	$c.3$	$c.4$	$c.5$	$c.6$	$c.7$	$c.8$
X_1	1	1	2	2	2	2	1	2
X_2	2	1	1	2	1	1	2	2

Assuming that all the parameters have an 'a priori' $D(1,1)$, the estimations are:

$$\theta^*_{111} = 0.4, \quad \theta^*_{112} = 0.6,$$

$$\theta^*_{211} = 0.4, \quad \theta^*_{212} = 0.6, \quad \theta^*_{221} = 4/7, \quad \theta^*_{222} = 3/7$$

Equivalent Sample Sizes



If the equivalent sample size (the sum of the parameters of the Dirichlet distribution) represents the strength of our past experience, Can we claim the same same experience when estimating the probabilities of sex that when estimating the probabilities of using credit car in a jewelry in a fraudulent way, for males above 50?

This is a difficult problem.

Consistency

A way of getting consistency and equivalent results for equivalent Bayesian networks (networks representing exactly the same independence relationships) is the following:

- Assume a global equivalence sample size for all the network S.
- For each distribution of parameters θ_{ij} consider an equivalent sample size of $S_{ij} = S/(q_i)$ and uniform parameters $\alpha_{ijk} = S/(q_i.r_i)$.

If in the smoking-cancer example, we use for Smoking variable a $D(1,1)$, then the global sample size is $S = 2$, and for X_2, as there are two possible configurations for the parents, then this should be divided by two for the conditional probabilities and we should assume $D(0.5,0.5)$ for each one of them.

I am not sure that this is a good solution: if a variable has too many parents, the parameter estimations will approach the maximum likelihood ones

Imprecise Interval Estimation

- It is clear that we can apply the Imprecise Dirichlet Model to estimate the probabilities instead of obtaining a precise estimation.
- The application is not immediate. We have physical probabilities relating the variables of the model. The sample is obtained by repetition independence.
- We are going to estimate a joint credal set M for all the variables $\mathbf{X}$. All the probabilities in this credal set should verify classical probabilistic independence relationships expressed by the graph.
- The basic property for epistemic independence should be verified, but it will not be necessary to get the epistemic natural extension.

Two Main Approaches

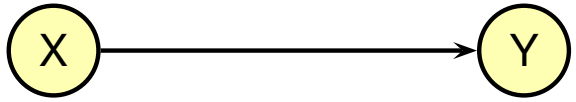
- The Local Approach.- Assume an IDM for each conditional probability distribution. For the variable X_i conditioned to the configuration pa_j^i, θ_{ij} follow an IDM with sample size S_{ij}. The estimation for $P(x_k^i|pa_j^i)$ is given by

$$\left[\frac{N_{ijk}}{N_{ijk}+S_{ij}}, \frac{N_{ijk}+S_{ij}}{N_{ijk}+S_{ij}}\right]$$

- The Global Approach.- Assume a global IDM for the joint probability distribution for all the variables with sample size S. Each θ_{ij} is an IDM with equivalent sample size S, but we will not have strong independence in the estimated credal sets: the conditional distributions can depend of the marginal ones.

The Local Approach

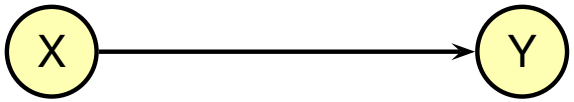
- The main problem is how to determine the local equivalent sample size: S_{ij}.
- If all the equivalent sample sizes are equal $S_{ij} = S$, this can produce that leave nodes have a very imprecise marginal.



The conditional probabilities will have small sample sizes, and will produce wider intervals than the ones for X, as the sample is divided between all the conditional probability distributions. When computing the marginal on Y we will *add* the imprecision of X and the conditionals obtaining a lot of imprecision about Y.

- If we do as in the Bayesian model $S_{ij} = S/q_i$ (where q_i is the number of parents configurations) then we will obtain too precise conditional distributions with small samples.
- I would consider an uniform sample size.

The Global Approach



The interval probabilities for $P(x)$ are

$$\left[\frac{N_x}{N+S}, \frac{N_x+S}{N+S}\right]$$

The interval probabilities for $P(y|x)$ are

$$\left[\frac{N_{x,y}}{N_x+S}, \frac{N_{x,y}+S}{N_x+S}\right]$$

However, we do not have strong independence. As the parameters for the Dirichlet distributions have to be consistent, we have that $S_x = \sum_y \alpha_{x,y} = \alpha_x$.
So to have the full interval for $P(y|x)$ we need that $P(x)$ is the upper limit of its interval. If the probability of $P(x)$ goes down, then the intervals for $P(y|x)$ are more precise.
Computation is more difficult. Only in some particular cases, Walley has unpublished results about computation.

Structure Learning

There are two basic approaches:

- Independence tests
- Score + Search procedures

PC Algorithm: Basic Hypothesis

Spirtes, Glymour, Scheines (1993) *Causation, Prediction, and Search*

- The independence relationships have a perfect representation by a DAG
- We have a very large database
- Statistical tests have no errors

Under these conditions, the algorithm will discover and equivalent Bayesian network.

Statistical Tests

The algorithm is based in asking for the true of independence relationships of the form:

$$I(X, Y|\mathbf{Z})$$

where $\mathbf{Z}$ is a subset of variables.
It can work with any source providing this kind of information.
It we have a database, this is answered by means of statistical tests of independence.

Independence Test: Conditional MI

Given three variables X, Y, Z the Conditional Mutual Information of X and Y given Z is defined as

$$MI(X,Y|Z) = \sum_{z} P(z) \sum_{x,y} P(x,y|z) \log \left(\frac{P(x,y|z)}{P(x|z).P(y|z)} \right)$$

It verifies the $CE(X,Y|Z) = H(X|Z) - H(X|Y,Z)$
I can be analogously defined when Z is a set of variables.
The Empirical Mutual Information is the mutual information computed from a database by considering $P = \hat{P}$, the maximum likelihood estimation in the database.

Independence Test

To test whether X and Y are conditional independent given $\mathbf{Z}$, we compute the empirical mutual information $MI(X,Y|\mathbf{Z})$.
The statistic used for the test is G^2 which is $2NCE(X,Y|\mathbf{Z})$ where N is the sample size.
It is known that, under the independence assumption, G^2 follows a χ^2 distribution with degrees of freedom equal to:

$$(r_X - 1)(r_Y - 1)\prod_{Z \in \mathbf{Z}} r_Z$$

where r_W is the number of values of variable W.
It is possible to decrease in one the number of degrees of freedom for each configuration of values of the variables that does not appear in the database.

Examples

Determine whether there is independence for variables X_1 and X_2 (with values 1 and 2) according to the following databases,

Case	1	2	3	4	5	6	7	8
X_1	1	1	2	2	2	2	1	2
X_2	2	1	1	2	1	1	2	2

can not be rejected with p-value 0.47. Independence

Case	1	2	3	4	5	6	7	8
X_1	1	1	1	1	2	2	2	2
X_2	1	1	1	1	2	2	2	2

Independence can not be rejected with p-value 0.001. Dependence

Sample Size

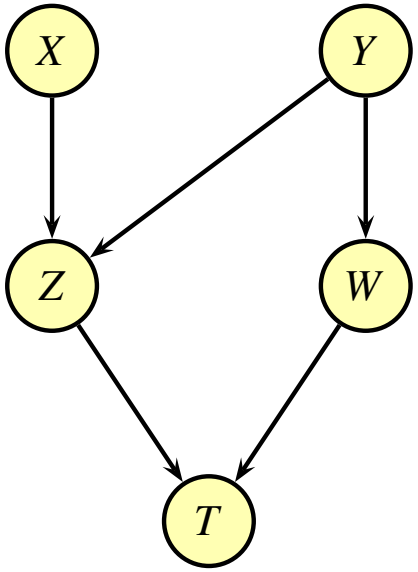
- The significance level is usually 0.01 or 0.05 (greater values are better).
- It assumes dependence when the probability of a χ^2 with the corresponding degrees of freedom is greater than G^2 is smaller than the confidence level.
- In case of accepting independence it does not mean that data support independence, but that there is no evidence in the data against it.
- When sample size or the conditional set is big, then the possibility of rejecting the null hypothesis is lower and independence will be assumed: Lack of support implies independence

The Algorithm Structure

1. Find a graph pattern (gp): an undirected graph
2. Find some head to head links by testing independences
3. Orient the rest of links without producing cycles

Remark: There is some degree of arbitrariness and sometimes, though independences can be represented by a DAG the direction of the arrows is counterintuitive with causality.

Graph Pattern: The Basic Condition



Two nodes, X and Y, are connected if and only if there is no subset S_{XY} of the set of vertices V such that $I(X,Y|S_{X,Y})$.

We could try to discover the graph pattern following this criterion, but it will be inefficient (too many tests) and inaccurate (conditioning to many variables).

Finding the Graph Pattern

V is the set of nodes, and every independence relationships can be tested. Each node has a set of adjacent nodes ADJ_X.

1. Start with a complete undirected graph gp
2. $i = 0$
3. **Repeat**
   4. **For each** $X \in V$
      5. **For each** $Y \in ADJ_X$
         6. Determine if there is $S \subseteq ADJ_X - \{Y\}$ with $|S| = i$ and $I(X, Y|S)$
         7. **If** this set exists
            8. Make $S_{XY} = S$
            9. Remove $X - Y$ link from gp
   10. $i = i + 1$
11. **Until** $|ADJ_X| \leq i, \quad \forall X$

Finding Head-to-Head Links

1. **For each** uncoupled meeting $X - Z - Y$
   2. **If** $Z \notin S_{XY}$
      3. Orient $X - Z - Y$ as $X \rightarrow Z \leftarrow Y$

If a variable is connected with other two variables and it is not in the separator of them, then the arrows have to be head-to-head.

More Orientations

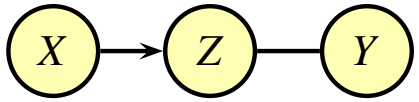
The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$

More Orientations

The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

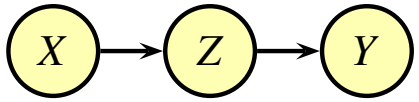
1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$



More Orientations

The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

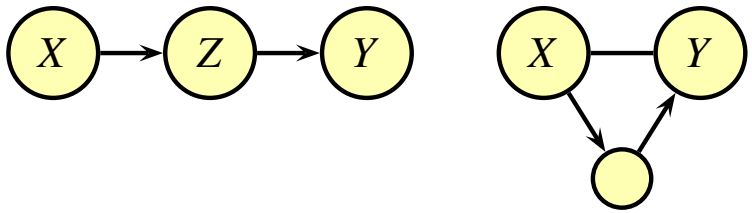
1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$



More Orientations

The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

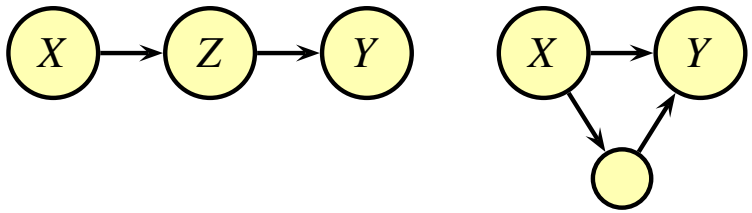
1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$



More Orientations

The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

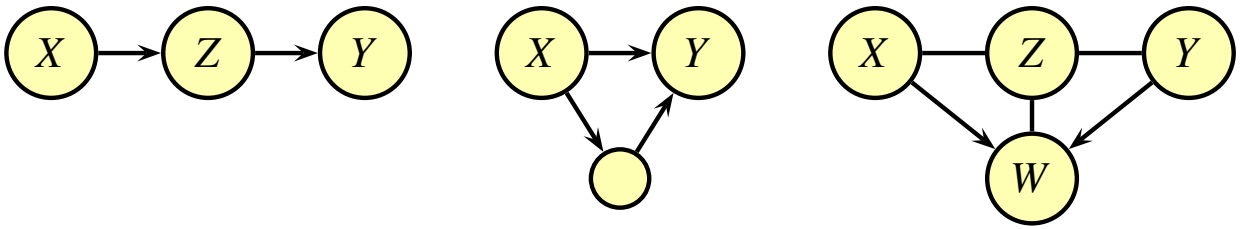
1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$



More Orientations

The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

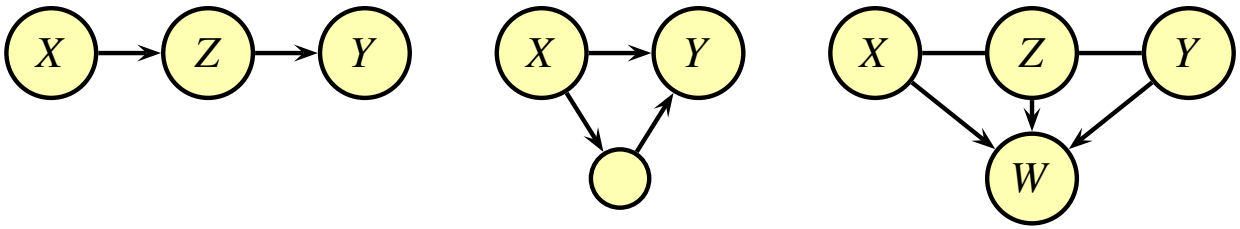
1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$



More Orientations

The basic idea is that no new head-to-head links are created and that the DAG condition is preserved.

1. **While** no more edges can be oriented
   2. **For** each uncoupled meeting $X \rightarrow Z - Y$
      3. Orient $Z - Y$ as $Z \rightarrow Y$
   4. **For** each $X - Z$ such that there is a directed path from X to Z
      5. Orient $X - Y$ as $X \rightarrow Y$
   6. **For** each uncoupled meeting $X - Z - Y$ such that $X \rightarrow W, Y \rightarrow W, Z - W$
      7. Orient $Z - W$ as $Z \rightarrow YW$



Imprecise Independence Tests

- The independence scores we shaw can be extended to conditional independence tests which is the only thing that has to be done in PC algorithm.
- Entropy based score. To test $I(X,Y|\mathbf{Z})$, we compute:

$$IND = \sum_{\mathbf{z},x} \hat{P}(\mathbf{z})\overline{H}(\mathcal{M}_Y^{\mathbf{Z}=\mathbf{z}})$$

where $\mathcal{M}_Y^{\mathbf{Z}=\mathbf{z}}$ is the credal set about Y estimated with the part of the sample for which $\mathbf{Z}=\mathbf{z}$, with the IDM and S.

$$DEP = \sum_{\mathbf{z},x} \hat{P}(\mathbf{z},x)\overline{H}(\mathcal{M}_Y^{\mathbf{Z}=\mathbf{z},X=x})$$

where $\mathcal{M}_Y^{\mathbf{Z}=\mathbf{z},X=x}$ is the credal set about Y estimated with the part of the sample for which $\mathbf{Z}=\mathbf{z}$ and $X=x$, with the IDM and S.

- Decide for dependence if $DEP > IND$ and for independence otherwise.

Independence Tests: Imprecise Dirichlet

- Compute the sample size: $S_{\mathbf{Z},X} = S/q_{X,\mathbf{Z}}$, where S is the global sample size and $q_{X,\mathbf{Z}}$ is the number of joint configurations for variables X and $\mathbf{Z}$.
- Assign Dirichlet parameters, $\alpha_{\mathbf{z},x,y}$ with the following criterion: all the parameters should be at least $S_{\mathbf{Z},X}/(2.r_Y)$. The rest of $S_{\mathbf{Z},X}$ is equally distributed between the values of Y with lowest value of frequency: $N_{\mathbf{z},x,y}$.
- Compute Dirichlet parameters: $\alpha_{\mathbf{z},y} = \sum_x \alpha_{\mathbf{z},x,y}, S_{\mathbf{Z}} = \sum_y \alpha_{\mathbf{z},y}$.
- Compute the scores for the conditional probability of Y: $DEP - IND$, where

$$DEP = \prod_{\mathbf{z},x} \frac{\Gamma(S_{\mathbf{Z},X})}{\Gamma(N_{\mathbf{z},x}+S_{\mathbf{Z},X})} \left(\prod_y \frac{\Gamma(N_{\mathbf{z},x,y}+\alpha_{\mathbf{z},x,y})}{\Gamma(\alpha_{\mathbf{z},x,y})} \right)$$

$$IND = \prod_{\mathbf{z}} \frac{\Gamma(S_{\mathbf{Z}})}{\Gamma(N_{\mathbf{z}}+S_{\mathbf{Z}})} \left(\prod_y \frac{\Gamma(N_{\mathbf{z},y}+\alpha_{\mathbf{z},y})}{\Gamma(\alpha_{\mathbf{z},y})} \right)$$

- Decide for dependence if $DEP - IND > 0$

Score+Search: Pros and Cons

- Can make compromises (ambiguous regions)
- Well justified
- Can take into account simplifications of conditional probabilities.
- It is computationally difficult and then the solutions are almost always approximate.

The Bayesian Score

Applying Bayes Rule:

$$P(G|D) = \frac{P(D|G).P(G)}{P(D)}$$

where

- $P(D)$ is the probability of data, which is constant.
- $P(G)$ is the 'a priori' probability of graph, which can be uniform or incorporate any other information that the experts can provide.
- $P(D|G)$ is called the marginal likelihood

If we assume $P(G)$ constant, then maximizing the 'a posteriori' information is equivalent to maximizing the marginal likelihood.

The Marginal Likelihood

We can express the marginal likelihood as:

$$P(D|G) = \int P(D|\theta, G).P(\theta|G)d\theta$$

A graph with more parameters implies a lower value: the probability of two identical values is greater than the product of the probability of the two values.

Marginal Likelihood Computation

- Under local and global parameter independence
- If 'a priori' distributions are Dirichlet
- If the data are all complete

Then, the marginal likelihood is

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(S_{ij})}{\Gamma(N_{ij} + S_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

Where each θ_{ij} is a Dirichlet $D(\alpha_{ij1}, \ldots, \alpha_{ijr_i})$ and $S_{ij} = \sum_{k=1}^{r_i} = \alpha_{ijk}$.

Parameters

The K2 metric is obtained when all the Dirichlet parameters are set to one:

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} (N_{ijk})!$$

With this score, equivalent graphs (they represent the same independencies) do not have the same score.
An equivalent score can be obtained by considering a global sample size S and then selecting the parameters for all the conditional probabilities according to the expression:

$$\alpha_{ijk} = \frac{S}{r_i . q_i}$$

Example. Neapolitan (1994)

Study about college students married by age 30. We want to test whether there is correlation between college graduation and getting divorced.
Variables:

- X_1 (1: graduate, 2: do not graduate)
- X_2 (1: divorced by 50, 2: do not divorce by 50)

We have two possible network structures:

Assume that we compute the equivalent sample size with $S = 4$. 'a priori' probabilities are D(2,2) and conditional probabilities are D(1,1).

Data

Suppose that we have the data in the following table:

$Case$	1	2	3	4	5	6	7	8
X_1	1	1	1	2	1	2	1	2
X_2	1	2	1	2	1	1	1	2

$$P(D|G_1) = \left(\frac{\Gamma(4)}{\Gamma(4+8)}\frac{\Gamma(2+5)\Gamma(2+3)}{\Gamma(2)\Gamma(2)}\right)\left(\frac{\Gamma(2)}{\Gamma(2+5)}\frac{\Gamma(1+4)\Gamma(1+1)}{\Gamma(1)\Gamma(1)}\right)\left(\frac{\Gamma(2)}{\Gamma(2+3)}\frac{\Gamma(1+1)\Gamma(1+2)}{\Gamma(1)\Gamma(1)}\right) = 7.2150\times 10^{-6}$$

$$P(D|G_2) = \left(\frac{\Gamma(4)}{\Gamma(4+8)}\frac{\Gamma(2+5)\Gamma(2+3)}{\Gamma(2)\Gamma(2)}\right)\left(\frac{\Gamma(4)}{\Gamma(4+8)}\frac{\Gamma(2+5)\Gamma(2+3)}{\Gamma(2)\Gamma(2)}\right) = 6.7465\times 10^{-6}$$

The first model (dependence) has more score (but little difference).

Searching an Optimal Graph

The space of possible directed acyclic graphs for n variables is huge:

$$\begin{cases} f(0) & = 1; \\ f(1) & = 1; \\ f(n) & = \sum_{i=1}^{n} (-1)^{i+1} . \binom{n}{i} 2^{i(n-i)} d(n-i) \end{cases}$$

Heuristics methods are necessary to optimize the score in this space.

The K2 Algorithm

- It is assumed an ordering of the nodes
- It is also assumed a maximum number of parents for each node
- Initial node does not have parents
- Then for each node, it starts with the empty set of parents and then add as parent the node that preceding it in the order, produces a bigger increasing in the score.
- It continues adding parents while the score increases and the number of parents in less than the maximum.

Decomposability

The key property of the score that makes this algorithm feasible is decomposability.

This score,

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(s_{ij})}{\Gamma(N_{ij}+s_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}$$

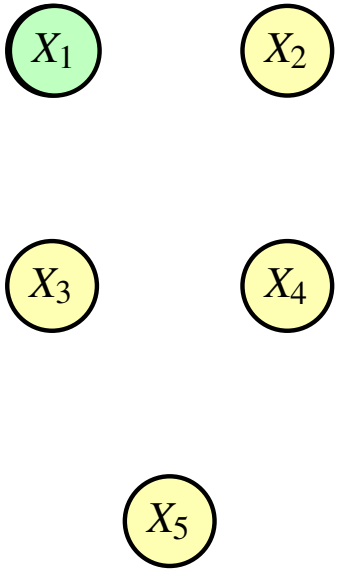
or the logarithm of it:

$$\log P(D|G) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \log \frac{\Gamma(s_{ij})}{\Gamma(N_{ij}+s_{ij})} \sum_{k=1}^{r_i} \log \frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}$$

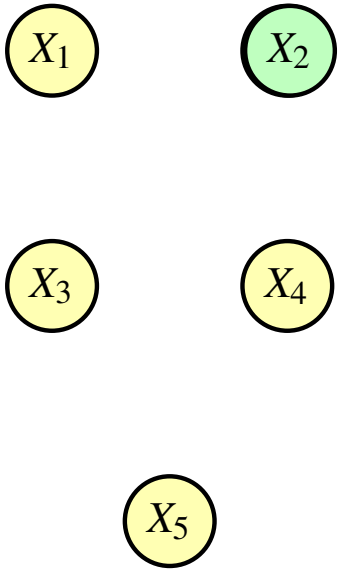
is that local changes imply only local computations.

The log marginal likelihood is a sum of functions depending of nodes and its parents. If only one node changes its parents, then only the part of the score corresponding to this node, has to be recomputed.

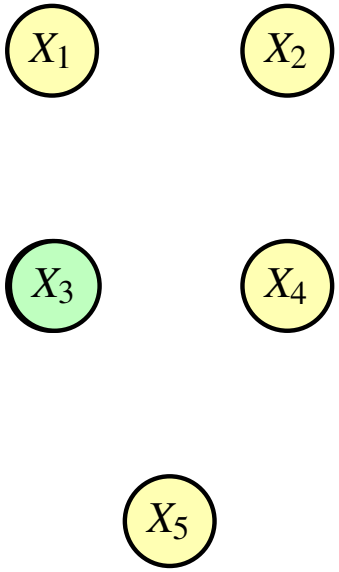
Example



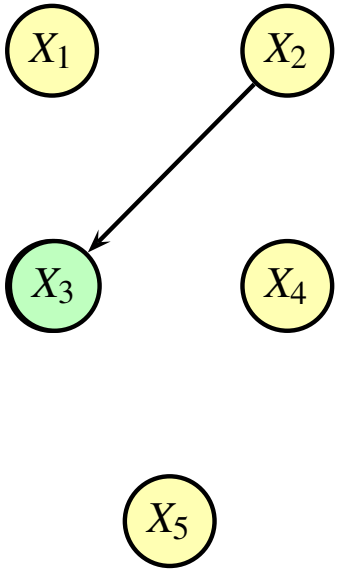
Example



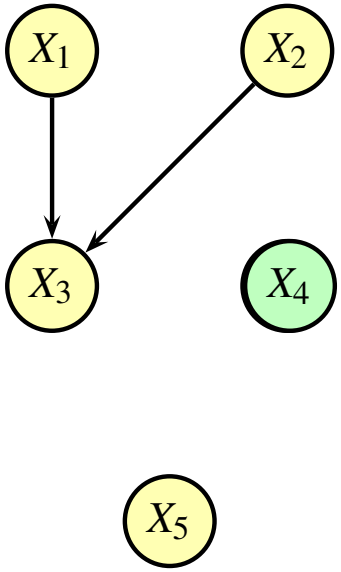
Example



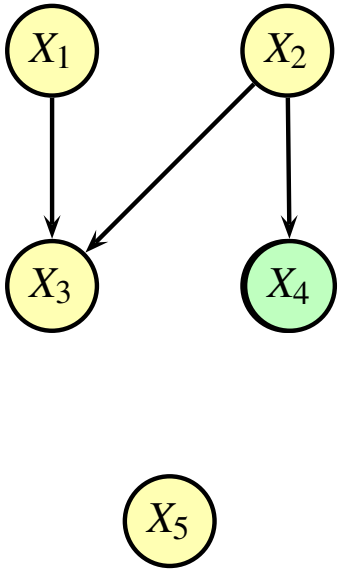
Example



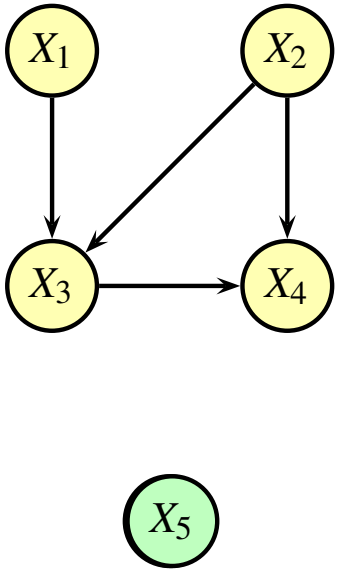
Example



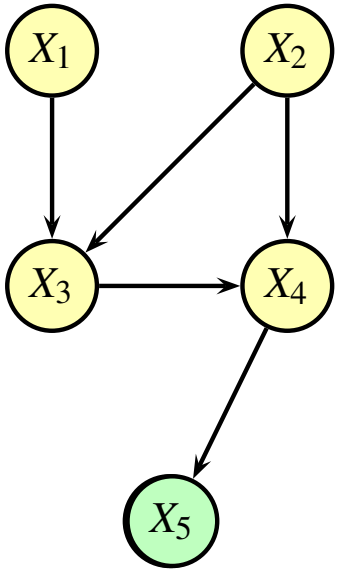
Example



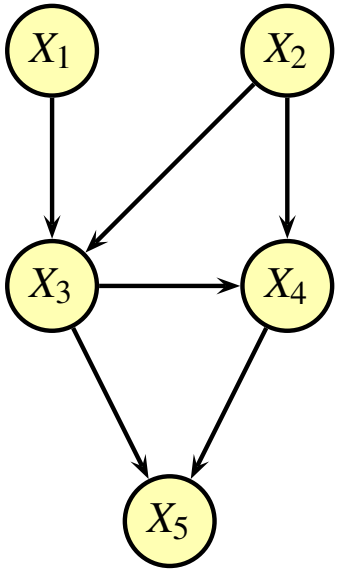
Example



Example



Example

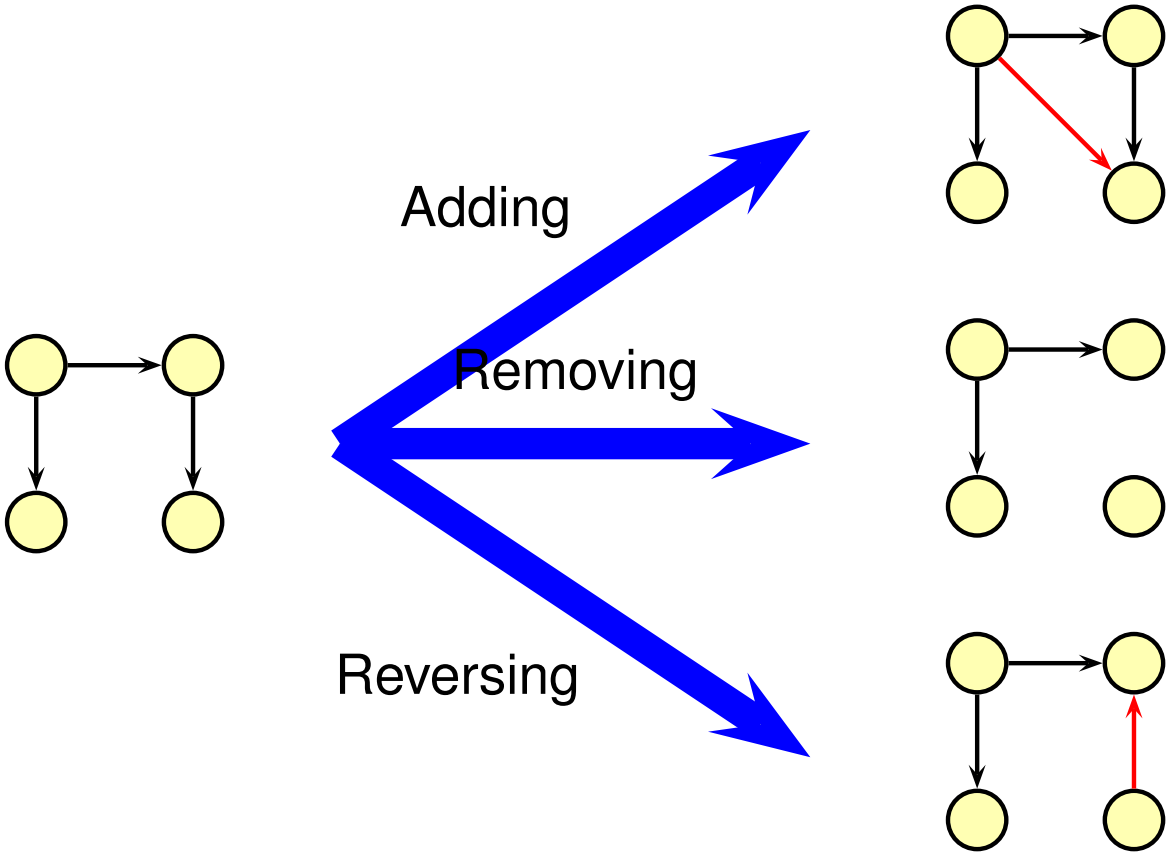


Learning under an order

- If we want to compute a network, compatible with the order $X_1, \ldots, X_n$.
- Build for each X_i a classification tree (upper entropy procedure) depending of variables $X_1, \ldots, X_{i-1}$.
- Assign interval probabilities to the variable X_i.
- Make all the variables appearing in the classification tree of X_i its parents, and the tree the conditional probability.
- It is possible to compute directly with trees.

Heuristic Search

More sophisticated search procedures have been considered. In general, they start with some Bayesian network (empty, K2, tree). The search space is explored with some basic movements. Typical ones are:



Exploring the Space

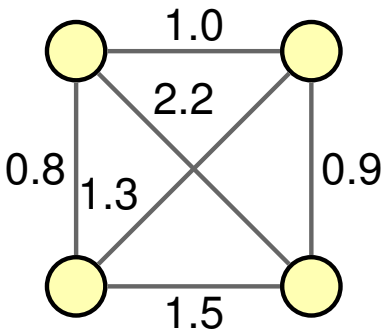
Different methods have been using to explore the space (NP-hard problem):

- Hill climbing
- Simulated annealing
- Genetics algorithms
- Variable neighborhood search

Searching for Trees

Searching for trees structures (each node at most one parent) can be done in polynomial time by means of the Chow-Liu algorithm.

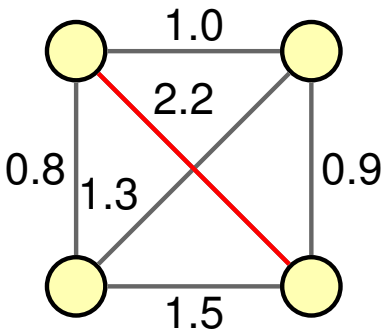
1. Construct a graph will all the nodes without arcs
2. For each pair X_i, X_j comp. weight $W(i,j) = Score(X_j|X_i) - Score(X_j)$
3. Find a tree with maximum weight
   (Kruskal algorithm)



Searching for Trees

Searching for trees structures (each node at most one parent) can be done in polynomial time by means of the Chow-Liu algorithm.

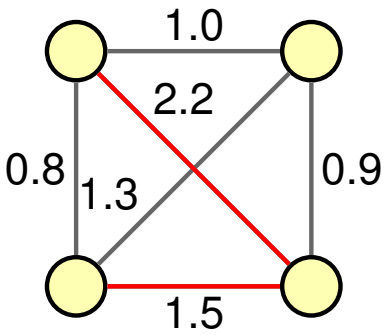
1. Construct a graph will all the nodes without arcs
2. For each pair X_i, X_j comp. weight $W(i,j) = Score(X_j|X_i) - Score(X_j)$
3. Find a tree with maximum weight
   (Kruskal algorithm)



Searching for Trees

Searching for trees structures (each node at most one parent) can be done in polynomial time by means of the Chow-Liu algorithm.

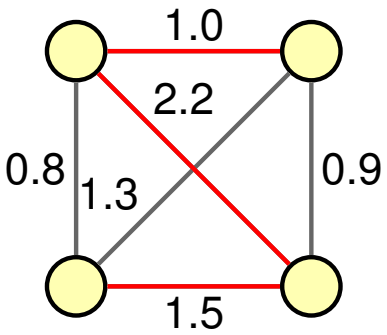
1. Construct a graph will all the nodes without arcs
2. For each pair X_i, X_j comp. weight $W(i,j) = Score(X_j|X_i) - Score(X_j)$
3. Find a tree with maximum weight
   (Kruskal algorithm)



Searching for Trees

Searching for trees structures (each node at most one parent) can be done in polynomial time by means of the Chow-Liu algorithm.

1. Construct a graph will all the nodes without arcs
2. For each pair X_i, X_j comp. weight $W(i,j) = Score(X_j|X_i) - Score(X_j)$
3. Find a tree with maximum weight
   (Kruskal algorithm)



Tree Construction

- Usually, it considers the mutual information as weight (likelihood score). The model complexity is controlled by the structure.
- It is efficient to build and sometimes provides good results
- Sometimes misses some important links, and it can add non-necessary links.

Model Selection-Model Averaging

In general, given some data we have selected an only model and all the predictions are done according to this model.
In a pure Bayesian approach to estimate the probability of an event E, we should average on all the possible models according to their 'a posteriori' probability given the data.

$$P(E|D) = \sum_G P(E|D,G)P(G|D)$$

In general, this is very difficult to compute and most of the models have very small probability. It is more common to select a set of more probable models $\mathcal{G}$ and then compute:

$$P(E|D) = \sum_{G \in \mathcal{G}} P(E|D,G)P(G|D)$$

Score+Search: Imprecise

- The upper entropy score can be generalized by considering:

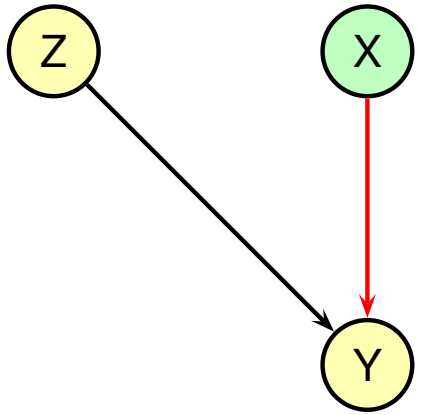
$$UENTR = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \hat{P}(\mathbf{pa_j}) \overline{H}(\mathcal{M}_{X_i}^{\mathbf{Pa_i}=\mathbf{pa_j}})$$

  Where $\mathbf{Pa_i}$ is the set of parents of X_i; q_i the number of configurations of $\mathbf{Pa_i}$; $\mathbf{pa_j}$ is the configuration number j of these parents; and $\mathcal{M}_{X_i}^{\mathbf{Pa_i}=\mathbf{pa_j}}$ is the credal set about X_i estimated from the part of the sample for which $\mathbf{Pa_i} = \mathbf{pa_j}$.

- This is a decomposable score.

Imprecise Dirichlet Score

- This is more difficult to generalize (even for two variables we used an approximation).
- There are a lot of algorithms (for example K2) based on determining the deletion or addition of an arc.



- To consider the score of adding arc from X to Y, when there is an arc from Z to Y, we could consider the same value $DEP - IND$ when testing independence of X and Y given Z.
- The score of deleting an arc is the opposite of the score of adding.
- We can design a greedy algorithm in which we always do the step with the highest score.

Different Models

- We have decide for independence if dependence does not dominate independence.
- We have the possibility of searching in different lines: the set of non-dominated models.
- One difficulty if the how to compare different models which have more differences than a single arc. It is possible to determine approximations (open problem).

Bibliography (non complete)

- P. Walley. *Statistical Reasoning with Imprecise Probabilities*. Chapman and Hall, London, 1991.
- M. Zaffalon, E. Fagiuoli. Tree-based credal networks for classification. *Reliable Computing* 9 (2003) 487–509.
- R.E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2004.
- D. Heckerman. A Tutorial on Learning with Bayesian Networks. MSR-TR-95-06, Microsoft Research, Redmond, Washington, 1996.